Web-based application for Collaborative Ethical Decision Making

Marzieh Alsadat Kazemi
Sam Eskandari
Abstract

Web-based application for Collaborative Ethical Decision Making

Marzieh Alsadat Kazemi and Sam Eskandari

Computer ethics is a scientific discipline, which can help us to reach ethical usability of IT-system by developing knowledge and provide the main principles and the guidelines of ethics. It helps us to collect and interpret information and use it in real design projects. To achieve this goal, we need an application to generate the structure of the procedure from analyzing the problem to gain the solutions.

The main objective of this thesis is to redesign and implement the EthXpert system as a web application in a way that users can access the software in a collaborative environment. The aspects of the thesis are choosing a suitable communication protocol, developing a collaborative ethical decision system as a web application and evaluating system by analyzing the answers of students (who will apply this software to our online survey as a research part of the thesis).

In order to implement ColLab, a research has been done about available APIs, which equipped by collaborative functionalities. ColLab is based on the most recent technologies such as Ajax, jQuery, MySQL, NodeJS, PHP, HTML5 (Canvas) and CSS3. The MVC pattern has been used to structure ColLab application. During the research phase of this thesis, the data gathered from the questionnaire has been analyzed based on qualitative and quantitative methods.
Acknowledgment

It would not have been possible to write this Master thesis without the help of the kind people around us.

First of all we want to thank Mikael Laaksoharju, our supervisor in Master thesis project. He helped us in all phases of this project and this thesis would not have been possible without his help. The good advice, support and friendship of our supervisor has been valuable in both an academic and a personal level. We really enjoyed of his professional ideas not only in the field of Computer Science but also in other fields of science and technology.

Sam Eskandari
I want to thank my lady of dreams, Farnaz, who helped me in all situations in my life. She gave me the best support during my education. I will never ever forget her kindness. I really couldn’t finish this thesis without her help.
I want to thank my sons, Shayan and Shervin, who gave me power and energy to complete this thesis project.
Finally I want to thank my parents who encourage me from childhood until now.

Marzieh Alsadat Kazemi
I would like to thank my husband Payam for his personal support and great patience at all times. My parents have given me their great support throughout, as always, for which my thanks likewise does not suffice.
# Table of contents

1. Introduction ..................................................................................................................... 1  
   1.1 Background ............................................................................................................... 1  
   1.2 Problem description .................................................................................................. 3  
   1.3 Aim ............................................................................................................................ 4  
   1.4 Delimitation ............................................................................................................... 5  
2. Theory and Related Work ............................................................................................... 6  
   2.1 Previous version ....................................................................................................... 6  
3. Technical Background ................................................................................................... 8  
   3.1 Collaborative Tools Comparison ............................................................................... 8  
   3.2 Drawing Tools Comparison ..................................................................................... 15  
4. Method ......................................................................................................................... 18  
   4.1. Implementation ...................................................................................................... 18  
       4.1.1 JavaScript......................................................................................................... 22  
       4.1.2 Ajax .................................................................................................................. 26  
       4.1.3 jQuery ............................................................................................................... 28  
       4.1.4 MySQL ............................................................................................................. 34  
       4.1.5 NodeJS ............................................................................................................. 44  
       4.1.6 PHP .................................................................................................................. 47  
       4.1.7 HTML5/Canvas and CSS3 ............................................................................... 62  
   4.2 MVC pattern ............................................................................................................ 65  
   4.3 User Experience Survey ......................................................................................... 68  
5. Result ........................................................................................................................... 70  
   5.1 Collaborative ethical decision making demo ........................................................... 70  
   5.2 Evaluation ............................................................................................................... 82  
6. Discussion .................................................................................................................... 96  
7. Future work ................................................................................................................ 103  
8. Summary and Conclusions ......................................................................................... 105  
9. References ................................................................................................................. 107  
10. Appendix ................................................................................................................... 112  
   10.1 Questionnaire ...................................................................................................... 112  
   10.2 Quantitative analysis............................................................................................ 112
1. Introduction

One of the advantages of human beings compared to the other creatures is human being’s ability to make a decision according to its knowledge and assumptions. One can make a good decision and other can make a bad one. Let’s ask some deeper questions about the words “good and bad”: What is the definition of good and bad? Does a “Good decision” hold its meaning in all communities and societies? Should a good decision satisfy a single person or it should also satisfy the community? When we talk about a good or a bad decision, do we consider laws and ethical issues? How deep do we do the analysis about the consequences of making an ethical decision?

Life is full of ethical decisions. Unfortunately, there are few people that can illustrate and tell how they come to make an ethical decision since it needs a deep knowledge of philosophy which is hidden behind the ethical decisions. Some people seem to think that ethical decisions are at about the same level of reasoning as their preference for a particular flavor of ice-cream. They will say, “I like strawberry; you like pistachio — and who is to say if it is right or wrong?”[1], but considering the questions above means that making an ethical decision is not a simple task.

In the next section, we will see two scenarios about why ethical decision making is important and what the consequences of each scenario can be according to the ethical decisions that are made by actors.

1.1 Background

In the previous section, we explained not only the importance of ethical decision making topic, but also how complicated such an ethical decision making is. In order to prove it, we will go through two scenarios:

Scenario 1:

Person X has a good friend Y, who has recently become a member of a dangerous gang. Both X and Y are over 18 which means that they are old enough to think and understand about the decisions, which are made. There will be no worries for X unless a dilemma is created. Whenever X becomes worried about Y, it indicates that X would like to act something about Y. Generally X has two options (there are obviously more than two options, but for simplicity we do not consider the other possible options):

Option A:

X does nothing. In this situation, X thinks that Y is experimenting his own life and it is a consequence of his age that Y is curious to experiment everything that is interesting for Y. X also considers that it is none of his business to do something and he has no right to interfere Y’s life.

Possible consequences:
Y will find out that being a member of a dangerous gang is not a good choice and he will quit himself from the gang soon.

Y increases his activity in the gang then he becomes more and more antisocial, dependent to that particular gang and possibly engages with some criminal activities to show his loyalty to the gang.

Y is either arrested or he gets killed because of criminal activities.

Option B:
X asks Y’s parents to advise him to quit the gang by his own wish before it becomes too late.

Possible consequences:

- Y becomes angry and he will be frustrated about the confidentiality of his friendship with X. Y continues his membership in the gang and the friendship between X and Y will be over.

- When Y’s parents were informed about Y’s participant in the gang and gave him some good advice, Y finds his fault and the risks of the membership in the gang, and after leaving the gang, he would be thankful of X’s concern. Their friendship becomes stronger because X proves his honesty and loyalty to Y.

Scenario 2:

Suppose that a website designer orders an online poker game website with real money transactions to a web designer. The web designer can design online application easily, but there are some moral and ethical issues that he should put into his consideration.

Some of these ethical and moral issues are:

- What if he designs this website so attractive that a lot of users are interested in playing with it, thus it will be so beneficial for the website owner, but what will happen to the users that might lose a lot of money while playing the game?

- What if he refuses to design this website while he is responsible to support his family economically?

- What if he designs it with the poor quality so that the users are not attracted to the game; however, he might lose his good profile he has achieved during years of designing websites?

He could make four decisions:

a. Not to design it at all.

b. Design it as real as possible.

c. Design it as an attractive game to satisfy the players and hold their expectations.

d. Design for the owner of the website’s economical benefits.
As you can see in the scenarios above, making an ethical decision is not simple and it can cause danger and harmful consequences. There are thousands of examples in the real world proving how important the ethical decision making is. In the following chapters we will focus on ethical decision making in more detailed perspective in the second scenario. In general, when making an ethical decision is misled, it can also cause members lose their jobs, lose money, being addicted to a bad behavior etc. In order to prevent making mistakes and misleading an ethical decision making, we should identify ethical problems and provide a good platform for dialogues by which these concepts are presented and discussed in the next section.

1.2 Problem description

In general, ethical issues talk about what is right (ethical) and what is wrong (unethical) by considering the concept of morality. It is not part of ethical issues when we do something right without any thinking in advance. Decisions fall into the realm of ethics when they pertain to things within our control, either showing respect or failing to show respect to human beings [23]. Look at ethics from the following perspective: Laws and rules were made not only to limit very bad behavior, but also to highlight good behavior. Bad behavior is further limited by your personal ethics, which tells you what is bad and what is good. Thus, some actions may be lawful, but still unethical. Unlawful actions are in the black area, good ones in the white area, and all in-between in the gray area. Ethics deals mainly with this gray area between the clearly described bad and good modes [25].

It is important to mention we have sometimes encountered such a situation that a problem is not choosing an alternative between a bad choice and a good one, instead we have to deal with a situation in which there are at least two good choices which we can choose between them. In other words, we have to choose between good alternatives and try to find the best choice between them. This is exactly the same as the first scenario that we talked about it in the previous section. In this case, an ethical problem becomes a dilemma. Consider that in scenario one, person X has two options, A and B, and both are good choices, but it is hard to choose the right option between A and B.

In order to find and select the best choice, we need a platform for dialogue. In this platform, we have to identify the problem itself and then try to get help from other persons that they have at least some knowledge and experience about the problem. We should at least consider the regulations, human behaviors, boundaries and communities in the dialogue, which is the way leading us to find the best choice. People should not think dogmatically when they are trying to make an ethical decision, instead it is important to think about the facts and analyse the problem itself. For more clarification about dogmatic thinking, it is important to mention two definitions:

- Heteronomy: In heteronomous thinking, we just follow the rules neither thinking about the consequences nor any further/deeper investigation about the rules related to the reasons why we should follow them. Heteronomy is automatic and it is dogmatic since there is no argument about why we should follow these kinds of
rules. It is also constrained by authoritarian thoughts. Heteronomy is based on human instincts and reflexes.

- Autonomy: In autonomous thinking, we combine heuristics and analytical thinking. In autonomy we also use systematic thinking. It is holistic and can be applied to the same moral problems and we can have a good supervision to handle the moral and ethical problems.

One example of heteronomous thinking is people’s belief of number 13 as an unlucky number in many cultures i.e. they avoid to start big projects or do important things on 13th of the month. By heteronomy, people just follow the rule that 13 is an unlucky number and they will not do any special or important activities on that day because in this case they will face a failure at the end. In autonomous thinking, people try to find out if 13 is really an unlucky number or it just comes from superstitious views, then they do some investigations and try to find the correct answer without just accepting it.

Again, when we want to make an ethical decision we should not think in heteronomous way. Instead we should think in an autonomous way. By autonomous thinking we are able to create a good environment for dialogue and finally a good dialogue leads us to social awareness. In order to achieve our goal which is making an ethical decision, we can use a computer based tool which can help us reach our goal. Implementing this tool is one of the major parts in the current thesis project that is explained in the next section.

1.3 Aim

The objective of this project is to redesign and implement the EthXpert system (see http://www.it.uu.se/research/project/ethcomp/ethxpert/) as a web application in a way that users can access the software both synchronously and asynchronously. To achieve this goal, we intend to implement a web-based application for collaborative ethical decision making with the following specification: a system that provides functionality for collaborative working, which is implemented on the basis of EthXpert system; even though, it represents itself as a web application with new features and functionalities. To use such an application, the user needs to sign up and if a person has already signed up then S/he needs to login to the system. Users are allowed to create a project and to invite other users, who have already signed up to the system, as an editor or a viewer to the project. The current system has been implemented on the basis of EthXpert system and contains description, network, relations, evaluation and conclusion tabs; however it applied new functionalities which have been added to the system and enhanced its productivity. The thesis will also present the result of the survey in which the participants are students who have taken IT Ethics and Organization course in spring 2012. This web-based application is called ColLab (Collaborative Laboratory).

The challenging part of this project is to connect NodeJS platform (or similar services for collaborating perspective) to both the application and the server so that the users will be
able to access the system synchronously. They will also be able to have a collaborative participation in a specific group of the students.

This project contains the following aspects:

- Choosing a suitable communication protocol and platform for the application.
- Development and implementation of a collaborative ethical decision support system as a web application, deriving from the ideas in EthXpert.
- Evaluation of the system by analyzing the students’ responses who will apply this software to our online survey as a research part of the thesis.

1.4 Delimitation

We will not put much focus on security aspects in the first implementation. However, since it is a collaborative space, and accessible everywhere, we need to make sure to have at least some login requirements and input validation.

The application is not responsible for correctness and consequences of decision that is made by users.

Because of the limitation of time, we did not go through/investigate all collaborative tools which are available in the world and we discovered some of them and chose the best one according to our needs.

According to the huge amount of web browsers, we check the functionality of the application in Mozilla Firefox 11.x and Google Chrome 11.0.x. ColLab is not working properly in Microsoft Internet Explorer because some technologies that are implemented in ColLab is not standardized yet e.g. (HTML5). Finally, during the implementation of ColLab we do not consider the mobile devices and tablets; however, most of the functionalities that are implemented in ColLab work properly on these devices as well.
2. Theory and Related Work

Ethical decisions come from conscience which is like an internal source of reward and punishment in some point of views. However, according to Lawrence Kohlberg, conscience is only one of the several ways in which ethical values are shown in the personality. [21] After deep discussion we occasionally have to accept the fact, some behavior and decisions must be considered as the correct one. Sometimes when we do not have a crucial personal interest, the conclusion is acceptable. When we do have personal interest and have to gain it in legitimate time, we might need some help to structure our thinking. We need to apply methods to counter our prejudice and inclinations. Thus we can evaluate the situation as objectively as possible. In fact, this is in the interest of decision maker. If one aims for personal gain, it will not be considered as a good judgment [13].

Related Work: There are a variety of tools with the aim of ethical concerns that make implicit or explicit assumptions. They use ethical theories which help us more in making ethical decisions. Different tools which support ethical decision making, have targeted to determine relevant information in various ways. We will discuss different tools which support ethical decision making in following parts. Paramedic Ethics (Collins and Miller, 1992) has targeted the obligations and responsibilities of decision makers. Based on the target of the system, a user provides relationships between stakeholders, and determines the considerations for various opportunities and instabilities which occur from alternative solutions. Eventually, a consulted social contract is assessed as a possible solution. In SoDIS (Gotterbarn, 2002), a user collects background information about the problem and related stakeholders and then immediately to answers questions targeted at determining causes for moral problems. In ETHOS (Mancherjee and Sodan, 2004), the user is supported to determine the open moral questions by taking the role of a moral agent, then the benefits of alternative solutions are assessed based on ethical theories. It needs to mention that the first two technical tools are aimed for computer professionals, working in technical development projects. However, ETHOS is not intended for specific audience and does not presume any specific content in the problem to be evaluated. Value Sensitive Design by Friedman, Kahn and Borning (2008), is defined by the authors as an interactional theory which contains all imaginable values that different philosophies have noticed to exist. All above systems and theories are great to systematize, coordinate and lead the user in ethical issues. Nevertheless, there is a risk that the user is confused and gets lost; therefore the user misses the practical problem which is the clear main goal [13].

2.1 Previous version

The project ETHCOMP - Ethical Competence for Decision Makers and Organizations - started in fall 2007. EthXpert is one of the methods developed in the research project, which is the previous version of current application. EthXpert is a tool targeted to enhance codes of leads and guidelines and helps users as decision makers in the process of handling ethical problems. It targets a broad audience and does not presume any specific content in the problem analysis. The process of ethical decision making with EthXpert
concentrates on the interests, values and principles of stakeholders involved in the problem situation. Since moral problems are complicated, they involve less and more number of stakeholders and interests. The matrix representation is one of the main parts of the tool, which assists to improve an aggregation of available information about a problem systematically. In the process of analysing problems, options and interests of each stakeholder are determined, and cell of the matrix are filled with profits and risks. Any relationship among stakeholders should be assessed and the evaluation should also be tried on other pairs. The tool should not given any clue about the rightness of any conclusion. Therefore, it will force users to analyze the problem cautiously. EthXpert has targeted to help users to organize and to handle a problem but the problem should not be narrowed down. Therefore, users can be grateful of having a full impact of a decision. Ethxpert allows users to face the problem and freely add information to the analysis whenever there is a reason for it but the real problem occurs when there is a large amount of data. In this case, a better way is to allow the decision maker to select when S/he needs to analyze the data. It is also worth to mention that, EthXpert is designed to block reasoning which is a type of prejudices and to support a systematic and self-critical approach towards moral problems [13].
3. Technical Background

A major part of this project focused on pre-studies in which it compares tools that should be chosen for real-time collaborative editing feature and drawing graphical object in Network tab (Drawing part). Since real-time collaborative editing feature is one of the main aspects of the project, we have compared different tools with pros and cons. It is also required to select an appropriate drawing tool to fulfill the goal behind the drawing part of the application. Therefore, it seemed necessary to choose the right one and implement the application based on those studies.

3.1 Collaborative Tools Comparison

A collaborative editor is a form of collaborative software application which enables several users to edit a computer file using different computers. There are two types of collaborative editing tools which are specified as real-time and non-real-time. Real-time collaborative editing (RTCE) tools allow users to edit the same file at the same time [9]. These type of tools can be used for both documentation and collaboration in several tasks. There are several tools for RTCE applications such as SubEthaEdit, Gobby, MoonEdit, DocSynch, LivePad, and etc, and they are altered in different dimensions. Some of the tools have been evaluated for the ongoing project.

**SubEthaEdit:** This collaborative tool allows several users simultaneously to type and see what other users are typing on the same document. SubEthaEdit was developed by a group of students at the Technical University of Munich who are known as The Coding Monkeys. Since it is available only for Macintosh, originally designed for coding, which means it is not compatible with our needs [2].

![SubEthaEdit application](http://www.codingmonkeys.de/subethaedit/)
**Gobby**: Gobby is available for all operating systems and this is a free collaborative editor tool which supports multiple documents in one session, on the one hand has the functionality of multi-user chat, On the other hand. However, Gobby is originally designed for coding and programming synchronously, is not compatible with our needs [3].

![Figure 2: Gobby application (http://gobby.0x539.de/trac/wiki/Screenshots)](http://gobby.0x539.de/trac/wiki/Screenshots)

**MediaWiki**: MediaWiki is a free web-based application, developed by wikimedia foundation. The first version of this application was used to supply the needs of Wikipedia encyclopedia in 2002. This tool is used to manage documents and it is also available for all operating systems. MediaWiki’s power allows multiple users to edit a document.

![Figure 3: MediaWiki application (http://mediawiki.en.softonic.com/images)](http://mediawiki.en.softonic.com/images)

**Text Flow**: TextFlow is an Adobe AIR application which is a collaborative document editor and providing some functionalities of editing online and back-end file storage. However, it
is not a real-time collaboration platform and the Sign-in functionality with Yahoo and Google accounts as links on the main page did not work. The error which appeared through the operation was identified as “User not found” but the user-name was valid [5].

Figure 4: Textflow application (http://www.appappeal.com/app/textflow/)

**Twiki:** Twiki is an easy to use collaboration and application platform. This collaborative tool enables users to create web applications and allows developers to extend its functionalities. Since it includes lots of facilities which could make our application heavy, it is not compatible with our requirements [6].

Figure 5: Twiki application (http://webscripts.softpedia.com/scriptScreenshots/TWiki-Screenshots-16171.html)
**JotSpot Live:** A real-time web-based collaboration tool allows users to signup and to create documents which users can work together on synchronously. It focuses on the whole text of the document, ending up with a document which everyone agrees on. However, the other users are not able to view what a user has edited when s/he has not saved his/her document. It could also cause some misleading mixtures when two users work at the same time on the same part of document [7].

![JSpotLive application](http://freewarewiki.com/w/page/14935778/JotSpotLive)

**SynchroEdit:** SynchroEdit is a web-based document editor and an open source which enables multiple users to edit a document simultaneously and continuously adjusting all modifications so that users always have the same version. However, SynchroEdit only supports Mozilla/Firefox and Internet Explorer and it doesn’t work on any other browsers. The official website was also closed [8].

![SynchroEdit application](http://foswiki.org/Development/SynchroEdit)
**Google Wave**: Google wave is a software framework for real-time communication and collaboration. Google wave was released on May 2010 to the general public. However, on November 2011 Google declared that the all Waves would become read-only in January 2012 and they all would be eliminated in April 2012. Google wave is an open source software and all software developers are allowed to develop its features. However, Google wave was shut down in April 30, 2012.

![Google Wave](http://de.wikipedia.org/wiki/Google_Wave)

**Etherpad**: Etherpad is a web-based real-time collaborative editor, allowing users to edit a text document simultaneously. Each user can create a pad which is a new collaborative document. Each pad has a pre-defined URL which is unique for itself and anyone who knows the URL can edit the same pad at the same time.

![Etherpad](http://www.enterprisescreenshots.com/screencast/Etherpad_screenshots/)

---

12
Etherpad Lite: Etherpad Lite is a complete version of Etherpad software based on varied technical foundations. A user can create a new collaborative document which is known as a “pad”. Each pad has a specific URL and if user wants to share the pad with other users, s/he should give the other users the required URL address and so that the other users can participate in the associated chat. The software saves the document automatically. The aforementioned tool also contains revisions feature so that the time slider feature allows users to check the history of the pad. There is also a possibility for the password protection. Etherpad Lite was implemented in JavaScript using nodejs with the real-time functionality (called Easysync).

![Figure 10: EtherpadLite application (http://www.mediawiki.org/wiki/Extension:EtherpadLite)](http://www.mediawiki.org/wiki/Extension:EtherpadLite)

Advantages

- Users can sign-in to the system.
- Users can change its name in the chat when s/he signs in with special identity.
- The last name which enters in the chat will appear in the next entrance to the system by user.
- The ability to create a public document quickly and easily.
- When another user signs in the system with the same identity, it makes the first user disable to change the document.

Disadvantages

- In revision part the name of the user is not saved in the history and the related date and time are recorded and we have a kind of versioning character by character.
- There is sometimes a lack of connection when some information is sent for other users to be share.
- At intervals, there is a delay problem for chatting windows when you submit a message.
Responsive Open Learning Environments (ROLE): ROLE technology enables users to build their own learning environment based on their requirements and selections. The elements of learning environment can be put together to make new functionalities, which can be matched by single learners or collaborating learners for efficient learning and meet their own needs. It allows users to make new tools and functionalities based on their needs [10]. However this technology has some problems such as:

- Very low internet speed
- When users work collaboratively, they are not able to see the changes on browsers without refreshing them.
- We could not find any possibility to delete the activity on the page.

Summary of tools comparison:

<table>
<thead>
<tr>
<th>Tool's name</th>
<th>OS</th>
<th>Browser</th>
<th>Versioning</th>
<th>Access Control</th>
<th>Simplicity</th>
<th>Open Source</th>
<th>real-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubEtha-Edit</td>
<td>Mac</td>
<td>Firefox, Internet Explorer, Opera, Safari, etc.</td>
<td>Yes</td>
<td>Editor/Viewer</td>
<td>Light</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Gobby</td>
<td>Windows and Unix</td>
<td>Firefox, Internet Explorer, Opera, Safari, etc.</td>
<td>Provide local group Undo</td>
<td>Owner/Editor</td>
<td>Heavy</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Media Wiki</td>
<td>All OS</td>
<td>Firefox, Safari, Opera, Chrome, Internet Explorer</td>
<td>Yes</td>
<td>Administator/Editor</td>
<td>Heavy</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>TextFlow</td>
<td>Windows/ Mac and Linux</td>
<td>Must support Flash 9 or more</td>
<td>Yes (Undo/Redo)</td>
<td>Owner/Editor</td>
<td>Heavy</td>
<td>Not clear</td>
<td>Yes</td>
</tr>
<tr>
<td>Twiki</td>
<td>All OS</td>
<td>Any web browser</td>
<td>Yes</td>
<td>Editor/Viewer</td>
<td>Light</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>JotSpot</td>
<td>Not clear</td>
<td>FireFox,</td>
<td>Yes</td>
<td>Owner/Editor</td>
<td>Heavy</td>
<td>Not</td>
<td>No</td>
</tr>
</tbody>
</table>
3.2 Drawing Tools Comparison

The first step in using ColLab application is to create an overview by drawing the network of a stakeholder, for example a map over the relations between all stakeholders [13]. The step also includes assigning interests in order to needs, values and principles to the stakeholder [13]. Focus should be on how to select an appropriate drawing tool to fulfill the goal behind this step.

**HTML5:** HTML5 is a text-based markup language for structuring content such as text, images and hyperlinks in documents [14]. The language is not yet fully standardized but there are already sophisticated designs. It includes many new features such as video, audio, canvas elements as well as the integration of scalable vector graphics (SVG) and dynamic 2D and 3D graphics [14]. These features are designed to include and handle multimedia and graphical content easily on the web without resorting to proprietary plugins and APIs. However, the canvas element and object is the big news in HTML5, an old graphics object in the form of SVG has also joined to the new technologies.

**SVG:** Scalable Vector Graphics is in the format of vector graphics file. Vector images are generated through commands which are text-based. JPEG and GIF images on the web are bitmapped so they always remain in a particular size. A bitmap is only a file which contains a list of color values for each pixel in the image. In contrast to JPEG and GIF
images, SVG images are scalable to be adjusted to the size and resolution of the viewing window. SVG is a programming language for graphics [15]. Using the SVG DOM, you are able to style SVG with CSS, having dynamic behavior.

**Canvas:** The canvas element in HTML5 is the feature that most developers used to develop rich web applications without requiring to install plug-in such as Adobe's Flash Player. Drawing are allowed to be designed on the canvas via the canvas 2D API. This API includes a plethora of functions which give you the power to draw anything you like on the canvas. Canvas is a new element in HTML5 that can be considered to draw graphics using JavaScript. It is allowed to be used for dynamic, scriptable rendering of 2D shapes and bitmap images [16]. Canvas looks like the `<img>` element. The only difference is that it does not include src and alt attributes. It consists of a region which is drawable in HTML code with the attributes of height and width. Both attributes are optional and can be set using DOM properties. The element can be sized by CSS; however during the process of rendering the image is scaled to fit the size of its layout. By coding a set of drawing functions in JavaScript, a developer can access the region. The developer is allowed to generate graphics dynamically. For instance, canvas can be used to draw graphs, to make photo compositions, and to build animations. Canvas was first introduced by Apple for the Mac OS. It was implemented in Safari and Google Chrome and also supported by Firefox [17]. When you send drawing codes to the canvas, the browser directly sends them to the graphics hardware directly. Speed of hardware works fast to deliver real-time animations and graphics without decreasing the speed of the surrounding user experience [18].

**Comparison of HTML5 Canvas and SVG:**

**Canvas:**
Advantages

- High performance of 2D surface to draw anything you like [15].
- The performance is constant. Since everything is a pixel, the performance is just reduced when the image resolution increases [15].
- You can save the image which is resulted as a .png or .jpg.
- It generates raster graphics (for instance in games) and edit images requiring pixel-level manipulation [15].

Disadvantages

- Since everything is a pixel, there are no DOM nodes for anything you draw [19].
- There is no API for animation [19].

**SVG:**
Advantages
• It covers a kind of Resolution independence which allows scaling for any screen resolution [15].
• SVG includes a very good support for animations.
• Developers have a full control over each element.

Disadvantages

• If the browser’s implementation is slow or buggy which limits the programmer’s chances to fix it. However, the programmer under canvas can usually find a way to have a better performance and draw things accurately [20].
• It is almost impossible to get a raster image like PNG from SVG content in a browser [20].

Canvas in HTML5 uses a simple retained mode graphics approach. This simplicity reflects how developers use a scene graph approach. The objects are held by the scene and they are rendered in the view port afterwards. Another advantage of simplicity of the specification is that different browsers look to behave the same way incredibly. You can access the pixels easily and fast. Canvas implementations have improved significantly and hardware acceleration has been massive performance improvements over last year in most browsers [20]. According to our research we have decided to choose canvas in HTML5 to draw graphical objects and their related features in Network tab.
4. Method

In this chapter we will see which technologies, why and how these are used in order to implement the web-based application tool for ethical decision making (ColLab) based on EthXpert software. We will not go through the programming code in this chapter, instead we will mainly explain how these technologies are used to implement ColLab, which modules that are used in ColLab and how they are connected and work together.

At the end of this chapter we will discuss also about the user experience survey, which we have created to get the feedback of students, who used the application to fulfill their assignments.

4.1. Implementation

ColLab is a web-based application that consists of two major parts:

- Server side parts
- Client side parts

Both server side and client side of ColLab are implemented, tested and run in NetBeans IDE. We have tested ColLab under Microsoft Windows 7 Professional (both x86 and x64) with Mozilla Firefox 11.x and Google Chrome 11.0.x. Since we are two students that we are working on the same thesis, we have used Uppsala University’s CVS in order to implement the application both in parallel and as individual.

Server side: Since ColLab is web-based application then there is a requirement that we should use a web server. In order to save and retrieve data that are a part of the service of ColLab we need also a database server and finally since ColLab is a collaborative application then we need a server to implement the collaborative environment. We have used Apache XAMPP as our web server for implementing and testing ColLab. MySQL is a part of XAMPP that it is used as our database in order to save and retrieve data and finally NodeJS server which it is based on JavaScript server side and it acts as a web-based collaborative real time editor. The software that we have used for web-based collaborative real time editor is Etherpad-Lite (see figures 13 and 14). It is important to mention that the programming language that is used in the server side is PHP which means ColLab is capable to run under Apache servers and it is not limited to use just in XAMPP. Another advantage of ColLab is that it is not limited to use just MySQL as database server instead it is fully compatible with other databases such as Oracle and Microsoft SQL Server.

When a client or browser requests a web page by sending a HTTP request, web server checks if all parts of the requested pages is available or not. If those parts are available then web server answers the HTTP request by HTTP response that is the requested page (figure 11).
If all parts of the requested page are not available then there are two possible situations according to the requested page:

1. Requested page contains data that it should be gathered just from database: In this case web server needs only to communicate with the database in order to retrieve previously saved data from database. By using either a query or couple of queries the web server will get the data that those are needed in order to show the page to the client/browser. Whenever web server retrieve the data from database then it answers the HTTP request by sending HTTP response that is the requested page from the client/browser. The scenario for this situation is shown in figure 12.

2. Requested page contains collaborative elements: In this case, when client/browser sends its HTTP request to web server, the web server answers to the HTTP request by HTTP response and also web server will create an iframe in order to client/browser be able to directly communicate with NodeJS server. Whenever the iframe gets created, NodeJS server opens a dedicated port to client/browser in the way that by using this port client/browser is able to listen to the NodeJS server and
send data to NodeJS server. Since the main act of NodeJS server is to prepare collaborative real time editor environment then the communication port between web server and NodeJS server is in two ways for listening and answering to the requests. NodeJS server needs to retrieve data from database server in order to send it back to the client/browser therefore after opening a port between client/browser and NodeJS server, the NodeJS server sends a single query or couple of queries to fetch data from database server. The result of query/queries from the database sends back to NodeJS server and finally the data will be send back by NodeJS server to the client/browser by the port that was opened by NodeJS server. The scenario for this situation is shown in figure 13.

![Figure 13: Communication between client and server when requested page contains collaborative elements.](image)

The entire procedures that are mentioned in the server side section are summarized in figure 14. It is important to mention, web server produces proper failure code in the case of communication failure between each parts of the server side i.e. if a requested web page is not available or there is failure in database connection in order to show in a web page.
Figure 14: Communication between servers that are used in ColLab.

We will discuss more about the server side parts in section 4.1.4, 4.1.5 and 4.1.6. In these chapters we will look more deeply how they are implemented, how each module communicates with other module(s) and finally which requirements are needed in order to make ColLab work properly.

**Client side:** ColLab is designed in the way that it reduces the network traffic as much as it can. In order to reduce the network traffic, some services just run at the client side without further involving the server side. Whenever the data is fully created and confirmed at the client side, the result will be sent to the server in order to either save into the database or start new procedure. One example is that when a user try to submit the registration form during signup process when the content of password retype field is different than the password field. S/he is not able to submit the registration form until the contents are the same and this procedure is controlled by JavaScript that is client side.

Sometimes ColLab should mix both client and server side technologies in order to process the data. An example of this situation is when a user wants to add his/her friend as a member of the project. In this case user writes the email address of his/her friend in a field and click on the invite button. ColLab checks the availability of the email address that is written by the user by sending an Ajax call to the web server. The web server itself sends the given email address to the database by a query and checks whether the email is available or not by getting result set from the database. If email is not registered in database then ColLab shows a message to user that the email address is not registered in the system and it prevents the server to send back the entire web page. Instead it just
shows a message by jQuery. We will discuss about these kinds of functionalities in sections 4.1.1, 4.1.2, 4.1.3 and 4.2.

In the following sections we will discuss about the technologies and programming languages that are used in order to implement CoILab. We will also discuss about the MVC pattern that we have followed during the implementation of CoILab.

### 4.1.1 JavaScript

JavaScript is a scripting language that is dynamic, weakly typed and has first-class functions. It is a multi-paradigm language which supports object-oriented, in both imperative and functional programming styles. JavaScript is primarily used in the form of client-side JavaScript, implemented as part of a web browser in order to optimize the user interfaces and dynamic websites. It copies many names and naming conventions from Java, but these two languages are unrelated and have quite different semantics [52].

Since we have selected canvas environment for network tab in our project, we decided to use a JavaScript library to draw, drag and drop and resize graphical images on the canvas. We have done research on JavaScript libraries between KineticJS and Simon Sarris. We will discuss about the one we have selected to fulfill our goals in the following.

- **KineticJS**: KineticJS is an HTML5 canvas JavaScript library that extends context by enabling high performance path and pixel detection for desktop and mobile applications. KineticJS applications need a container DOM element in the HTML page to contain a stage which is made up of layers, shown in figure 15. Each layer can contain groups or shapes and it is tied to its own canvas element. You can add event listeners to shapes, groups, layers, and the stage, and move them, rotate them, put them on top of each other [53].

  For event handling, the KineticJS stage is made up of a background layer and a buffer layer which provides high performance path and pixel detection. Each shape is part of a particular layer depending on the shape which can do its activity in related type. Animations, transitions, and drag and drop operations are particularly smooth because developers can create an unlimited number of the user’s defined layers which enable them to redraw some shapes while not touching others [53].

![Figure 15: A stage is made up of layers [53].](image)
KineticJS is a quite big JavaScript library. You have to add shapes to the stage and the layers when you want to draw a shape (graphical object) on the canvas. However, we wanted to use a simple library that only has the functionalities such as selecting, moving and resizing. We wanted to be able to maintenance codes and adapt them easily to what we need. Therefore, we decided to use a small JavaScript library which was written by Simon Sarris and will be discussed in this section, later.

- How we used JavaScript library for HTML5/Canvas in Network tab:
We will discuss how to create a simple data structure for shapes on an HTML5 canvas and how to make them selectable, movable and resizable. It covers all functionalities which we required in the network tab. The codes in the JavaScript library are written primarily to be easily comprehensible.
In fact, a canvas is not smart; it is just a place for drawing pixels. If you ask canvas to draw something, it will execute the drawing command and then immediately forget everything about what it has just drawn. Therefore, we must keep track of all things when we want to draw (and re-draw) each frame by ourselves [55]. The main parts of the codes in JavaScript in Network tab will be illustrated in the following:

1. Keeping track of objects: To keep things simple we made a rectangular object (stakeholder) called Box which has 9 attributes such as width, height, color, name, array of interests, id and its x and y axis. We also made a method to create boxes called addRect(). We have added a bunch of variables to keep track of the drawing and mouse state. Once each Box object is created, it will be added to an array of objects called boxes[ ], to keep track of each object. However, we also need a variable for the canvas, the canvas’ 2d context where wall drawing is done [55]. We also have used a second canvas for the selected purposes which will be discussed later.

2. Drawing: Since our canvas is animated, we need to set up a draw loop which was done in init() function. We have to draw at a frame rate, for example every 20 milliseconds. However, redrawing does not only mean drawing the shape over and over; we also require to clear the canvas before every redraw. If we do not clear canvas, dragging will look like the box which is creating a solid line because none of the old positions of the box will disappear. Therefore, to clear the entire canvas before each Draw frame, we consider the canvas to be either valid or invalid. If anything just got drawn, the canvas is valid and there is no need to draw it again. Although, if we do something like adding a new box or trying to move a box by dragging it, the canvas will get invalidated and mainDraw() function will do a clear-redraw-validate. The canvas invalidation is the only optimization we have used to draw on canvas. In order to draw graphical objects on canvas, as we have mentioned above, we went through all of boxes[ ] and draw each one, in order from the first box to the last one. We have a Box class in which each box is responsible to draw itself with draw() method. In the Box class, the draw function is added to all instances of the Box class. This will give the nice appearance of the last boxes looking as if they are on top of earlier boxes. After all the boxes have been drawn, a
selection handle got drawn around the box which is called as mySel. You can see a stroke and 8 selective handles along the edge of the selected box in figure 16.

3. Mouse move events: Now we have objects, initialization and a loop that constantly re-draw when needed. We also have to make the mouse do things such as upon pressing, releasing, and double clicking. With MouseDown event we require to see if there are any objects we could have clicked on. In order to handle this event, we could do something easy and only check the bounds of each of the boxes, see if the mouse coordinates x and y positions within the boxes width and height range. However, we have done the required selection in a general way, we have drawn each shape on to a “ghost” canvas, to see if the mouse coordinates lie on a drawn pixel or not. A ghost canvas is a second and a temporary canvas that we have created in the same size and shape as our first and normal canvas. It will never be seen because we only create it in the code and is never added to the page [55].

4. Drawing the eight boxes that make up the selection handles: The eight selection handles are unique which allows us to resize an object in a different way. For instance, by clicking on the top-middle one, the user can make the shape taller or shorter. We check in the draw() function if the current box is selected, and if it is so, we draw the selection outline as well as the eight selection boxes which will be placed on the selected object’s bounds. In the init(), initialization function, we defined an array, selectionHandles[], which holds the 8 tiny boxes (as it is shown in figure 17) that will be our selection handles [55].

```
0 1 2
3  4
5 6 7
```

Figure 17: The numbers shows the order of 8 tiny boxes around the selected box.
5. Changing the mouse cursor when it is over a selection handle: Since we need to have selection handles in our program, it is nice to have the mouse cursor change to show that an action can be performed. In order to do this, we need to know where the mouse is located all the time and see if it is over one of our eight selection handles. In this case, we made a myMove() function to handle the events when the mouse moves inside the canvas. Thus, if there is something selected and we have not already dragged, we will execute some codes to see if the mouse position is over one of the selection boxes. If it is, we give the mouse cursor the related arrow. If it is not over a selection box, we should be sure that the mouse cursor changes back to the normal pointer. We have defined two other variables, isResizeDrag and expectResize. expectResize could be a number between 0 and 7 to indicate which selection handle is active currently. If none of them is active, it will set to -1 by default. isResizeDrag works like isDrag and its value could be true or false. It becomes true if two conditions are met:
1. expectResize variable is set to one of the selection handle number (which is not -1).
2. The mouse is pressed down. In other words, it happens if the mouse is over a selection handle and has been pressed.
We wanted to draw arrows from the selected box to the other boxes (with the aim of making relations and connections between stakeholders), therefore we made canvas_arrow() method. This method is called in the draw function whenever a box is selected on the canvas. While an arrow is drawn, the x and y axis of the beginning path is based on arithmetic dealing on the x and y axis of the selected box. To draw the arrows, which are mainly lines on the canvas, the other boxes will be checked if they are on the left or on the right side of the selected box. Then, the arrows are drawn based on some arithmetic dealing according to the x and y axis of the selected box that is shown in the figure 18. Since we intend to have resize functionality for the graphical objects on the canvas, we need to have myMove() function to be active all the time [55].

6. Resizing the graphical images: There are bunch of arithmetic dealing to resize the box by each handle accurately. For instance, as you can see in figure 17, handle #6 is middle-bottom; it only resizes the height of the box [55].

Figure 18: The arrows are drawn from the selected box to the other boxes on its left and right side.
- JavaScript form validation: All input text boxes, for both Add stakeholder and Edit stakeholder dialog, have been validated with JavaScript regular expression.

- JavaScript in Evaluation tab: we have dynamically created HTML form with JavaScript to add fork option on evaluation tab.

4.1.2 Ajax

Ajax (Asynchronous JavaScript and XML) is a group of associated web development techniques. It is used on the client-side to generate asynchronous web application. Web application can send data to a server and receive data from server asynchronously without intervening with the display and performance of the current page. Ajax is a group of technology, not a single one. Combination of HTML and CSS can be utilized to markup and style information. The DOM is accessible with JavaScript to display dynamically, and to allow the user to interact with the information presented. Data can be restored using the XMLHttpRequest object. JavaScript and XMLHttpRequest object provide a method for transferring data between a browser and a server asynchronously and prevent reloading the full page.[22] There is a comparison between the traditional model for web applications and Ajax model in the following section.

**Classic web application model:** The User actions in the interface invoke an HTTP request back to a web server. The server does some operating such as retrieving data and then sends an HTML page back to the client. While the server is proceeding its process, the user is waiting. Therefore in every step of a task, the user waits some more. By investigating this model, the following question comes to our mind “why should the user see the application go to the server at all?”.

**Ajax web application model:** An ajax application removes the start-stop-start-stop procedure of interaction on the web by introducing an Ajax engine between a client and a server. The browser loads an ajax engine, written in JavaScript, instead of loading a webpage at the start of a session. Ajax engine is in charge of rendering the interface which the user sees, on the one hand, and communicating with the server on the user’s side, on the other hand. This engine allows the user to interact with the application asynchronously which is independent of communication with the server.
jQuery maintains at least five methods such as load(), $.get(), $.post(), $.getJSON(), $.getScript() and $.ajax() to make Ajax calls. The difference between each can be a little confusing. These methods are divided into low level and high level functions. We have chosen $.ajax() which is a low level Ajax function among those functions because it offers more functionality in some processes, such as error call back rather than higher level functions such as loading the data, get and post methods. By utilizing this function you will also have more control over your request [28].

In case of performing asynchronous HTTP request we need to set the key/value pairs which configure the Ajax request in the application as follow:
1. It is needed to choose the type of request to make “POST” or “GET” which we have chosen GET type for all HTTP request in whole application.
2. Set the URL key with a string containing the URL to which the request is sent.
3. Data object which should contain the data we want to send to the server. It is converted to a query string and will append to the url for GET requests. [29]
4. Success function will be called if the request succeeds. We have used the data returned from the server through the function to have the data which we require in the application. For example, since network tab is purely client side we required the data like stakeholders’ ID back when we send the data to store in database. Therefore, we achieved the data after storing information in database from the server by the argument of success function through Ajax request. You can see the code for Ajax request that sends to server (to add the stakeholder with the information which sends through request) below.

The GET type has chosen for the HTTP request. The URL is set with index.php, to which the request is sent. The data contains:
1. action: Since request is sent to server including the action’s key and value pairs, the related if statement will find on the controller to perform user’s request.
2. st_x and st_y: The x and y axis of the shape(stakeholder)
3. st_w and st_h: The height and width of the shape(stakeholder)
4. st_name: The name which was inserted by user in related text box on the dialog.
5. project_id and user_id which are accessible everywhere on the application by project session object and user session object. (With the aim of tracking both user and project to check which user works on which project).

Then if the action performed successfully, the stakeholder id will send back from success function. Otherwise, if the data sent back is equal to -1, as we defined in the controller, means the action was not performed successfully and something goes wrong.

4.1.3 jQuery

jQuery is a fast and compact JavaScript library which designed to simplify client-side scripting of HTML, event handling, animating and Ajax interactions for fast web development.[30][31] jQuery is designed to alter the way that you write JavaScript. jQuery was released in January 2006 and used by over 55% of the 10,000 most visited websites. jQuery is a free, open source software which is the most popular JavaScript library in use today.[30] The syntax of jQuery is designed to navigate a document easily and select DOM elements, create animations, handle events and develop Ajax applications. The modular approach to the jQuery library allows creating dynamic web pages and powerful web applications [30].

We have used lots of jQuery functionalities to develop collab application and to build a powerful client-side which will be explained in following.

- jQuery Selectors: They are one of the most important parts of the jQuery library and allow developer to select and manipulate the elements of HTML as a single element or a group of elements. In fact, if you cannot select an HTML element then you cannot work with it [38]. Selectors allow a programmer to reach the exact element/attribute that S/he wants from the HTML document. We have optimised jQuery selectors and applied more advanced selector combinations to our markup elements. There are lots of usages of jQuery selectors, some of them will be exemplified in following.

Network tab:
- Delete interest: In Edit Stakeholder and Interests dialog a user needs to click the cross image to delete the interest of the stakeholder, which is selected before by a user. To find the closest input to the cross image we have used closest() selector of jQuery library. attr() selector has also been used to reach the id of input, which is equal with the id of related interest that is shown in the textbox input.
Figure 19: Edit Stakeholder and Interests dialogue.

- Add interest: In add stakeholder and interest dialog user needs to press enter button on the keyboard, after inserting the interest in the input textbox, the user has to add it to the stakeholder. Since the Dom element (input textbox) is added to the dialog dynamically, the element should be selected and styled at the same time while it is being added to HTML document. Each input element which adds dynamically is attached to a div with a label and styled with CSS in the selector.

Evaluation and Relation tab:
There are tables in both evaluation and relation tabs. Once a user clicks a cell, a pad is opened; when the user unclicks the cell, the content of the pad with latest modifications shows up. Since the id of the HTML element (cell of the table which is clicked) that is equal to the id of the pad, we need to keep it in order to show the content of correlated pad whenever the cell is unclicked. The id of the selected cell is kept to show its content whenever the cell is unclicked by attr() method of selectors from jQuery library.

- jQuery effects:
jQuery library provides several techniques to add animation to a web page. This library brings a simple interface to do several models of amazing effects. These techniques include simple, standard animations which are commonly used and the ability are able to simplify complicated custom effects. jQuery methods allow us to apply frequently used effects with a minimum configuration. We have used several jQuery effect methods in our application such as fadeIn(), fadeOut(), slideDown(), slideUp(), etc which will be discussed in the following sections for each tab of the application.

Network tab:
- animate the height of dialog: By animate() method of jQuery library, the height of
adding stakeholder-interests and editing stakeholder-interests dialogs change when dialogs are opened. If the height of the dialog is less than the maximum height, as have been defined before, the height of the dialog increased by animate() method of jQuery library.

- Fade-in and fade-out error messages: The error message is shown in a label which is added to the HTML document dynamically by fadeIn method and will be disappeared after amount of time (2 second later) by fadeOut method. The usage of this method is to notifying users if they insert some invalid data in the input text boxes or want the data which does not exist in the database.

Evaluation and Relation tab:
- Zooming functionality: Once mouse enters to the cells of tables in evaluation and relation tabs, part of the content of the cell, which is hidden in default, will be shown by show method. When the mouse leaves the cell, it will be hidden and looks like the first view of the cell.

Conclusion tab:
- Show and hide contents: In conclusion tab, the important relation and evaluation cells, which are marked in both relation and evaluation tabs by users beforehand with the related stakeholder and interest, are listed in two tables. However, the content of the pads are related to the cells that are hidden in default. The .slideToggle() method of jQuery library, is used to show or to hide the pad content. This method animates the height of the pad content. It makes the content of the pad appearance revealed or concealed [29]. The pad content is initially hidden, it will be displayed and if either the show message or the hide message is clicked, the pad content will be hidden. The message (show or hide) on the cell, toggle between show and hide according to the action occurs.

- jQuery events:
  These methods are used to register behaviors when a user interacts with the browser and to further manipulate those registered behaviors [29]. We will discuss how these methods are used for each web page in the application separately.

Network:
- Animating the buttons: The animate() method performs the animation of a set of CSS properties. This method modifies an element from one state to another with CSS styles. By using hover method of jQuery library, whenever mouse enters the button, the images in the shapes of plus or cross will be moved top in three levels gradually. Whenever the hover() method is called by the action of the user (mouseenter to the button) the animate() method will occur. By the animate() method, the image will be moved in three scale to top with the defined speed in the method. When mouse leaves the button, the image will be back to the normal view.
- The difference between using bind and living events methods:
  The bind and live methods attach event handlers for selected elements of the HTML document, and defines a function to run when the events occur. The difference between bind and live is not always clear [43].
  Figure 21 helps the readers to visualize the DOM tree of an HTML document. For example, when user clicks a link, a click event on the link element fires, which triggers any function we bound to that element’s click event. Therefore the click event propagates up to the tree, broadcasting to the parent element and then to each ancestor element that click event was triggered on one of the descendent elements. Bind method is used to attach the event handler to the document so jQuery scans the document for all elements which are the same as the selector and bind the function to each of their click events. However, the live method binds event handler to the document element. Then, any time an event bubbles up to the document node, it checks to see if the event was the event handler and if the target element of that event matches the CSS selector. If both are true then the function will execute. Since there are DOM elements for Add and Edit dialogs, which are created dynamically to attach handlers to DOM elements that might not exist in the DOM, we use live method instead of bind method.
  Bind method binds handlers to the individual elements directly so it can not bind them to elements which are not created yet. For example, attaching keypress event handler to input text boxes, which have not existed yet by live method. However, Live method has several disadvantages which caused some bugs in the application. Delegate() method, one of the jQuery’s methods, could be a proper substitution for live() method; to have automatically handling dynamic data.
- Using keypress events methods:
To implement the interface of this project, we have tried to develop new features and functionalities. For instance, using Enter key to submit information from the client side to the server side, or creating new input elements (text boxes) instead of clicking buttons. In this case, to create a stakeholder and to add interests to the stakeholder in add stakeholder and interests dialog, keypress() event handler was attached to the input text boxes with live() method. Since text box elements, which show the stakeholder and its interests names to edit in edit dialog, have already been created while users open the edit dialog, input text boxes elements have been attached to keypress() event handler with bind() method.

![Edit Stakeholder and Interests](image)

**Figure 22:** Edit stakeholder and interests dialog.

- jQuery Dialog:
jQuery dialog is one of the scripts in our web application. Since Network tab on our web application requires intensive use of AJAX and JavaScript, We have found the custom jQuery dialog as one of the best option for our purpose. It is a floating window which contains both a title bar and a content area [37]. The dialog window can be resized, moved and closed by default. If the content length exceeds the maximum height of the dialog, a scrollbar will automatically appear. However, there is also inconsistency across different browsers but latest version of Chrome, Firefox and safari do not have any considerable problems with it. jQuery dialog is completely customizable via CSS which makes the application look much more professional with a custom title for each dialog. It will also reposition itself if a user resizes the browser window. A jQuery UI Dialog also contains several options which can be set with initializing the dialog. Those options, which are included in the custom jQuery dialog for our application, are as follow:
- auto open: If it is set to false, dialog will stay hidden until the dialog(“open”) is called on it.
- buttons: this option specifies which buttons should be displayed on the dialog.
- closeOnEscape: Defines whether the dialog should be closed when it has focus on it and the user presses the escape key (ESC).
- maxHeight: Defines the maximum height to which the dialog can be resized, in pixels.
- maxWidth: Specifies the maximum width to which the dialog can be resized in pixels.
- modal: If it is set to be true, other items on the page will be disabled. Dialog with modal properties which is set to be true creates an overlay below the dialog but above other page elements.
- title: Defines the title of the dialog.
- open: This event is triggered when dialog is opened.
- close: This event is triggered when dialog is closed.
- destroy: By using this method, the dialog functionality will be removed completely.

We use three dialogs in our Network tab to add stakeholder and interests, edit stakeholder and interests and delete stakeholder with all its interests. However, as it was mentioned in jQuery events, we have tried to avoid using buttons in the dialogs to submit information. In this case, by pressing enter key instead of using buttons:

Add stakeholder and interests dialog:
1. A user adds stakeholder, and creates new text box to add interest
2. A user adds interests by pressing Enter key, after inserting name of interest in new text box.

Edit stakeholder and interests dialog as shown in figure 22:
1. A user edits stakeholder name, interest name by pressing enter key.
2. A user deletes interest by clicking cross image beside interest name.
3. A user adds more interest by clicking add image which creates a new text box. A user should press enter key, after inserting interest name, to add new interest.

Delete stakeholder dialog:
First user should click the stakeholder graphical image on the canvas which S/he wants to delete. Then a user should press delete button on the page; delete dialog will appear so the user should click OK button to delete the stakeholder with all interests.

- jQuery plugin:
jQuery is the most popular JavaScript libraries and there are many sites which use jQuery for dynamic effects and Ajax functionality. jQuery allows developers to add methods to its library. By extending jQuery to jQuery plugins, a developer creates methods and components which can be reusable on any web page [44]. The developer’s code is encapsulated so there is less risk that S/he will use the same function elsewhere.
- Etherpad Lite Plugin: Since we have selected using etherpad lite functionality to handle collaborative issue for our project, there is a Plugin which is used in relation and evaluation tabs to create pads for the cells of tables and to get the content of the pads if they are not connected to the server any more. Therefore, Etherpad jQuery Plugin allows developers to add and access a pad from Etherpad in a web page easily. The Plugin creates the pad into a DOM element using iframes. It can also read the contents of a pad and write it to DOM elements. There are also many options and parameters such as the host, the port, the base URL of the pad and etc which could be set with other values instead of default values, listed in section 3.1.5 of the report. The Etherpad lite Plugin is part of the program as a JavaScript file which is called newEtherpad.js.

4.1.4 MySQL

One of the main reasons that a database is important in this project is that we have to save and retrieve data into and from database. In other words, choosing a proper database is one of the important parts in this project which it helps us to have more efficient and fast responded application. It is important to mention that we are not trying to say which database is better than the others instead, we are discussing about choosing a database that is more convenient to use according to the application needs.

In this thesis project, MySQL server version 5.5.16 is used in order to retrieve and store information. The reason of choosing it, is because MySQL can be install in all operating systems (OS independent), it supports all the functionalities that needed in order to implement ColLab, MySQL is fully compatible with Etherpad-Lite, it is free and finally it is one of the most important databases that are used in the real world for web applications.

MySQL is a relational table based database server which means in order to retrieve and save information we should create and use tables. These tables represent data that are used in ColLab. There are nine tables and one view that are communicating with ColLab core system. Before we start discussing about the database structure it is important to mention that most of the tables represent one to many relation and some of them represent many to many relation. The view in database used inner join syntax in order to combine multiple columns from multiple tables. Since view takes less space and it is sometimes faster than ordinary table in a database in order to show the data, then it should be more efficient to use view instead of table in the case of just representing the data.
The entire database structure is shown in figure 24.

In ColLab each projects are separated from each other and they are identified by unique ID. Project table in our database server contains three attributes:

1. project_ID: It represents a unique ID for each project. This attribute is auto incremented in order to keep uniquely identification of each project and it prevents conflicts between project IDs. Whenever a project gets created, a unique project ID will be specified for it and each project will be tracked by its ID. This attribute is a primary key for project table.

2. project_Name: Each project has a name in order to be easily identified by users. Project names can be the same since projects are identified by their unique IDs and the reason of non-unique project names is because of we don't want to limit users choosing unique names and limit the naming space.

3. project_Creator: Since in ColLab users are able to create projects with same names then it is sometimes hard to identify which projects with same name are belonging to which group. To solve this problem each project contains an attribute
which holds the user ID of user who is created this particular project. We will discuss about user ID in users table.

ColLab is a web-based application which means users are able to connect to ColLab from all around the world. In order to separate users and identify them we need a table to save their information, identify them and finally give them their resources whenever they ask for a service. Users Table is responsible for this purpose and it contains five attributes:

1. user_ID: Like a project, each user is identified by an ID that is given by database server automatically. User ID is a unique number for each user and it is auto incremented like project ID. Whenever a user registered itself into ColLab, a unique user ID will be specified to it and each user will be tracked in ColLab by its ID. This attribute is a primary key for users table.

2. email: Email attribute acts as username in the system. It is chosen by user during signup procedure and it should be identical for each user. The reason that ColLab uses email address instead of username is that because for future improvements, users are able to get email notification about their registration information, password reset and invite other members to join into a certain project. Email notification is not implemented in the current system and it will be implemented in the next version of ColLab (please see chapter 7 about future work).

3. password: Since ColLab is a web-based application and it is accessible from anywhere there should be have an authentication mechanism in order to increase security. Like other user-based applications, ColLab requires password whenever a user wants to login into the system. This field is encrypted by md5 algorithm.

4. firstName: This column holds first name of the user and it will show it for welcome message and also in the user page it is used in order to show project creator’s first name.

5. lastName: This column holds last name of the user and the main purpose of this attribute is to show project creator’s last name in user page.

ColLab is implemented in the way that each user is able to create as much as projects that he/she wants and it is also available that each project creator or project member is also able to invite new members to join into the project. In other words, each project can have several members and each user is able to be member of several projects. This scenario is typically represents many to many relation (MxN relation) in the database. In order to represent many to many relations in a database we need an extra table. This table is projectUsers table that is contain two attributes; one for project ID and other for user ID. Both attributes are primary keys. The project_ID attribute has dependency to project_ID attribute in project table as a foreign key and the user_ID attribute has dependency to user_ID attribute in users table as a foreign key. This table is also responsible for checking either a certain user is granted to edit the project elements or he/she is just able to access the project as a viewer (read only access).

1 For more information about user page please see section 5.1 about Collaborative ethical decision making demo.
In order to ColLab helps decision makers to make an ethical decision, each member of a project is able to create stakeholders and their interests, evaluation options and their child and write about the effect of each interest toward other stakeholders and also each interest regarding to other evaluation options. To achieve this goal we have to create and use corresponded tables in the database in order to use them in ColLab.

Stakeholder table is responsible to save and fetch stakeholders’ information that is used in ColLab. There are seven attributes that are used in stakeholder table:

1. **stakeholder_ID**: Each stakeholder is identified by its ID. The ID is unique and it is auto incremented. Whenever a stakeholder is gets created, the database will give ID to the stakeholder. This attribute is primary key for stakeholder table.
2. **stakeholder_Name**: Because ColLab simulates reality for decision makers then each stakeholder should have a real name. It is acceptable by ColLab to give a similar names for stakeholders except in the situation that stakeholders belong to the same project.
3. **project_ID**: It defines a certain stakeholder is belonging to which project by the project ID that is stored in stakeholder table. The project_ID attribute has dependency to project_ID attribute in project table as a foreign key in order to make a relation between stakeholder and project.
4. **stakeholder_x**: This attribute defines the x-axis of a certain stakeholder which it is a starting point of a rectangle for the stakeholder that is used in Network tab.
5. **stakeholder_y**: This attribute defines the y-axis of a certain stakeholder which it is a starting point of a rectangle for the stakeholder that is used in Network tab.
6. **stakeholder_w**: It defines the width of a stakeholder that is used in Network tab.
7. **stakeholder_h**: It defines the height of a stakeholder that is used in Network tab.

For each stakeholder that is defined within a project there are some interests that are belonging to that stakeholder. Interests are stored in interest table. This table contains three attributes:

1. **interest_ID**: Each interest is identified by its ID. The ID is unique and it is auto incremented. Whenever an interest is gets created, the database will give ID to the interest. This attribute is primary key for interest table.
2. **interest_Name**: This attribute represents the name of an interest. For each stakeholder, interests’ names must be unique but it is acceptable to use similar interest names within a project.
3. **stakeholder_ID**: It defines a certain interest is belonging to which stakeholder by the stakeholder ID that is stored in interest table. By accessing stakeholder ID it is easy to find which interests are belong to a certain project since project_ID attribute in stakeholder table tells us about the project that a stakeholder is belong to it. The stakeholder_ID attribute in interest table has dependency to stakeholder_ID.

---

1 For more information about Network tab please see section 5.1 about Collaborative ethical decision making demo.
attribute in stakeholder table as a foreign key in order to make a relation between interest and stakeholder.

Before we continue discussing about other tables that have been used in Collab it is important to take a short look at the Relations tab that is completely related to relPad table. Relations tab is a part of Collab which enable users to write about the effect of a certain interest towards a stakeholder. It consists of a table which each row corresponds an interest and each column consists of a stakeholder. Each cell in the table is a place to write about the effect of an interest towards a specific stakeholder. In the relations table each cell calls a pad. Each pad has its own unique address that is stored in relPad table in the database server. relPad stands for Relation Pads. For each pad there is a checkbox that identified either the content of a pad is important or not. If user checks the checkbox for each pad then the content of the pad will be shown in Conclusion tab. We will discuss about the Conclusion tab in the result chapter. Figure 25 shows a part of the Relations tab.

As it was mentioned in the last paragraph, the main responsibility of relPad table is remembering information about each pad. Since one interest can have several relation
pads then the relation between interest table and relPad table is one to many relations. relPad table consists of four attributes:

1. relPad_ID: This attribute identifies relation pad ID. It is unique and it is automatically generated by database server. It is also auto incremented that it prevents further conflicts between pad addresses. Whenever an interest gets created, ColLab will create n relation pads where n is the total number of stakeholders and also whenever a stakeholder gets created, ColLab will also create m relation pads where m is the total number of interests in the project. relPad_ID is primary key for this table.

2. interest_ID: Since each pad is related to an interest this attribute is used in order to bind a pad to its related interest. By accessing interest ID it is easy to find which pads are belong to a certain project because by accessing interest_ID we are able to capture its related stakeholder_ID and by accessing stakeholder_ID we are able to access project_ID. The interest_ID attribute in relPad table has dependency to interest_ID attribute in interest table as a foreign key in order to make a relation between relation pads and interests.

3. stakeholder_ID: According to the definition of pads in Relations tab, each pad describes the effect of an interest towards a stakeholder. Interest_ID points to the specific interest for each pad and stakeholder_ID attribute is responsible to point to the corresponded stakeholder that each pad is making the effect of an interest towards a stakeholder. In other words stakeholder_ID attribute is responsible to remember the column that each pad is pointing to it. The stakeholder_ID attribute in relPad table has dependency to stakeholder_ID attribute in stakeholder table as a foreign key in order to make a relation between relation pads and stakeholders.

4. important: As it was mentioned in the short overview of Relations tab, important checkboxes make users able to mark important statements (pads) in order to be visible in the Conclusion tab. To achieve this goal this attribute holds the states that either a checkbox is checked or it is unchecked. For checked situation this attribute puts one into the corresponding cell in relPad table and for unchecked option it puts zero into cell.

For now we have discussed about six out of nine tables that are used by ColLab. We have to take a look at the Evaluation tab that is a part of ColLab before we continue discussing about those three remaining tables because two of them are completely related to the Evaluation tab.

Like Relations tab (see figure 25), Evaluation tab is also use a table in order to enable users write their own statements about interests regarding to the evaluation options that they create them in the Evaluation tab. One of the big differences between Relation and Evaluation tabs is that in Relation tab users just need to write their own statements about interests towards stakeholders but in Evaluation tab, first they have to create evaluation options and then they can begin writing about the statements for interests regarding to the evaluation options. In other words, in Relations tab columns in the relations table are stakeholders but in Evaluation tab those are evaluation options. The other difference
between Relations and Evaluation tabs is that users are able to do fork function in order to create children of an evaluation option. By performing a fork, the statement that is written for the parent evaluation option will be copied as parent content of the children that is not editable.

Since the implementation of Relations tab and Evaluation tab is similar in general except the columns in the corresponded tables then it is natural that the nature of the table that should responsible for evaluation options should be similar to the table that is responsible to stakeholder table. Figure 26 shows a part of Evaluation tab in order to get a better picture about evaluationOption table.

![Figure 26: An overview of Evaluation tab and its components.](image)

evaluationOption table is responsible to save and fetch evaluation options' information that is used in ColLab. There are five attributes that are used in evaluationOption table:

1. **evaluationOption_ID**: Each evaluation option is identified by its ID. The ID is unique and it is auto incremented. Whenever an evaluation option is gets created, the database will give ID to the evaluation option. This attribute is primary key for evaluationOption table.

2. **evaluationOption_Name**: Because ColLab simulates reality for decision makers then each evaluation option should have a real name. It is acceptable by ColLab to
give a similar names for evaluation options except in the situation that evaluation options belong to the same project.

3. **Parent_ID**: This attribute identifies the parent ID of a forked evaluation option in order to fetch the parent content. If the evaluation option is not created by fork functionality then the parent ID becomes zero otherwise it defines the evaluation option that is the parent of the forked evaluation option.

4. **project_ID**: It defines a certain evaluation option is belonging to which project by the project ID that is stored in evaluationOption table. The project_ID attribute has dependency to project_ID attribute in project table as a foreign key in order to make a relation between evaluation option and project.

5. **totalParents**: Since forked functionality allows users to create children of children recursively without any limitation of creating children of a forked children then it is important that ColLab keeps track that how many parents is available for a certain evaluation option in order to show all the parents’ contents. This attribute tells to ColLab core system that how many iterations is needed to loop through the parents’ contents in order to show all of them. If an evaluation option has not created by fork functionality (it has not any parent) then the value of this attribute becomes zero otherwise it becomes a positive number corresponded to the total number of parents.

Like Relations tab, Evaluation tab is also uses pads in order to write statements about an interest regards to an evaluation option. Each pad in Evaluation tab has its own unique address that is stored in evalPad table in the database server. evalPad stands for Evaluation option Pad. For each pad there is a checkbox that identified either the content of a pad is important or not. If user checks the checkbox for each pad then the content of the pad will be shown in Conclusion tab. We will discuss about the Conclusion tab in the result chapter. An overview of Evaluation pad is shown in figure 26.

Since one interest can have several evaluation option pads then the relation between interest table and evalPad table is one to many relations. evalPad table consists of six attributes:

1. **evalPad_ID**: This attribute identifies evaluation option pad ID. It is unique and it is automatically generated by database server. It is also auto incremented that it prevents further conflicts between pad addresses. Whenever an evaluation option gets created, ColLab creates n evaluation option pads where n is the total number of interests and also whenever an interest gets created, ColLab creates also m evaluation option pads where m is the total number of evaluation options in the project. evalPad_ID is primary key for this table.

2. **evaluationOption_ID**: According to the definition of pads in Evaluation tab, each pad describes the effect of an interest regarding an evaluation option. Interest_ID points to the specific interest for each pad and evaluationOption_ID attribute is responsible to point to the corresponded evaluation option that each pad is making the effect of an interest regarding an evaluation option. In other words evaluationOption_ID attribute is responsible to remember the column that each pad is pointing to it. The evaluationOption_ID attribute in evalPad table has dependency
to evaluationOption_ID attribute in evaluationOption table as a foreign key in order to make a relation between evaluation option pads and evaluation options.

3. interest_ID: Since each pad is related to an interest this attribute is used in order to bind a pad to its related interest. The interest_ID attribute in evalPad table has dependency to interest_ID attribute in interest table as a foreign key in order to make a relation between evaluation option pads and interests.

4. important: As it was mentioned in the short overview of Evaluation tab, important checkboxes make users able to mark important statements (pads) in order to be visible in the Conclusion tab. To achieve this goal this attribute holds the states that either a checkbox is checked or it is unchecked. For checked situation this attribute puts one into the corresponded cell in evalPad table and for unchecked option it puts zero into cell.

5. totalParents: Since forked functionality allows users to create children of children recursively without any limitation of creating children of a forked children then it is important that CoLLab keeps track that how many parents is available for a certain evaluation option in order to show all the parents’ contents. This attribute tells to CoLLab core system that how many iterations is needed to loop through the parents’ contents in order to show all of them. If an evaluation option that is related to the pad has not created by fork functionality (it has not any parent) then the value of this attribute becomes zero otherwise it becomes a positive number corresponded to the total number of parents.

6. importantParent: This attribute is not used in CoLLab and the reason of creating this attribute is that if in the future the content of the parents will be either shown by selecting the corresponded checkboxes (not implemented in the current version of CoLLab) or hide them by de-selecting the checkboxes then this attribute can remember the state in order to either show or hide them.

The last table that is used in CoLLab is store table. According to the discussion in section 4.1, Etherpad-Lite is responsible for collaborative functionalities in CoLLab. All the contents of pads that are created by Etherpad-Lite is saved in store table. This table is based on key-value pair mechanism that in each row, a value describes the key part. This table contains two attributes:

1. key: It can be global author, read only pad name, a normal pad or revision history of a pad. The key attribute is primary key for store table.

2. value: The value attribute may varies according to the key part that is corresponded to value part. If key part is:
   - global author: The value part consists of the color that is chosen by user, user name and timestamp.
   - read only pad name: The value part is consists of the read only address of the pad. The read only address is the address of a pad that users are just able to see the content of the pad and they are not able to edit them.
   - normal pad: The important fields are the content of the pad, timestamp, the author that its ID is hashed and the password that have been used in order
to make a password protected tab. The password field is hashed and it is not usable by others.

- revision history of a pad: It consists of timestamp, author which its ID is hashed and change set data.

For more clarification about store table, figure 27 shows the result of two queries; one shows the global author and read only pad key-value pairs and second query shows a normal pad and revision history of two pads.

![Figure 27: The results of two queries of store table that shows key-value pair values.](image)

These nine tables that we have discussed about them in this section store and fetch the entire data that are needed in order to ColLab be able to make its necessary objects and shows information regarding to the request of its users. In section 4.1.6 we will also talk about how objects get created according to the corresponded data that are stored in the database. As it was mentioned earlier, there are nine tables and one view that are used in the database so before we close this section it is important to discuss about projectsUsers view. The main reason that projectsUsers is defined in the database is that to combine users, project and projectUsers tables into one table since the relation between users and project tables have many to many relations. Since we do not want to manually insert data into projectsUsers and we just want to represent the data that are stored from other tables then it is more efficient to use view instead of table (since view is faster and it takes less space than an ordinary table).

In the user page that a part of it is shown in figure 28 there is a table that enables users to view the projects that they are members of them. Users are able to open the projects and begin their modification or reviewing the projects by click on the open project buttons. The table that is shown in figure 28 is a part of projectsUsers view that represents the useful information which is gathered from three tables. We will see that whenever a user is logged into the system, ColLab reads the user_ID attribute from users table and by accessing to user ID, ColLab knows which user is member of which projects and by sending a query to usersProjects view, it draws the table that are shown in figure 28.
The results of two queries of projectsUsers view are shown in figures 33 and 34. If we compare figures 33 and 34, we will see exactly the same results in the Project name columns in both figures because the table that is used in user page (figure 41) extracts its data directly from the view.

```
mysql> select * from projectsUsers;
+-----------------+------------+--------+-------+-------+------------+--------+------------+
<table>
<thead>
<tr>
<th>Project ID</th>
<th>Project name</th>
<th>User ID</th>
<th>Email</th>
<th>First name</th>
<th>Last name</th>
<th>Creator ID</th>
<th>Created by</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Demo</td>
<td>1</td>
<td><a href="mailto:admin@admin.com">admin@admin.com</a></td>
<td>Admin</td>
<td>Adminisson</td>
<td>4</td>
<td>Admin Adminisson</td>
</tr>
<tr>
<td>2</td>
<td>Project Sam</td>
<td>2</td>
<td><a href="mailto:san@san.com">san@san.com</a></td>
<td>San</td>
<td>Eskandari</td>
<td>4</td>
<td>San Eskandari</td>
</tr>
<tr>
<td>3</td>
<td>Project</td>
<td>3</td>
<td><a href="mailto:mario@marco.com">mario@marco.com</a></td>
<td>Mario</td>
<td>Ramoni</td>
<td>4</td>
<td>Mario Ramoni</td>
</tr>
<tr>
<td>4</td>
<td>Project</td>
<td>4</td>
<td><a href="mailto:mario@marco.com">mario@marco.com</a></td>
<td>Mario</td>
<td>Ramoni</td>
<td>4</td>
<td>Mario Ramoni</td>
</tr>
<tr>
<td>5</td>
<td>Website</td>
<td>5</td>
<td><a href="mailto:san@san.com">san@san.com</a></td>
<td>San</td>
<td>Eskandari</td>
<td>4</td>
<td>San Eskandari</td>
</tr>
<tr>
<td>6</td>
<td>Website</td>
<td>6</td>
<td><a href="mailto:admin@admin.com">admin@admin.com</a></td>
<td>Admin</td>
<td>Adminisson</td>
<td>4</td>
<td>Admin Adminisson</td>
</tr>
<tr>
<td>7</td>
<td>Website</td>
<td>7</td>
<td><a href="mailto:san@san.com">san@san.com</a></td>
<td>San</td>
<td>Eskandari</td>
<td>4</td>
<td>San Eskandari</td>
</tr>
<tr>
<td>8</td>
<td>Website</td>
<td>8</td>
<td><a href="mailto:mario@marco.com">mario@marco.com</a></td>
<td>Mario</td>
<td>Ramoni</td>
<td>4</td>
<td>Mario Ramoni</td>
</tr>
</tbody>
</table>
+-----------------+------------+--------+-------+------------+-----------+------------+------------+
9 rows in set (0.00 sec)
```

Figure 29: The results of a query of projectsUsers view that combine all the stored data from users, project and projectUsers tables.

```
mysql> select * from projectsUsers where 'User ID' = 2;
+-----------------+------------+--------+-------+-------+------------+--------+------------+
<table>
<thead>
<tr>
<th>Project ID</th>
<th>Project name</th>
<th>User ID</th>
<th>Email</th>
<th>First name</th>
<th>Last name</th>
<th>Creator ID</th>
<th>Created by</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Demo</td>
<td>1</td>
<td><a href="mailto:admin@admin.com">admin@admin.com</a></td>
<td>Admin</td>
<td>Adminisson</td>
<td>4</td>
<td>Admin Adminisson</td>
</tr>
<tr>
<td>2</td>
<td>Project Sam</td>
<td>2</td>
<td><a href="mailto:san@san.com">san@san.com</a></td>
<td>San</td>
<td>Eskandari</td>
<td>4</td>
<td>San Eskandari</td>
</tr>
<tr>
<td>3</td>
<td>Project</td>
<td>3</td>
<td><a href="mailto:mario@marco.com">mario@marco.com</a></td>
<td>Mario</td>
<td>Ramoni</td>
<td>4</td>
<td>Mario Ramoni</td>
</tr>
<tr>
<td>4</td>
<td>Project</td>
<td>4</td>
<td><a href="mailto:mario@marco.com">mario@marco.com</a></td>
<td>Mario</td>
<td>Ramoni</td>
<td>4</td>
<td>Mario Ramoni</td>
</tr>
<tr>
<td>5</td>
<td>Website</td>
<td>5</td>
<td><a href="mailto:san@san.com">san@san.com</a></td>
<td>San</td>
<td>Eskandari</td>
<td>4</td>
<td>San Eskandari</td>
</tr>
<tr>
<td>6</td>
<td>Website</td>
<td>6</td>
<td><a href="mailto:admin@admin.com">admin@admin.com</a></td>
<td>Admin</td>
<td>Adminisson</td>
<td>4</td>
<td>Admin Adminisson</td>
</tr>
<tr>
<td>7</td>
<td>Website</td>
<td>7</td>
<td><a href="mailto:san@san.com">san@san.com</a></td>
<td>San</td>
<td>Eskandari</td>
<td>4</td>
<td>San Eskandari</td>
</tr>
</tbody>
</table>
+-----------------+------------+--------+-------+------------+-----------+------------+------------+
3 rows in set (0.00 sec)
```

Figure 30: The results of a query of projectsUsers view that shows all information about a certain user and relate it with all the data corresponded to all projects that a certain user is member of them.

The database itself is not directly accessible by ordinary users which mean the data that is stored in the database is not useful without accessing it. The procedures that describe how users are able to access the stored data and insert new data into the database will be discussed in sections 4.1.5, 4.1.6 and 4.2.

4.1.5 NodeJS

NodeJS is based on JavaScript server side technology and it acts as a web-based collaborative real time editor in ColLab. The software that we have used for web-based
collaborative real time editor is Etherpad-Lite that is based on NodeJS. By default, Etherpad-Lite uses dirtydb.db file as its own database in order to save and fetch data. The data that is save for Etherpad-Lite is described in section 4.1.4. Since dirtydb file that is predefined in Etherpad-Lite is limited to store just up to one megabyte of information though it is obvious that this type of database is not sufficient to use in ColLab. Fortunately Etherpad-Lite is capable to connect to other types of databases such as MySQL. In order to define MySQL server as a database server for Etherpad-lite there is a setting file to modify the default settings of Etherpad-Lite. This setting file calls settings.json and it is available under the Etherpad-Lite root directory. The settings.json file enables administrators to define their own settings to make Etherpad-Lite as efficient as possible.

There are two major settings that enable administrators to configure Etherpad-Lite according to their needs. One is database setting and the other one is network setting.

- **Database setting:** Since settings.json file is a json-file which based on key-value pairs, the database setting consists of four keys and their values:

  1. **User:** It determines the username that it has administrative rights to connect to the database server. Username must be created in advance into the database server and it should has sufficient privileges.
  2. **Host:** If the database server is located somewhere else than Etherpad-Lite then the address (URL) should be defined here. If the database server is located locally (not accessible via internet and it is installed on the local machine) then it is enough to write localhost which tells to Etherpad-Lite that the database server is on the same local machine.
  3. **Password:** The password that is used for User field in order to connect to the database server.
  4. **Database:** Defines the name of the database that Etherpad-Lite should connect to it and creates store table.

- **Network setting:** Like database setting, network setting is also based on key-value pairs. It consists of two fields:

  1. **IP:** It binds the given IP address to Etherpad-Lite. The default setting is 0.0.0.0. By modifying the default value administrators are able to run Etherpad-Lite in a specific IP address.
  2. **Port:** Like IP address, this field binds a specific port number to Etherpad-Lite. The default port number is 9001. By entering IP:Port# in the browser users are able to access Etherpad-Lite. For accessing Etherpad-Lite on the local machine it is enough to type http://localhost:9001 if administrator uses the default port number.

By creating a new pad, users are able to start working on the document that is created by Etherpad-Lite simultaneously. Figure 31 shows a Pad which was created by a user and there are two users that are working on a same document simultaneously. Each pad
consists of an area to write text, a chat mechanism that users are able to communicate with each other, revision control and buttons in order to format the text and other useful tools in order to help users working on a same document synchronously and simultaneously.

As you can see in figure 31, Etherpad-Lite shares documents by its own services. This is not the case that we want to use Etherpad-Lite in ColLab, instead Etherpad-Lite should give us its service within ColLab as a service that is used by ColLab. In order to achieve this goal there is a JavaScript file that is placed in between PHP-files which are responsible to produce web pages and Etherpad-Lite and the job of the Javascript file is creating iframes in order to open Etherpad-Lite service within the web pages. We will discuss about this mechanism in section 4.1.6 since it is one of the responsibilities of using PHP in ColLab.

Whenever an iframe gets created by the request of users, Etherpad-Lite will open a network stream in order to listen to the changes that are made by other users and it shows the changes of document to the users and also sending the changes that are made by other users to the Etherpad-Lite in order to broadcast to other users that are working on a same pad.

In order to start the Etherpad-Lite service on the server or local machine we just need to run a batch-file under Etherpad-lite root directory. This file calls start.bat.

Considering about Relations and Evaluation tabs (figures 29 and 30), each cell in the tables is a pad in Etherpad-Lite. Etherpad-Lite is also used in description and conclusion tabs. Some of the functionalities (e.g. chat, formatting text buttons and revision control)
are disabled in Relations and Evaluation tabs since those functionalities are not necessary to use in the cells.

In the next section we will discuss about PHP files that are heart of ColLab system. We will also discuss that how PHP as a core of ColLab connects all the modules together and brings ColLab services to the users.

4.1.6 PHP

PHP is the server side programming language that is used in ColLab. It is typically used with the Apache web server but it is also available for IIS [32]. Apache web server is an open-source web server that can run on any major operating system. It supports many server-side scripting languages and can interact with many database servers. The most common configuration is known as LAMP, which consists of Linux, Apache, MySQL and PHP [32]. Since Apache server and PHP are compatible with most of the operating systems then those are chosen to use in order to implement ColLab.

PHP is responsible for the following tasks:

- Communicating with database server.
- Creating classes and their objects.
- Creating web pages.
- Controlling user access and security.

**Communicating with database server:** As it was mentioned earlier, users are not able to communicate directly with the database. PHP is a medium between users and database server to create this communication channel. PHP works with MySQL database by MySQL API. An API (Application Programming Interface) provides a way for an application to work with other applications. To use the MySQL API, PHP provides three extensions [32]. These extensions are PDO, PHP’s mysqli extension and PHP’s MySQL extension. The advantages and disadvantages of each extension are listed below [32]:

- **PDO (PHP Data Object)**
  
  Pros
  
  - Is included with PHP 5.1 and later and available for 5.0.
  - Provides an object-oriented interface.
  - Provides a consistent interface that is portable between other database servers such as Oracle, DB2, Microsoft SQL Server and PostgreSQL.
  - Takes advantage of most new features found in MySQL 4.1.3 and later.

  Cons
  
  - Does not work with versions of PHP 4.x, 3.x, or earlier.
  - Does not take advantage of some advanced features found in MySQL 4.1.3 and later, such as multiple statements.

- **mysqli (MySQL improved extension)**
  
  Pros
○ Is included with PHP 5 and later.
○ Provides both an object-oriented interface and a procedural interface.
○ Takes advantage of all new features found in MySQL 4.1.3 and later.

Cons
○ Cannot be used with other database servers.

MySQL extension

Pros
○ Is included with PHP 3.x, 4.x and 5.x.

Cons
○ Does not take advantage of the advanced features found in MySQL 4.1.3 and later.
○ Is not under active development.

Considering pros and cons of each extension, we have decided to use PDO in order to PHP works with MySQL server since one of the big advantages of using PDO is that it is fully compatible with other database servers. Database connection variable gets created in database.php and as the result, PDO object placed in a variable calls $db. $db is a global variable which means by placing global $db; syntax in each function that is responsible to perform queries it is possible to access the database and perform queries. All the statements that are responsible for creating PDO object and performing queries are placed between try and catch statements. The reason of using try and catch statements is because of if something goes wrong within the database connectivity or performing queries then the entire application will not crash instead it produces error and the error is traceable in order to find and debug the error.

As it was mentioned in section 4.1.4, there are nine tables and one view which those are used in ColLab and those are not directly accessible by users. In this chapter it was mentioned that communication between users and database is performed by PHP-files. It was discussed projectUsers table makes many to many relations between users table and project table and projectsUser view fetch all information regarding to users and projects. It means those are usable either for collecting all projects that a certain user is member of them or collecting all users that those are member of a certain project. In order to perform queries to either save, fetch, edit or delete data there are eight PHP-files that are responsible to do that. Since ColLab consists of eight major entities then for each entity there is a PHP-file that is responsible to perform queries. Each file contains some functions that each function is responsible to perform a task with the database. It is important to mention each function may have several queries in order to perform a single task. Some tasks need to send back the result set of the queries to the caller in order to be completed. All the PHP-files that are listed in each entity send back the result set if it is needed. Those eight major entities and their corresponding PHP-files are listed below:

1. Project: All the information regarding to each project in ColLab is collectable by performing queries. It is also possible to add a new project or modify it by performing some queries in the database in order to save data into the database.
All these queries are divided into functions in order to perform their jobs. project_db.php is responsible to hold all such queries.

2. User: Like project_db.php, user_db.php is responsible to maintain and perform all the queries that are related to the users. Add new users, modify user information, add a user to a project and remove a user from a project into database are some tasks that are performed by the functions that those contain queries to do these tasks.

3. Stakeholder: All the queries that are related to the stakeholders are gathered in stakeholder_db.php file. It consists of functions that each function performs a task such as add, modify, delete a stakeholder into the database or fetch stakeholder information from the database.

4. Interest: interest_db.php is responsible to perform all the queries to the database regarding to interest entity in ColLab.

5. Relation pad: Sending queries and performing tasks regarding to each pad in Relations tab to the database are the jobs of relPad_db.php.

6. Evaluation option: evaluationOption_db.php is responsible to send queries to the database and send back the results of the queries regarding to each evaluation option in ColLab to the caller module.

7. Evaluation option pad: Sending queries and performing tasks regarding to each pad in Evaluation tab to the database are the jobs of evalPad_db.php.

8. Pad content: Figure 27 in section 4.1.4 show how Store table save information for Etherpad-Lite. As it is visible in figure 40, there is extra information regarding to each pad that is saved in the table. In order to extract just the content of each pad and perform the corresponded query ColLab needs a function to extract the content part of each pad. To achieve this goal there is a file that contains a function which gives a pad address as argument, find the correct row in the database, extract the content part of each pad and finally return it to the caller. This file calls etherpad_db.php.

All the queries that are mentioned above are done by prepared statements. Prepared statements improve performance when a statement is executed multiple times, because they allow the database server to reuse some of the work that it does when it prepares a statement [32]. Prepared statements also improve security because they can prevent most types of SQL injection attacks [32].

As a summary of communicating with database server, all the PHP-files that perform queries into the database server are shown in figure 32. If we say figure 32 is a module in CoLab, in this section we will see how these modules are connected together and bring all the functionalities in CoLab.
Creating classes and their objects: We have discussed about the eight major entities in ColLab. By analyzing further in entities we can discover that the last entity (Pad content) is a part of Relation and Evaluation pads entities. In other words, ColLab needs seven classes in order to fully perform its job. These classes are completely related to the information that is saved in the database server. For each entity of ColLab that is described above (except the Pad content) there is a PHP class that represents the data by creating the corresponding object that is an instance of one of these seven classes. The classes that are implemented in ColLab and their properties and major functionalities are listed below:

- Project class: This class is responsible to maintain all information about each project. When a project is opened by a user, a project object gets created by instantiating of Project class. The information about each project is gathered from the database server. Project class consists of five properties:
  - project_ID: This property holds project ID.
  - project_Name: This property holds project name.
  - project_creator: This property holds the user ID of the user who created this project.
  - stakeholder: This property is an array of Stakeholder classes. All the stakeholders that belong to a certain project are stored in the array.
  - evaluationOption: This property is an array of EvaluationOption classes. All the evaluation options that belong to a certain project are stored in the array.

Project class contains methods in order to add, modify and remove a certain stakeholder or evaluationOption objects.
- User class: This class is responsible to maintain all information about each user. Whenever a user is logged or signed up into the system a user object gets created by instantiating of User class. The information about each user is gathered from the database server. User class consists of six properties:
  
  - user_ID: This property holds user ID.
  - password: This property holds user’s password.
  - email: This property holds user’s email.
  - firstName: This property holds user’s first name.
  - lastName: This property holds user’s last name.
  - projects: This property is an array. Each element of the array consists of project name, project creator user ID and finally project ID. The purpose of using this property is that whenever a user enters into ColLab, the projects that this user is member of them can be easily represented in user page.

User class contains methods in order to add or remove a single project whenever a user creates a new project or unsubscribe itself from a project (this is different from the definition of setter functions thus it needs specific methods to perform these tasks).

- Stakeholder class: This class is responsible to maintain all information about each stakeholder in a project. Whenever a project that contains at least one stakeholder is opened or a stakeholder gets created by a user, the total number of stakeholders that are available in a project (when a project is opened by a user) or a single stakeholder object gets created or by instantiating of Stakeholder class. The information about each stakeholder is gathered from the database server. Stakeholder class consists of eight properties:

  - stakeholder_ID: This property holds stakeholder ID.
  - stakeholder_Name: This property holds stakeholder name.
  - project_ID: This property holds the project ID that this stakeholder belongs to it.
  - stakeholder_x: This property holds x-axis of the stakeholder which it is a starting point of a rectangle for the stakeholder that is used in Network tab.
  - stakeholder_y: This property holds y-axis of the stakeholder which it is a starting point of a rectangle for the stakeholder that is used in Network tab.
  - stakeholder_w: This property holds width of the stakeholder that is used in Network tab.
  - stakeholder_h: This property holds height of the stakeholder that is used in Network tab.
  - interests: This property holds all the interest objects in an array that are belong to this stakeholder. Each element of the array is an interest object.
Stakeholder class contains methods in order to add, edit or remove a single interest whenever a user creates, modifies or removes an interest which is related to the stakeholder.

- **Interest class**: This class is responsible to maintain all information about each interest in a project. Whenever a project that contains at least one stakeholder and one interest is opened or an interest gets created by a user, the total number of interests that are available in a stakeholder within a project (when a project is opened by a user) or a single interest object gets created or by instantiating of Interest class. The information about each interest is gathered from the database server. Interest class consists of two properties:
  - interest_ID: This property holds interest ID.
  - interest_Name: This property holds interest name.

- **RelPad class**: This class is responsible to maintain all information about each relation pad for Relations tab in a project. Whenever a user opens Relations tab in a project, MxN relPad objects gets created by instantiating of RelPad class where M is total number of stakeholders and N is total number of interests in the project. The information about each relation pad is gathered from the database server. This class is mainly used in Relations tab. RelPad class consists of five properties:
  - relPad_ID: This property holds relation pad ID.
  - interest_ID: Since each pad is related to an interest this property is used in order to bind a relation pad to its related interest. This property holds the interest ID of a certain relation pad.
  - stakeholder_ID: According to the definition of pads in Relations tab, each pad describes the effect of an interest towards a stakeholder. Interest_ID points to the specific interest for each pad and stakeholder_ID property is responsible to point to the corresponded stakeholder that each pad is making the effect of an interest towards a stakeholder. This property holds the stakeholder ID.
  - relPadContent: This property holds the statements that are written by users about the effect of an interest towards a stakeholder. In other words this property holds the content of a pad in Relations tab.
  - important: This property holds the state of important checkboxes in Relations pad. If a checkbox is checked then the value that holds in this property becomes 1 otherwise it is 0.

- **EvaluationOption class**: This class is responsible to maintain all information about each evaluation option in a project. Whenever a project that contains at least one evaluation option is opened or an evaluation option gets created by a user, the total number of evaluation options that are available in a project (when a project is opened by a user) or a single evaluation option object gets created or by instantiating of EvaluationOption class. The information about each evaluation
option is gathered from the database server. EvaluationOption class consists of five properties:

- evaluationOption_ID: This property holds evaluation option ID.
- evaluationOption:Name: This property holds evaluation option name.
- parent_ID: This property holds the parent evaluation ID for forked evaluation option. If an evaluation option has not parent then this property becomes 0.
- project_ID: This property holds project ID.
- totalParents: This property holds the total number of parents for forked evaluation option. If evaluation option is not created by forked function then the totalParents property becomes 0.

- EvalPad class: This class is responsible to maintain all information about each evaluation option pad for Evaluation tab in a project. Whenever a user opens Evaluation tab in a project, MxN evalPad objects gets created by instantiating of EvalPad class where M is total number of evaluation options and N is total number of interests in the project. The information about each evaluation option pad is gathered from the database server. This class is mainly used in Evaluation tab. EvalPad class consists of seven properties:

  - evalPad_ID: This property holds evaluation option pad ID.
  - evaluationOption_ID: According to the definition of pads in Evaluation tab, each pad describes the effect of an interest towards an evaluation option. Interest_ID points to the specific interest for each pad and evaluationOption_ID property is responsible to point to the corresponded evaluation option that each pad is making the effect of an interest regarding an evaluation option. This property holds the evaluation option ID.
  - interest_ID: Since each pad is related to an interest this property is used in order to bind an evaluation option pad to its related interest. This property holds the interest ID of a certain relation pad.
  - evalPadContent: This property holds the statements that are written by users about the effect of an interest regarding an evaluation option. In other words this property holds the content of a pad in Evaluation option tab.
  - important: This property holds the state of important checkboxes in Evaluation pad. If a checkbox is checked then the value that holds in this property becomes 1 otherwise it is 0.
  - totalParents: This property holds the total number of parents for forked evaluation option pad. If the evaluation option that is related to this pad is not created by forked function then the totalParents property becomes 0.
  - importantParent: It was mentioned in section 4.1.4 about evalPad table in database part that this property is not used in ColLab.

All the properties of these seven classes are defined as private properties thus setter and getter functions are implemented in order to modify these properties for each object that is
instantiated by each of the classes. All of the classes above also contain their constructors with initial values.

Whenever a user adds a project in ColLab, first it saves in the database and then the corresponded object gets created. There are two reasons that describe why we chose this strategy: The first reason is that because if something goes wrong in the CoLab system or in general since everything is stored in the database then it is very easy to just reload the page in order to fetch the stored data. Otherwise, by saving data into objects and then into database then there is a huge risk that the data get lost because it is save in the memory that is not sustainable and retrievable. The second reason is that since Collab is integrated with auto repair mechanism then if i.e. and interest gets created by a user and it saves in the database but something goes wrong in the system (i.e. lost electricity) and the corresponded tables and objects cannot be created after CoLab starts the service again, all the tables and objects that could not create by the problem that caused will be created automatically by ColLab. In other words saving the data first into database enables CoLab to rebuild itself if something goes wrong because the parent data is saved so CoLab knows about that some data is not created properly and it will created by itself but this functionality is not available when a data gets created first in the objects because the object itself will be lost by malfunctioning of the system.

As a summary of creating classes and their objects, all the PHP-files that containing classes are shown in figure 33. Figure 33 shows also relation between the PHP-files that are responsible perform queries to database and return the result and the PHP-files that are defining classes.

![Figure 33: PHP class files and those corresponded PHP-files that performs queries for each class.](image)

**Creating web pages:** In this chapter we have started discussing about the technologies that ColLab is implemented based of these technologies. We have discussed also how these technologies are used in order to ColLab performs its services. In order to create
connection between users and ColLab system we need web pages that users are not only be able to enter their data but also to view the data that is stored in ColLab. One of the jobs of PHP-files is create web pages. In general the web pages that are produced by PHP-files contain HTML tags, JavaScript and PHP statements. These pages are created by the web server which means whenever those are created then they cannot be changed and that is why we have mixed several client side technologies with server side technology in order to make web pages as flexible as possible. There are eight web pages that are created by PHP-files. Each of which contains simple or completely complicated elements that working together in order to either send the data that is entered by users or get the data from other parts in order to show the data that is saved in other parts. These eight pages are listed below:

- **signup.php:** This page is created in order to enable users to sign up and register them into ColLab and use ColLab functionalities. Sign up page consists of HTML5 tags in order to create HTML tags and use predefined regular expressions in order to check user inputs before it sends to the server. By using regular expressions that are used in HTML5, ColLab is able to stop sending incomplete or incorrect data that are entered by the users. It not only reduces the network traffic but also reduces the workload of the server because by missing this functionality unacceptable data is sent to the server side, the unacceptable data is processed and finally an error message should be return as unacceptable data entry by the users. By implementing this functionality all the checks are performed in the client side and the acceptable data will be sent to the server. This page is mainly contains input tags in order to complete a form which contain basic user information such as email address, password, first name etc. Since email address must be unique for each user, there is JavaScript function that gets the email address that is written by the user as an argument, get the most recent email addresses that are stored in the database by Ajax call and check if it is available for use or it is occupied before by someone else. If it is occupied then an error message is printed by jQuery. Whenever all the form fields are completed and all of them passed the requirements that are defined by the regular expressions then it is possible to submit the entire form by clicking on register button at bottom of the page. The result of pressing register button is registering the user into ColLab and logging in automatically into the system.

- **login.php:** This page is mainly contains HTML5 tags in order to create web page and regular expression checks for email and password entry fields. There is one PHP statement that is used in this page to show the login failure message in the case of username (email address) and/or password are not matched or email address is not registered in the system. Login.php is responsible to get user's email address and password and send it to the server.

- **userPage.php:** In this page users are able to view the projects that they are member of them and also create new projects. This page is just available for registered users and users must to login into ColLab in order to view this page. Whenever a user logged into the ColLab, this page will be showed automatically. Another way of accessing this page is to click on Home button on navigation bar.
The navigation bar will be discussed in chapter 5 but in general navigation bar consists of seven buttons. All of the buttons except invitation, transfer users to other pages that are available in CoLab. This page consists of HTML5 tags and PHP statements. HTML5 is used in order to create a table that shows all the projects that a user is member of them, buttons for opening corresponded projects, unsubscribe from projects or creating new projects and finally regular expression in order to validate the name of the new project. userPage.php is also consists of PHP statements in order to go through the projects that a user is member of them via user session object. User session object is created whenever a user is logged into CoLab and it is a copy of User object. As it was mentioned in User class, this class contains projects property which holds basic information about all the projects that a certain user is member of them. By using foreach statement and loop through this property we are able to represent the information about each project in userPage.php. This page is also contains JavaScript statements (jQuery) for sorting projects table.

- description.php: This page is responsible creating collaborative environment in order to users be able to write about the description of project. Since description.php brings collaborative functionality then it uses pad from Etherpad-Lite. The pad address that is created or used by this page begins with character d and follows by project ID. By using this method each pad is unique since it begins with character d that d stands for description and there are no other pads except description pads which begin with character d and since each project has its own project ID then combining d and project ID creates a unique address for description pad of a project. description.php consists of PHP statements to produce either iframe if the user is member of the project or show the content of the description in the read only mode if user is not member of the project. In both situations there is a dedicated div that either creates an iframe (if user is member of the project) or show content of page using jQuery statements. Both situations are described below:

  - User is member of the project: In this case it calls a JavaScript function that is stored in an external JavaScript file in order to create an iframe. The syntax is: 
    \[
    $("#description").pad({'padId':"d" . 
    $_SESSION["project"]-&gt;getProject_ID() ?>", 'userName':"&lt;?php echo 
    $_SESSION["user"]-&gt;getEmail() ?>", 'height':'89% ' });.
\]
  This is exactly the syntax that we promised to discuss in section 4.1.5. The arguments that are passed into the function are a part of the arguments that are available to pass according to discussion at the end of the section 4.1.5. As you can see the padID key uses “echo "d" . $_SESSION["project"]-&gt;getProject_ID()” as its value in order to identify its related pad from Etherpad-Lite. The project ID is read from the project session object by calling getProject_ID function which implemented in project.php class. User’s email address is passed to Etherpad-Lite in order to Etherpad-Lite be able to identify users.

  - User is not member of the project: In this case .html function in jQuery is called in order to show the content of the description pad. The syntax is
getGeneralPadContent("d" . $_SESSION['project']-getProject_ID()); where getGeneralPadContent is a function that is implemented in index.php in order to call getPadContent function that is implemented in etherpad_db.php file which is responsible to perform query to store table in the database, remove extra information that is stored in the store table of each pad and returns the actual pad content.

- network.php: This page is almost pure client side based on JavaScript, Ajax, jQuery and PHP. Users create stakeholders and interests, edit and delete them on the network tab. The stakeholders are drawn as graphical objects on the canvas/HTML5 by using a JavaScript library which was discussed in section 4.1.1.

- All PHP statements which we have used in network.php:
  - In fact, if user opens a project, the project session is created. All the stakeholders’ and interests’ objects of the project will store into the project session object. In this case, they are reachable anywhere in the application that we need. Whenever user clicks the network tab or reload the page, all stakeholders with its all attributes are read from the project session object. In this case, by using foreach statement, we loop over all stakeholders objects, which are belong to the project, with their all information and add them with JavaScript function (addRect()) on the canvas. It needs to mention that, we also loop over the interests of the stakeholder by using another foreach statement. So that, each interest of the stakeholder push as an array with two fields (interest id and interest name) oneInter[], to the main array of interests inter[], as one of the information of the stakeholder.

- Here, we illustrate how to use all functionalities (JavaScript, Ajax, jQuery and PHP) together for adding stakeholder on network tab:
  - For instance, When user wants to add a stakeholder with its interests, s/he should press the add stakeholder button. The add dialog opens, by using $.Ajax function data is sent to the server. If it is added successfully in the database, the success function of Ajax will send the id of stakeholder back. When the dialog is closed, the id of the stakeholder will validate. If it is valid (not equal with -1), the stakeholder with all attributes will add to the shape array (box[]). The stakeholder’s attributes are:
    1. x and y axis: initialize by arithmetic method which produce a number randomly.
    2. Width and height which is initialized by 40 in default.
    3. Name of stakeholder which inserted by user.
    4. Stakeholder’s id.
    5. The array of interests of the stakeholder: The data of interest also send by $.ajax function to server. The ID of each interest validates. If it is valid, the interest pushes to array with its id and name.
    Therefore, the stakeholder (with the shape of rectangular) with its name will draw on the canvas by addRect() function. Furthermore, project_id and user_id are respectively attributes of project object and user object (which are PHP objects).
They are accessible on each page of the application by project session object and user session object. We always send these data by ajax request to track both user and project. According to the example, all these functionalities implemented with JavaScript, Ajax jQuery and PHP on network.php page.

- relations.php: In this page users are able to write statements about the effect of interests toward stakeholders. relations.php is mainly based on PHP and jQuery statements. This web page contains a table that each row belongs to an interest and each column belongs to a stakeholder. The table and its components are created by PHP statements that generate HTML tags. Since a project contains its stakeholders and each stakeholder contains its interests then by using foreach statement ColLab is able to read all information about stakeholders and their interests that are belong to a project by looping over Stakeholder and Interest objects that those are stored in project session object. Whenever a user opens a project, all the stakeholders and their interests objects get created and stored into project session object so those are reachable any time when those are needed. Pads get created after drawing relations table by looping over stakeholder and interest. The content of each pad is read by considering of a pad is pointing to an interest and a stakeholder. The address of the pad which is stored in Etherpad-Lite begins with character r follows by relation pad ID. Since r is just used in Relations tab and it does not used in other parts of ColLab and the address of each relation pad is unique and stored in relPad table in the database then ColLab guarantees that each cell in the relations table is related correctly to its stakeholder and interest. Whenever a user opens Relations tab, the content of each pad reads directly from the database in order to fetch the most recent data that is written by other users. Like description.php, there are two situations that can occur according to the user that viewing Relations tab:

  - User is member of the project: In this case user is able to click on each cell. By clicking on each cell the corresponded iframe gets created and user is able to communicate directly with Etherpad-Lite by opening a network stream. Whenever an iframe gets created, the collaborative environment that is available by Etherpad-Lite allows user to work collaboratively by other members of the project. If user clicks on the other cell of relations table then the last connection will be terminated and the communication between user and the pad of Etherpad-Lite gets closed and the last iframe will be closed. Then new network connection between user and Etherpad-Lite gets created, new iframe that is related to the new cell which user is clicked on it will be created and user is able to communicate with Etherpad-Lite without overwhelming Etherpad-lite since the old connection that is listening and sending to the old address is closed. By opening just one connection at the time and closing old connection ColLab does not open too many connections between user and its server and by using this idea ColLab reduces its network traffic. This mechanism is implemented by delegate
functionality that is implemented in jQuery [33]. The new states of important checkboxes are sent by Ajax GET method whenever those get changed.

- User is not member of the project: In this case user is able to view the content of pads without ability to edit them. The content of each pad is read from relPadContent property of relPad object.

In both situations above users are able to see maximum 100 characters in each cell in order to minimize the height and width of the Relation tab. By implementing hovering function they are able to move their mouse on each cell in order to see the entire content of each cell.

- evaluation.php: In this page users are able to write statements about the effect of interests regarding evaluation options. evaluation.php is mainly based on PHP and jQuery statements. This web page contains a table that each row belongs to an interest and each column belongs to an evaluation option. The table and its components are created by PHP statements that generate HTML tags. Since a project contains its evaluation options and interests then by using foreach statement ColLab is able to read all information about evaluation options and interests that are belong to a project by looping over EvaluationOption and Interest objects that those are stored in project session object. Whenever a user opens a project, all the stakeholders, their interests and evaluation option objects get created and stored into project session object so those are reachable any time when those are needed. Pads get created after drawing evaluation option table by looping over stakeholders, interests and evaluation options. The content of each pad is read by considering of a pad is pointing to an interest and an evaluation option. The address of the pad which is stored in Etherpad-Lite begins with character e follows by evaluation option pad ID. Since e is just used in Evaluation tab and it does not used in other parts of ColLab and the address of each evaluation option pad is unique and stored in evalPad table in the database then ColLab guarantees that each cell in the evaluation option table is related correctly to its evaluation option and interest. Whenever a user opens Evaluation tab, the content of each pad reads directly from the database in order to fetch the most recent data that is written by other users. Like relations.php, there are two situations that can occur according to the user that viewing Evaluation tab:

- User is member of the project: All the discussion about relations.php that is related to if user is member of the project is applicable to this situation however users in Evaluation tab in this case are able to add, rename and delete an evaluation option. They are also able to use fork functionality that creates a child of an evaluation option. Whenever a user wants to add an evaluation option, the name of the evaluation option checks by regular expression that it does not begin with white spaces on the client side. If the name is acceptable then it sends by a form and Evaluation tab will be refreshed. In order to rename an evaluation option after checking the new name that does not begin with white spaces, by using GET method of Ajax new name will be saved in the database. For deleting an evaluation option
there is a button that submits a form in order to delete an evaluation option. A re-confirmation appears by JavaScript in order to ask if user really wants to delete the evaluation option. Fork functionality is totally performed by JavaScript and jQuery. Whenever a user clicks on fork button, a JavaScript function gets called by two arguments. The first argument is evaluation option ID which will be saved as parent evaluation option ID and the second argument is the total parents of the current evaluation option which will be incremented by one and it will be saved as total parents for new valuation option that is created by fork functionality. The name of the forked evaluation option reads from dialog API by jQuery. The name checks by regular expression that it does not begin with white spaces. All the information regarding to an evaluation option that is created by fork functionality is sent by a form. The form itself is created by JavaScript because if it is implemented by HTML tags then it is not possible to include dialog in order to enter the name and it is not either possible to check if the name is unique or not. Whenever the name is unique and it does not begin with white spaces then the form will be submitted to the server and the corresponding information will be saved into the database.

- User is not member of the project: In this case user is able to view the content of pads without ability to edit them. The content of each pad is read from evalPadContent property of evalPad object.

In both situations above users are able to see maximum 100 characters in each cell in order to minimize the height and width of the Relation tab. By implementing hovering function they are able to move their mouse on each cell in order to see the entire content of each cell.

- conclusion.php: In this page users are able to write their conclusion about the entire project collaboratively and also view important statements from Relations and Evaluation tabs. The important statements are gathered from the important checkboxes that are available in both Relations and Evaluation tabs. conclusion.php consists of two major parts:
  - Like description page it contains a collaborative environment in order to users write their own conclusion simultaneously. The entire scenario to create iframe is exactly the same as the scenario that is discussed in description.php except the pad address that is begins with character c instead of d. The collaborative environment is just available for project members and non-members can just view the content of the conclusion without ability to edit it.

  - There are two tables that are responsible to show important relations and evaluations. In order to show the important relations and evaluations, ColLab iterates over all the stakeholders and interests that are defined in the project and then by checking if the value of important property of each relPad and evalPad object is equal to one then it shows the content of it.
As a summary of this part, all the web pages in CoLab are shown in figure 34. All these web pages are the answers of HTTP requests by HTTP response.

![Web browser](image)

**Figure 34: Web pages that represent information by HTTP response to users.**

**Controlling user access and security:**

In this part we are going to discuss about two different aspects which are implemented in CoLab. These aspects are users access control and security.

- **User access control:** Each project in CoLab has its own members. Whenever a user creates a project he/she becomes automatically member of the project. Each project member is able to invite new users to the project. When a user gets invited to a project, he/she becomes a new member of the project. Project members are able to write description of the project, add/edit/delete stakeholders, interests and evaluations options, modify cell contents in Relations and Evaluation tabs, change the states of important checkboxes in Relations and Evaluation tabs, write the conclusion of project, invite other CoLab users to become members of the project, chat with other members of the project and finally use revision history of description and conclusion parts. CoLab is not restricted users to be a member of the project in order to view each project. There is a public address for each project that enables both non-members of the project and non-registered users in CoLab be able to view the project. By accessing public address all the abilities which are described above for project members are disabled and in this case users are just able to view whatever project members are done in the project. If a member of a project opens the project by accessing public link, since the user is a member of the project then he/she is able to modify the project. CoLab automatically gives all permissions to project members whether the user opens the project via user page.
or it was opened by accessing the public link. The mechanism that gives ability to ColLab in order to check whether a user is a member of the project or not is simple. Whenever a user logged into ColLab, he/she gets his/her user ID from users table in the database according to the email and password which are entered during login phase. The user ID is saved in the user session object. When a user opens a project by accessing public link he/she gets zero as his/her user ID. Since user IDs is auto incremented and these are began from one then a user that gets zero as user ID cannot be a member of any project. When a project is opened whether by opening it via user page or by accessing the public link, the project ID will be saved in project session object. There is a function in project_db.php that takes user ID and project ID as arguments and returns whether this user ID is the member of the project ID by returning TRUE or not by returning FALSE. This function sends a query to the projectUsers table and if the result set that returns from the database server is non-empty then it means user is the member of the project otherwise (the result set is empty) user is not member of the project. The query that is sent to the database uses prepared statements that is “SELECT project_ID FROM projectUsers WHERE user_ID = :user_ID AND project_ID = :project_ID;” where :user_ID is bounded to the user ID as an argument and :project_ID is bounded to the project ID that is also the other argument of the function. In each web page which some functionalities are disabled for non-members there is PHP statements that calls a function in index.php (we will discuss about index.php in section 4.2). The PHP statements are simple and these are if-else statements: ```php if(checkIfEditor($_SESSION['user']-getUser_ID(), $_SESSION['project']->getProject_ID())) { ?>. The functionalities that should be available for project members is surrounded in if closure and if there is a simplified version of a certain functionality that is implemented for non-members is surrounded in else closure. checkIfEditor function is implemented in index.php and it calls another function in project_db.php that is isProjectMember and it is described above.

- **Security:** Since ColLab is a collaborative space, accessible from anywhere, we need to make sure to have at least some login requirements and input validation. In order to each user access their own projects, they must login to ColLab. During login phase, a user must submit his/her email address and the password that belongs to the email address. Email address and password checks with database and if both of them are correct then user redirects to user page. Submitting username and password is performed by POST method in order to hide them in the address bar. Password field is encrypted in the database and it is not retrievable by ColLab.

### 4.1.7 HTML5/Canvas and CSS3

- **HTML5:** As we mentioned in section 3.2, HTML5 is a markup language for structuring and presenting content for the World Wide Web, and it is a core technology of the internet [14]. It is the fifth revision of HTML standard, which is still under development. Its main aims have been to improve the language with support for the latest multimedia while keeping it readable easily by human and consistently understood by computers and...
devices such as web browsers. HTML5 is a response to the observation that the HTML and XHTML are commonly used on the World Wide Web, a mixture of features introduced by several specifications, along with those introduced by software products such as web browsers. Such products are produced by common practice, and many syntax errors in existing web documents [14]. It is also an attempt to define a single markup language that can be written in either HTML or XHTML syntax. Html5 adds particularly many new syntactical features. These contain the new <video>, <audio> and <canvas> elements, as well as the integration of scalable vector graphics (SVG) content. These features are designed to make include and handle multimedia and graphical content on the web easy without any need for particular plugins and APIs [14]. In the following, we will discuss about how we used some features of HTML5 in our web application.

- Canvas in HTML5: The <canvas> tag in HTML5 is a really useful tool which allows a developer to draw graphics using a script like JavaScript to create the image. When using the canvas tag, the first thing to do is adding the tag to the HTML5 document.

<canvas id="canvas"></canvas>

It will create a blank space on the web page, as we have not placed any information yet on the canvas, yet. The canvas tag has one attribute, id which gives the canvas a unique name. We did not specify the height and width attributes of the canvas in the tag because it should not be static in our program. In this case, we specify height and width of the canvas in the script to make things relative to the canvas' current width/height. As it was mentioned, the canvas tag uses JavaScript to draw directly on the web page. Therefore, in order to get anything to show up on the canvas, we need to add a script to our document [35]. The script, which is added to the document to make anything show up on the canvas, was discussed in section 4.1.1, explaining how the graphical objects are drawn on the canvas in network tab.

- Validating forms with HTML5: The option of using pure HTML for form validation was unbelievable until recently. However, there were all kinds of JavaScript plugins over these years, which the goals of developing them were achieving something similar. The simplest change is to mark a text input as ‘required’ in the form. This informs the web browser that the field should be considered mandatory [47]. Different browsers may mark the input box in some way displaying a warning or preventing the form from being submitted if the field has no value. We have used HTML5 form validation in web pages such as login, sign up and evaluation pages. However, we have used JavaScript for form validation in Network tab, too. For example, in evaluation tab, input box for inserting option name and fork name using HTML5 form validation which is illustrated with the code in the following.

```html
echo '<input type="text" name="optionName" class="addOption" required placeholder="Option name" pattern="[^ \t].*" title="First character cannot be white space."/>';
```

The text boxes in the form have been marked as ‘required’ which doesn’t allow a user to submit the form with empty field. By adding the pattern attribute (pattern="[^ \t].*") to the tag, the user would not be allowed to insert white space as a first character in the text input.
CSS: CSS (Cascading Style Sheets) is a language used to make the web pages pretty [42]. It is a most popular application to style web pages which are written in HTML and XHTML markup languages. CSS is designed to enable the separation of document content (written in HTML or other markup languages) from document presentation, containing elements such as the layout, colors, and fonts [39]. This separation can improve accessibility of content, provide more flexibility, enable multiple pages to share formatting and reduce repetition in the structural content (like allowing the tableless design). CSS can also allow the same markup page to be presented in different styles for different rendering methods like on screen, in print, by voice. It can also be used to make the design of the web page differently depending on the screen size or device which it is being viewed. CSS3 is a new version of CSS with many more features and supports a rich interface and extra functionalities such as Text shadowing, opacity, border radius and multiple backgrounds [36]. We have chosen CSS3 features to style pages of our web application because of many reasons which is mentioned in the following parts:

1. It provides faster web page response times because of the reduction of images which required for things such as text effects and web buttons for user Interface of our web application [34].
2. It reduces our dependency on JavaScript for visual presentation purposes such as animation effects, not only for having less code which is always a good thing but also resulting to better web page performance [34].
3. It brings less reliance on expensive graphics software such as Photoshop [34]. For example, in evaluation tab, since we use button tags to add and to delete options by submitting forms to the server, we put images which styled by CSS instead of using Photoshop inside the buttons.

The new features of CSS3 should be treated like any other web technology which we work with; besides it should always support user-centered design and the goal of the website. For instance, CSS3 brings better support for media queries, and we can develop flexible and responsive web designs which enhance the user’s experience depending on what browsing device the users apply. That is a great improvement, which introduces plenty of new concepts in our traditional site-building methodology. By using new CSS3 selectors, we could have much better ability of targeting HTML elements and accomplishing animation effects without using JavaScript or jQuery library. CSS3 is not only used for the things such as fancy text shadows and gradients which are part of the surface of it. It also optimizes the user experience and is able to produce better websites in a more efficient way [34]. Figure 35 also shows one of the benefits of using CSS3 features on the website.
Each page of our web application has a unique CSS file such as relations.css, evaluation.css, login.css and etc., which are in separate folder with the name of CSS. The purpose of using unique CSS file for each page is because of the presence of certain elements on each page, which is different from another page. We have also intended to prevent one CSS file from getting too large. There are also two extra CSS files for network tab, which is to style dialogs on Chrome browser, and table sorter plugin. We had a problem to adapt styling jQuery dialogs on Chrome which was different from Firefox. In this case, we decided to have a different CSS style whenever the browser is Chrome for network tab.

4.2 MVC pattern

Whenever an application grows larger, it becomes more difficult to keep the code organized and also more difficult to maintain that code later on. It is important to keep the code well organized otherwise it becomes more difficult to code, test, debug and maintain large applications. To fix this problem, professional web developers commonly use a programming pattern known as the MVC (Model-View-Controller) pattern. The MVC pattern is commonly used to structure web applications that have significant processing requirements. That makes them easier to code and maintain [32].

The model consists of the PHP files that represent the data of the application. The view consists of the HTML and PHP files that represent the user interface of the application and finally the controller consists of the PHP files that receive requests from users, get appropriate data from the model, and return the appropriate views to the users [32]. Figure 36 shows the MVC pattern in general.
Considering figure 36, each arrow has a number. Each number shows the sequence of processing sequence whenever a user requests a web page follows be the scenario that is described below:

1. Web browser sends a request to the controller in order to view a web page by sending HTTP request.
2. The controller takes HTTP request that is sent by web browser but the web page that is requested by web browser contains data that the data should be collected from the database. The data that is needed is requested from model by the controller.
3. Model sends a single query or couple of queries to the database in order to collect the requested data that is sent by the controller.
4. The database collects the requested data that is sent by model by sending result set of the queries to model.
5. Model takes the result set that is sent by the database, delivers corresponded classes and send back collected data to the controller.
6. The controller creates objects which are instantiated from the classes which are available in the model and also the other data that are collected from the database by model and sends them to the view.
7. The view represents all collected objects and data which are sent by controller in a web page and send it to user that requested the web page by HTTP response.

Now we have clear picture about the steps from requesting a web page by a user to representing the web page by server according to MVC pattern. The implementation of ColLab follows MVC pattern because of usefulness using MVC pattern. The model, view and controller parts of ColLab are described below:
- **Model**: The model part consists of PHP files which are completely discussed in “Communicating with database server” and “Creating classes and their objects” parts.
- **View**: The view part consists of PHP files which are completely discussed in “Creating web pages” part.
- **Controller**: contains several if and else if statements which perform actions according to users’ requests. Each action that is performed by controlled can have several other actions in order to perform users’ requests. Users’ requests are submitted by either GET or POST methods. In general, index.php is responsible to create, modify and delete objects which are instantiated from their classes, create project and user session objects in order to collect and maintain the data that is needed for each project and user. Since web is stateless and it does not remember which user is working on which project, by creating session objects and put information about them in corresponded session object then tracking both project and user is applicable and meaningful.

Considering figure 36 that shows the definition of MVC pattern in general, all the PHP files which are used in CoLLab are divided into three parts namely Model, View and Controller according to their role. As a summary all of them are shown in figure 37 and for more clarification those are separated into Model, View and Controller parts which shows which PHP file is belong to which part of MVC pattern.

![Figure 37: PHP files which are used in CoLLab according to MVC pattern.](image-url)
4.3 User Experience Survey

The thesis project is divided into two main parts. For the first phase of the project, we have focused on implementation of colLab web based application. Evaluation of the system, provided by analyzing the answers of students to the online survey, was the second part which is the research part of the thesis. The resulting software has been evaluated from a User Interface point of view and the software’s functionality on a group of students performing an ethical analysis with help of the tool. We made 27 questions which were based on two types of open-ended and close-ended questions. The close-ended questions were made to evaluate the user experience and functionality of the system. We have made first 23 questions of our survey based on flexibility in use, perceptible information, simple intuitive use, low physical effort, user satisfaction, ColLab functionality.

To answer the first 23 questions, which are closed-ended questions, participants should select the answer from a set of scaled items between strongly agree and strongly disagree. We decided to use Likert Scales to scale responses to the closed-ended questions in our questionnaire. A psychometric response scale primarily used in questionnaires to obtain participant’s preferences or degree of agreement with a statement or set of statements. Likert scales are a non-comparative scaling technique. Respondents were asked to indicate their level of agreement with a given statement by way of an ordinal scale [51]. We scaled the responses to our closed-ended questions as a 5-point scale ranging from “Strongly agree” on one end to “Strongly disagree” on the other with “Neither Agree nor Disagree” in the middle which is shown in figure 38.

Figure 38: Sample scale used in Likert scale questions [51].

The rest of the questions (4 questions remained) are open ended questions which are useful to provide in-depth information and allow us to probe deeply into issues.

- Why we decided to use web survey?
  A survey is a method of gathering information from a number of individuals who usually participate through a system of standardized questions. However, surveys are in several forms and serve a variety of purposes and share certain characteristics. Surveys can be divided into two categories: the questionnaire and the interview. Sometimes, it is hard to define the difference between a questionnaire and an interview. For instance, some people think that questionnaires always ask closed-ended questions while interviews always ask
open-ended ones. However, you can see lots of questionnaires with open-ended questions and there are also a series of closed-ended questions asked in an interview [48]. The web survey is a costing effective way of administering a survey which allows us to collect large amounts of information without having to paper supplies or postage, and does not require separate data entry for responses to be processed. The increase of using web based surveys is due to the widespread availability of computers. The ability to receive a questionnaire and complete it at home, at the university or in the office on a computer is very convenient for most people. Information can be collected and processed only in a few days [49].

We used LimeSurvey to design a web survey. LimeSurvey is a web application which was installed in the Uppsala University’s server. Users can use rich text in questions and messages, while using a rich text editor. Once the survey is finalized, the user can activate it, make it available for participants to view, and answer. Questions can be imported and exported through the editor interface. Questions are added in groups in which we divided 23 first questions in two groups based on user’s experience and functionality of the system. LimeSurvey also provides several advanced features such as basic statistical and graphical analysis of survey results [50]. We have sent the notifications of fulfilling the survey to the students, who have taken IT Ethics and Organization course and used colLab tool to do their assignments, three times. We are thankful for the help from the students who completed our survey, without whom this study could not have been done.

All questions which are available on the web survey are listed in the appendix at the end of the report.
5. Result

In this chapter first we will show the functionalities of ColLab as the result of implementation phase that it was discussed in chapter 4. After representing the functionalities of ColLab we will discuss about the study that is performed by analyzing the answers that are made by students who worked with ColLab via an online survey.

5.1 Collaborative ethical decision making demo

In this section we are going to show how ethical decision making about the web designer scenario can be done as it was promised earlier. We start the scenario from scratch which means we will start from user registration and continue showing each step that is performed by ColLab in order to show how ColLab helps users to make their ethical decisions.

Whenever a user requests ColLab application via web, the user automatically calls index.php. Since user is not logged into ColLab then he/she redirects to login.php page.

Users are able to register themselves by clicking on Create account button. Whenever the user clicks on Create account button he/she sends a request to index.php (controller) in order to visit signup.php by sending HTTP request to the controller. The HTTP response that is the answer of the user's request is Sign up page.

As it was mentioned in chapter 4, Sign up page contains regular expression in order to validate user inputs. Regular expressions check if First name and Last name fields do not start with white spaces, Email field contains accepted email address format and the email address is not taken by other users and finally Password and Retype password fields have the same values. If these requirements are passed then by clicking on Register button users are able to register them in ColLab otherwise sign up form is not submitted to server until all the required fields passed requirements.

Whenever a user registered itself or signed in into ColLab, he/she redirects automatically to user page (Home tab) by controller. All users automatically become member of Demo project. Demo project is created in order to users be familiar how ColLab works. In user page each user is able to create new project, open existing projects that the user is member of them and also unsubscribe himself/herself from projects. User page is shown in figure 39.
As it was mentioned above, users are able to create new projects in user page. There is no limitation choosing the name of project except it should not begin with white spaces. If the new of the project does not begin with whitespace and user clicks on Add project button then new project gets created, it gets its unique ID from database server and the user page will be refreshed automatically in order to user be able to view the new project. The result of creating new project is shown in figure 40.

By clicking on Open project button users are able to open a project. They will redirect to description page by the controller. As it was mentioned earlier, in description tab/page users are able to write the description of the project collaboratively. A chat system is available in order to the collaborative job becomes more efficient and also users are able to use revision history is something goes wrong. As it was mentioned earlier there are two situations that can happen according to user is member of the project or not. If user is member of the project then he/she can modify the description text, use chat and revision control otherwise user accesses to description tab in read only mode which means he/she is just able to view the content of the description tab. Figure 41 shows description tab when user has editor rights, figure 42 show revision control that is available in description tab and figure 43 shows description tab in read only mode.
Figure 41: Description tab in CoLab when user is editor.

Introduction
Suppose that a website designer orders an online poker game website with real money transactions to a web designer. The web designer can design online application easily, but there are some moral and ethical issues that he concern.

Some of these ethical and moral issues are:
- What if the designer designs this website attractive so that a lot of users are interested in playing with it, it will be so beneficial for the website owner but what will happen to the users that might lose a lot of money while playing the game?
- What if he refuses to design this website while he is responsible to support his family economically?
- What if he designs it with the poor quality so that the users are not attracted to the game however he might loses his good profile that he has achieved during years of designing websites?

He could make four decisions: first of all not to design it at all, the other choice is to design it as real as possible, to design it as an attractive game so that players are so satisfied and finally design for economical benefits of the owner of the website.

Figure 42: Revision control in Description tab.
The next part that we are going to discuss is Network tab. In Network tab users are able to add, edit or delete stakeholders and interests. Figure 44 shows the result of pressing Add stakeholder button that brings opportunity to users to add new stakeholder and its related interests. Since each stakeholder name in a project should be unique, if new stakeholder name is already existed then ColLab does not allows submitting duplicated name to the server. This check is performed at the client side. This scenario is also happened for each repetitive interest name within stakeholder.

Delete a stakeholder is simple, user simply select the stakeholder and press Delete stakeholder button. A reconfirmation message appears in order to prevent mistakes if user mistakenly pressed Delete stakeholder button.

Edit a stakeholder means rename the stakeholder name, rename its interests, add new interests to the stakeholder or delete interest(s) from the stakeholder. In order to edit a
stakeholder user double clicks on the stakeholder. The dialog that is responsible for editing a stakeholder gives all these functionalities which are described above. New interest name is checked whether it is redundant or not. If it is redundant that ColLab does not allow users to all similar names for interests for a stakeholder by showing error message to users. This procedure is performed in the client side. Figure 45 shows editing stakeholder scenario.

![Edit Stakeholder and Interests](image)

Figure 45: Network tab in ColLab when user is editing a stakeholder and its related interests.

Users are also able to resize and move the rectangles. Each rectangle represents a stakeholder. Another usefulness functionality that is implemented in Network tab is red arrows which show the relation between stakeholders in order to simplify and help users having better overall view of the Network part. The Network tab and the relation between a stakeholder with other stakeholders that is marked by arrows are shown in figure 46.
Figure 46: Network tab in ColLab and the relation between a stakeholder with other stakeholders that is shown by red arrows.

The result of visiting Network tab when user is not member of the project is shown in figure 47. All the functionalities which are available for add, edit or delete stakeholders and interests are not available in this case. Non-project members are not able to move or resize stakeholder rectangles and also arrows do not work in read only mode.

Figure 47: Network tab in ColLab when a non-member user visits it.
The next tab that we are going to show as the result of implementation phase is Relations tab. As it was mentioned earlier each cell in Relations matrix is a pad. The pads are created automatically by ColLab. Add, edit or delete a stakeholder or interest directly affects the Relations table. If a project member visits Relations tab he/she is able to write in each pad in a collaborative way with other members otherwise the user is just able to see the content of each pad. Figure 48 shows an overview of Relations tab when a project member is viewing it and figure 49 shows Relations tab when a non-project member is viewing Relations tab. The important checkboxes which are used in Conclusion tab is editable for project members and those are not editable but viewable for non-project members.

Figure 48: Relations tab in ColLab when a project member visits it.
Like Relations tab, Evaluation tab is consisted of stakeholders, interests and pads but there are more elements and functionalities which are available in Evaluation tab. These elements are evaluation options and their child. The functionalities which are added to Evaluation tab are add, edit and delete evaluation options and fork functionality. These functionalities are just available for project members however non-project members are able to see evaluation options and those children without ability of modifying them. The big difference between Relations and Evaluation pad is users should add an evaluation option in Evaluation tab in order to write statement about the effect of an interest regarding to an evaluation option but in Relations tab the rows and columns are created automatically whenever stakeholders and interests get created. Figure 50 shows an overview of Evaluation tab when a project member visits this page.
As it was mentioned above, fork functionality is one of the advantages that enables user to add a children of an evaluation option to the project. By adding a forked evaluation option, all the statements which are written in the parent evaluation option are earned to the children. If a user wants to create an evaluation option by fork functionality, there is an icon that opens a dialog in order to take the name of fork option name. The given name is checked whether it begins with white spaces or not, and also it checks if fork option name is unique in the client side. If it passes the requirements above then it sends to the server in order to save into the database and also creating corresponded objects. After sending to the server and creating the objects the page gets refreshed. Figure 51 shows Evaluation tab after creating a fork option.
Like Relations tab, important checkboxes are available for project members in order to mark important evaluation and fork options. Guest and non-project users are just able to see which cells are marked as important. Figure 52 shows Evaluation tab when a non-registered or non-project member visits it.
The last tab that we are going to discuss about it is Conclusion tab. Conclusion tab consists of three major parts:

1. A collaborative environment: it enables project members write the final conclusion about the project. All the functionalities of Etherpad-Lite are available for project members in order to write their conclusion in most efficient way.

2. Important Relations: Since important relations which are marked by important check marks in Relations tab have big role to draw a conclusion then all of them are available to review by users. There is option for each important relation that either it is possible to show its content or simply hide the content. The reason of creating show/hide content functionality is to prevent users overwhelming of huge amount of information. They are able to show or hide how many important relations that they want.

3. Important Evaluations: The functionality is exactly the same as Important Relations except those are available for important evaluation and fork option which are marked by important checkboxes from Evaluation tab.

Figure 53 shows Conclusion tab when a project member visits it. The only difference between project members and non-project members when they are visiting Conclusion tab is that Etherpad-Lite functionalities are not available for non-project members however they are able to see the conclusion text in read only mode. Figure 54 shows Conclusion tab when a non-project member visits it.
Figure 53: Evaluation tab in ColLab when a project member visits it.

<table>
<thead>
<tr>
<th>Important Relations</th>
<th>Important Evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designer</td>
<td>Future Perspective towards Website Owner</td>
</tr>
<tr>
<td>Designer</td>
<td>Income towards Website Owner</td>
</tr>
<tr>
<td>Designer</td>
<td>Social Status towards Public</td>
</tr>
<tr>
<td>Website Owner</td>
<td>Economy towards Designer</td>
</tr>
<tr>
<td>Website Owner</td>
<td>Economy towards Player</td>
</tr>
</tbody>
</table>

Figure 54: Conclusion tab in ColLab when a non-project member visits it.

Maybe one good suggestion to the designers of these websites is to ask the user to make an account. So based on that account there are some limitations for the amount of time spent to play the game and also the amount of money that each player can gamble per week. Also the website should have some limitations about teenagers and children that are so excited about playing with these online websites and on the other hand so potential of being addicted to them. The threats can also prevent the access of the addicted poker gamers at the their job in order to do their job instead of playing poker.
Before we close this section it is good to mention about two useful functionalities which are the result of the implementation phase. The first functionality is invitation that enables users to invite registered CoLab members in order to join a project. Whenever a CoLab user sends invitation to a registered CoLab member then the person who gets invited to the project will automatically be member of the project. The person who gets invited to the project must be registered in CoLab. A dialog is available in order to invite users to projects. During invitation phase the email address of the person that gets invited to the project is checked by Ajax GET method in order to be available in the database which means that the user is registered in CoLab. This check is performed in the client side and if it fails then the request does not send to the server and the person who wants to invite a non-registered user will get an error message.

The second functionality that is available by invitation dialog is public address. Each project has a public address. Public address is the address of the project that both members and non-CoLab members are able to view the project in read only mode as a viewer. Project members are able to send the public link to experts or their friends in order to get feedback about the decision that they want to make by CoLab. It is also useful for students to sends the public address to other students if they should do the opposition against their project. Figure 55 shows the invitation dialog.

![Invitation dialog in CoLab](image)

**Figure 55: Invitation dialog in CoLab.**

### 5.2 Evaluation

In the current study 8 groups of the students, who have taken IT Ethics and Organization course, have been chosen at Department of Information Technology at Uppsala University. There are 3 and 4 participants in each group to make a compromised mode among all 8 groups. The CoLab system have been evaluated in the sense of how well the application supported the process of analyzing ethical problems as well as how well participants could apply the functionality of the application. Prior to the study, students were given a lecture in how to use the tool to make an analysis of the chosen problem.

After the assignment, in which the chosen students have used CoLab system, participants were asked to give feedback by filling the online questionnaire on how CoLab supported the process of handling ethical problems as well as how usable the application was in general [13]. There are 23 participants who answered to online questionnaire with
19 full responses and 4 incomplete responses. The result of participants’ analysis contains both qualitative and quantitative data based on open-ended and close-ended questions, which have determined in the questionnaire. In this case, we made both qualitative and quantitative analysis in details based on our data.

**Qualitative analysis:**
Qualitative research is a method for exploring issues, comprehending phenomena and answering questions. The qualitative method seeks out the ‘why’ not the ‘how’ of its topic through the analysis of unstructured information such as open ended survey responses [11]. Qualitative methods produce information only on the distinct cases which have been studied and it seems that more general conclusions are just propositions. Therefore, quantitative methods can be used to seek empirical support for research hypotheses [12].

In the study participants were given an online survey with two different types of questions (Close-ended and open-ended questions) to give their feedback on how well ColLab supported the process of handling ethical problems as well as how usable the application was in general. We have realized using qualitative method to analyze the given answers to the open-ended questions.

Open-ended questions are useful for providing in-depth information and allowing researcher to probe deeply into issues. Since given answers to the first open-ended question describe issues from all parts of the application, these types of questions allow researchers to be prepared for a variety of interpretations and responses. We have made some categorizations of the given answers to the first open-ended question into the meaningful categories such as security issues, bugs and etc and analyzed qualitative data in each category in details.

1. **Is there anything in ColLab that you think has to be improved?**

Security issues: In this category, there are user statements about security issues in which participants have faced during utilizing the application, discussed in depth in the next sections.

“- Also if you log in and then you enter in your own project and you have for example:
http://interact.it.uu.se/collab/index.php?action=Open+project&project_ID=20. If you change project_ID=xx you can access to other projects and can modify delete everything you want without be member of the project. So I think this bug of security should be improve.
- the security, you can enter to others assingments just cahnging some codes in the url and also invite yourself to be a member of these others assingments”

The above statements describe a security bug which users faced while testing the system. Users could have easily changed their role from viewers to editors by modifying URL and
access the project of another group and change it as one of the group members. Since system was under implementation while users started working, we could resolve some bugs which the users were reporting during the testing process. This bug has been solved by do not modifying the user session object whether user is a member of a certain project or not a member of a project. In this case, if the user tries to modify URL and alters his/her role from a viewer to an editor and attempts to change another project the system will track the user as a non-member of the project so it does not allow the user to do any action in the project.

Bugs: In this category, the user’s statements about bugs which participants have faced through utilizing the application that will be discussed in depth later.

- “It took to much time to try to change things when it just did not work.”

It seems that the problem which the user tried to mention is not clear and S/he did not clearly specify the tab, that bug occurred in through the process of the application. The following scenarios might have occurred:

In the first Scenario, the problem could have happened in relation or evaluation pages while user was filling the cells of the table out and user encountered the aforementioned problem. Thus the user tries to delete the stakeholder with all interests from the first tab (Network tab) and redone all activities for the stakeholder which took too much time.

In the second scenario, the user suddenly faced a bug in a particular situation so that s/he could not continue his/her activities any more. Therefore, the user forced to re-insert data from the beginning which could take too much time.

- “cannot open two projects at the same time.”

This problem might be occurred for a user when s/he tries to open a project in a new page by right clicking on the open project button. In the current version of the application, system allows users to open a project and copy and paste the URL from address bar to another browser address bar, then click home in the first browser and open another project. In this case, users are able to have many projects open at the same time in different browsers.

- “Only 2 out of 4 group members was able to create an account. That meant that we still had to use only 1 or 2 computers. Everybody could not use their own computers and look at it from home. It was the first 2 persons who created an account that could use it. The others could not create an account.”

This statement reveals that two of the group members have failed to create an account. They might have tried to use the same email address. The bug could also occur if these two group members signed up before. Since there was a bug in the system (for inviting others to join into the project) they could not continue working though. However, the bug
was fixed and invited option worked correctly so that users could invite others as an editor or a viewer to the project.

- “Also there were some other bugs in the system. For example, if we changed something, when we added interests, suddenly it went back to how it was before. It took to much time to try to change things when it just did not work.”

The user has tried to clarify the problem which s/he has faced while editing stakeholder in Network tab. The following scenarios might have occurred:

In the first scenario, the bug could occur while the user was trying to add more interests to stakeholder by Edit dialog( which is a kind of dialog implemented to edit stakeholder with its interests in network tab) and another his/her group modified the name of the stakeholder at the same time. However, this collaborative problem has been solved while users were working with the system synchronously.

In the second scenario, which might also happen while a user was trying to add his/her interests to the stakeholder. Since the user was not informed by any message such as “the interest has been added to the stakeholder”, thus s/he considered it as a bug. However, it is not a bug, and it is easy to show specific messages to users when any interest is added to the stakeholder.

- “After having filled out one or more fields it sometimes happens that no other field is clickable anymore. One has to reload the page manually to continue working
- It is also sometimes a challenge to select the last row at the bottom”

One of the bugs of the system happens when a user clicks on the cells which are not clickable. The bug has occurred because of the last design of tables in Relation and Evaluation tabs but it has been handled before. According to the new design of tables since checkboxes have been previously added in each cell, in separate columns, this bug took place in the system.

Usability problems: In this category, there are parts reflecting the user’s statements about usability issues which have faced during applying the application that will be discussed in depth in this section.

- “No clear workflow”

It seems like that users had confusion choosing the proper order which s/he should begin the process of analysis and decision making through it, on the one hand, and how it should be processed with the application to perform the assignment. However, we have provided a demo video to show users how and which order in the application should be used to fulfill their assignment. In order to understand the model of workflow and to
enhance the application usability, a kind of functionality could have been added as a help page or tooltips to the system.

- “too many steps”

Since the design of CoLLab application is based on EthXpert software, one of the main requirements of the tool has been provided to guide users to complete their analysis by completing different steps. These steps help users to investigate the relations and contrasts during the association process that should be completed in order to help users to make decisions.

- “The automatic resizing responded on hovering with the mouse so that the table was constantly flickering and hard to use. Use a fixed size when editing.
- All the fields resize constantly when you hover over them this is not only very tiring for the eye, but also incredibly annoying as it leads to errors, t.ex. clicking in another field than intended.
- When I used it, textfields and the matrix jumps around when navigating and editing it. They should probably be static and not move around so much to make it easier to get a good overview and also give a more firm and stable impression.
- Clicking on fields to edit, resets the viewport and really confuses me.
- when working in relation-mode it is really hard for the eyes to cope with text jumping up and down all the time when the mouse cursor is moved around the field. When one field is expanded all the other content jumps also. id several people work in the same mode this becomes horrible every time new updates are stored to the database and the view changes. The orange rectangles made this effect even worse.”

All five statements above have described the usability problem which users have faced while they were working on Relation and Evaluation pages. This usability problem is because of zooming functionality, implemented as one of the interface design feature in tables of relation and evaluation tabs. It shows more text which belong to the cells when the mouse cursor is inserted into the cell and gets back to the normal view when the mouse cursor leaves the cell. This functionality causes lots of text jumping around, being flickered through the content of the cells. This usability issue is annoying for users which will be improved for next version of the application.

Moreover, all information that are inserted by group members simultaneously should be visible on the table while they are working at the same time in order to support collaborative issues and improve the process of decision making. However, this functionality has not implemented yet and users are required to refresh the page in order to visit what other group members have added or edited in Relation and Evaluation pages. Therefore, when a user refreshes the page or clicks on the tab the whole table view will be reset and shows all information which all group members have inserted. It could cause
some confusion for users to see them all at once when S/he just refreshes the page. The orange rectangles also are not in part of zooming functionality anymore and they have been removed from the design of the tables in the second phase of the project, when users were evaluating other group’s analysis.

Interface design/issues: In this category, there are users’ statements about the Interface design issues, which participants have faced during utilizing the application and it will be discussed in depth later.

- “When your mouse cursor comes to a writing it is going to be yellow and all writing are going down and it is hard to read while everything is moving. you should fix it
- The matrix fields seem to contain two times the same text if they're not clicked. Once a "nicer" display view and then an abridged version. Why not just one? ”

This design issue occurs because of zooming functionality in Relation and Evaluation tabs which was also mentioned in usability issues category. Since this functionality has caused high physical effort such as making eyes irritated, it will be replaced by another functionality in a new version of the application.

- “main navigation is not clear, no indication were you are Naming is confusing of the main navigation
- Highlight in the main meny in what mode the user is in... this is sometimes confusing.
- Highlighting in the main navigation links, to show where you currently are at, so it is easier to understand and remember where you are in the navigation.”

In these three statements, all users have clarified an interface design issue which is related to the navigation menu. They all agree to add the characteristic of highlighting the current page tab, which users selected to clarify the workflow. This functionality will be added to the next version of ColLab application which is also mentioned as one of the tasks in future works.

- “When click a box to expend the text, the whole matrix change its format dramatically, which is annoying.”

According to the aforementioned statement, and the user’s description, an interface design issue which occurs when the user clicks a cell in the table that opens an iframe and changes other cells size. In fact, by clicking each cell of the tables in Relation and Evaluation tabs, an iframe opens, allowing users to edit the content of the cell collaboratively. The cell will be extended once the user clicks on it because of giving users more space to write and to read easily.
• “Columns are getting smaller and smaller if you add more. It is very uncomfortable to type in them.”

From a user’s experience and point of view, the current design of tables in Relation and Evaluation page requires some enhancement. The main reason for this issue is keeping the information integrated in one view without scrolling a page to right and left. We have tried to keep the table view, and to fit it to the size of a normal page which helps users have all information in one view to make the process of decision making as simple as possible.

• “I would like to have the head of the table always shown in the table view so I know which column is which.

• When scrolling down, the column headings scroll out of the view.”

Another issue is making the columns head of the tables visible while scrolling down the page, which keeps the table integrated and the user would not lose concentration on his/her analysis. This functionality will be added to the new version of ColLab application.

• “There is no way to easily see to which parent solution a child-solution belongs to”

It seems that the user preferred to see the name of the parent next to the children while adding a column by fork option which is an understandable reaction. However, the aim of the design was allowing users to select and to write the name of the column freely and to keep the design simple.

Collaborative issues: In this category, there are the users’ statements about collaborative issues which participants have faced while using the application that will be discussed in more details later.

• “No realtime updates, like google docs has so you don’t know what the others have done

• changes made by collaborators are only visible after having reloaded the page manually.”

The collaborative issue, which is one of the main aspects of the system, has been regarding as a challenging issue which could be partly achieved in the application development. In order to visit the modification and the other group members’ activities, users need to reload the page which will be handled by message passing and real-time update in the next version of the application. This functionality is mentioned as one of the tasks in future work.
Desired functionalities: In this category, there are the users’ statements about desired functionalities which participants suggest to improve the system during utilizing the application that will be discussed in depth.

● “make a general chatbox, not only for every cell, which was sometimes not visible?
I did not understand it. Maybe name the cells so you can refer to them.”

Description and conclusion pages already had chat boxes in which group members could chat with each other simultaneously. However, this chat boxes are below the iframes, which means that users could add description and conclusion of the project, to them. We have been planned to provide a general chat box which would be accessible in all tabs except home page.

● “No password reset functionality existing
● No confirmation email on account creation”

Password reset functionality enables users to reset their password when they cannot access their accounts, on the one hand, and the Email confirmation notifies users when their accounts are created, on the other hand. These are good suggestions that we have considered them as the requirements of the login page at the beginning of implementing this page. However, we decided to use noreply@it.uu.se but then we required a password for SMTP(Simple Mail Transfer Protocol) server. Furthermore, we were not allowed to use non-university email service. These functionalities will be added to the next version of ColLab application which is also mentioned as tasks in future works.

● “not possible to change the order of the rows/columns”

This functionality is also a good suggestion and makes a system more flexible and allows users to choose more options which can be considered in next version of the application.

● “Forked notes should be displayed visually not by copying the parent text.”

It seems like that a user preferred to see the text of the parent in the children column when S/he added a column by fork option which is an understandable reaction. However, the aim of the design has been allowing users to write and to add comments in the column freely and to keep the design simple. Furthermore, the sole intention of the system should be to help the user organizing the problem at hand. Therefore, adding the parent text by user will help the person to analyze the problem and make decision very carefully, of course, if it is needed.
● “In the conclusions part, you have a bit of an overview of the important stuff, but still, it feels to less.”

There is an overview of the text of important table cells in conclusion tab. It shows all important statements from relation and evaluation tables which have been marked by the user as important and s/he will have them all as an overview in the conclusion page. This overview gives all important information which the user needs to make reliable and fulfilled conclusion. Since the user should be able to make the conclusion easily, we have chosen this interface design for the conclusion page to keep simplicity and to avoid any complexity in the system.

Conceptual difficulties:

● “There should be a better example of how to use the system and how to think under for example the relation part. Is it how the stakeholder on the vertical slot affects the stakeholder on the horizontal slot or is it just how they are connected to each other of what??
● It was very unclear and we had to guess how to do it. It is not enough just to explain the system, it is useless if you do not know how you are supposed to write and what kind of connections you are supposed to do......”

It seems that users have faced conceptual difficulties to understand the association between rows and columns of the table of relation and evaluation tabs and how to fill out the cells according to those associations. The tool should help users to recognize in which order and aspect (each interest of stakeholder to each stakeholder) they should make their analysis. In fact, the relation and association process between the interest of stakeholders in rows and stakeholders in columns should be clarified. However, by applying such an association process, users lose their control over the problem which makes decision making difficult. Teaching users, providing guidelines, are probably required to solve such a problem. The logic of association process in order to proceed decision making in table needs to be reviewed and reconsidered [13].

2. Is there anything in Collab that you think is functioning particularly well?

● “The marking of notes as important and then showing them in the conclusions tab”

One of the leading requirements in the design of the application has been to support the overview of the analyzed situation [13]. It shows all important statements from relation and evaluation tables which have been checked as an important comment by the user and s/he will have them all as an overview in the conclusion page. This overview gives all important information which user needs to make the conclusion.
“The evaluation part was interesting.
it is good to have parent child relation in options in evaluation page.”

When all relationships between stakeholders have been analyzed, the user should start working on the options which can handle the problem. Users are allowed to make deeper analysis in evaluation tab. These deeper analyses are compromise scenarios which can be derived from existing options. This feature is useful for including many options which differ partly in the analysis [13].

“Generally it was functioning well.
Yes, work in parallel with your members.”

From a user’s experience and point of view, it is good the application helps users fulfill the task and make the simulation work properly. It seems that the process in the collaboration application supports users in taking the perspective of others. The second statement expresses a kind of subjective opinion, which states that the user gets help to investigate a scenario from other group members’ point of view in real time [13].

“The different tabs made it easy to get an overview of what we had done.
links from one section to the next”

In fact, the application is deceptively simple. There are only seven tabs and a handful of functions. The application gives the required help to users to bring the order which has already been appreciated by some users. Such different tabs will assist users in both identifying and analyzing the system.

3. Other comment

“Nice software, seems to be able to become a really useful tool, with further development and testing.
I really like the idea behind collab. However, as we were using Google Docs for the first assignment, we already had most of the advantages of collab.”

It seems that the application has already been appreciated and utilized well by some users. Theses quotes support ColLab application with collaborative functionality that allows users to focus on the core dynamic of a problem. Furthermore, the second quote supports important requirement on the tool which is a collaborative functionality. The user compared ColLab application with Google Docs which has been applied to do the first assignment of the IT Ethics and Organization course. Therefore, ColLab application supplied the collaborative functionality in a way which helps the user compare the tool and Google Docs which is an incredible achievement.
Quantitative

Quantitative data is data measured or identified on a numerical scale. Numerical data can be analyzed using statistical methods, and results can be displayed using tables, charts, histograms and graphs. For example, a researcher will ask a questions to a participant that include words how often, how many or percentage. The answers from the questions will be numerical. Quantitative data involves amounts, measurements, or anything of quantity [40].

Quantitative Research refers to the systematic empirical investigation of social phenomena via statistical, mathematical or computational techniques. The objective of quantitative research is to develop and employ mathematical models, theories and/or hypotheses pertaining to phenomena. The process of measurement is central to quantitative research because it provides the fundamental connection between empirical observation and mathematical expression of quantitative relationships. Quantitative data is any data that is in numerical form such as statistics, percentages, etc. In layman's terms, this means that the quantitative researcher asks a specific, narrow question and collects numerical data from participants to answer the question. The researcher analyzes the data with the help of statistics. The researcher is hoping the numbers will yield an unbiased result that can be generalized to some larger population. Qualitative research, on the other hand, asks broad questions and collects word data from participants. The researcher looks for themes and describes the information in themes and patterns exclusive to that set of participants [41].

The quantitative data for ColLab is gathered by collecting the answers of 27 questions from students. 23 out of 27 questions used Likert scale method. A Likert scale is a psychometric scale commonly involved in research that employs questionnaires. It is the most widely used approach to scaling responses in survey research, such that the term is often used interchangeably with rating scale, or more accurately the Likert-type scale, even though the two are not synonymous. The scale is named after its inventor, psychologist Rensis Likert. Likert scaling is a bipolar scaling method, measuring either positive or negative response to a statement [46]. In our questionnaire we have used five-level Likert item which are listed below:

1. Strongly agree.
2. Agree.
3. Neither agree nor disagree.
4. Disagree.
5. Strongly disagree.

These questions are generally based on usefulness of ColLab, usefulness of different parts of ColLab which help students to complete their assignments, how ColLab is easy or hard to use, ColLab functionality and boringness or attractiveness of ColLab. We will pick up and discuss about some of the important answers from survey but all of them, and their results are available in appendix.
The first result that we got from quantitative data if we compare agree and strongly agree against disagree and strongly disagree answers is most of the students had no difficulty learning how to work with ColLab (table 1).

![Table 1](image)

Table 1: The results of “I have no difficulty learning how to work in ColLab” question.

Most students cited Evaluation and Conclusion views/tabs are very valuable for solving the assignment (graph 1 and 2).

![Graph 1](image)

Graph 1: Results of the question “The Evaluation view was very valuable for solving the assignment”.

![Graph 2](image)

Graph 2: Results of the question “The Conclusion view was very valuable for solving the assignment”.

Total numbers of the students who are agree or strongly agree that ColLab is useful for ethical decision making is more than the students who are disagree with that (graph 3).
ColLab is based on several technologies and new ideas. It was not easy to combine these technologies and apply the ideas which are behind ColLab to the entire system. Our goal was to implement ColLab as sustainable as possible but unfortunately the time was limited and it was the first version of ColLab. We knew there are some bugs which are existed in ColLab and it is exactly that we faced with the answers we got from the students. 73.69% of the students are disagree or strongly disagree that ColLab functioned without problems. Only 5.26% of the students are strongly agree that ColLab functioned without problems and 21.05% of them are neither disagree nor agree with that. The correlation between students who disagree or strongly disagree that ColLab functioned without problems and those who agree or strongly agree with ColLab made me bored with the given assignment is 0.53. The bugs that are existed in ColLab leads to the situation that 47.37% of the students tell they would rather have used another application (other than ColLab) to do the assignment.

The participants who are disagree or strongly disagree that ColLab functioned without problem, more likely agree or strongly agree that they would rather have used another application than ColLab to do the assignment, ColLab made them bored with the given assignment, ColLab decreased their motivation to complete the assignment and sometimes they were annoyed with ColLab. They are also likely disagree or strongly disagree that ColLab was convenient to use and they are more likely disagree or strongly disagree that My ColLab is easy to work in.

The participants who are strongly agree or agree that they had no difficulty learning how to work in ColLab more likely prefer to use another application than ColLab to do the assignment and they are more likely disagree or strongly disagree that ColLab is easy to work in.
The participants who are disagree or strongly disagree that ColLab increased their motivations to complete the assignment likely disagree or strongly disagree that ColLab provided good support for solving the assignment and ColLab is easy to work in. 47.37% of the students are also complained about too many mouse clicks that are required in ColLab that should be considered in the next version of ColLab. One of the main goals of implementing ColLab was ColLab make it easy to collaborate with other group members. 10.53% of the students are strongly agree, 26.32% are agree, 31.58% are neither agree nor disagree, 26.32% disagree and 5.26% are strongly disagree that ColLab made it easy to collaborate with the other group members that is not a satisfactory result for us. We will discuss about the suggested solution for this result in the future work chapter.
6. Discussion

We have implemented ColLab application for ethical competence in decision making. This application is a new version of EthXpert software. We have maintained EthXpert software and adding features to it, so we called it ColLab application. ColLab is a support system for structuring and systematizing considerations in the ethical analysis. This ethical support tool is intended for both analyst and decision makers [55]. We will discuss about some important differences between EthXpert software and ColLab application, the problems we have encountered while implementing ColLab application, advantages and disadvantages of ColLab against Ethxpert system in this section.

- Important differences between EthXpert software and ColLab application:

1. Collaborative functionality: This is the most important differences between EthXpert and ColLab. ColLab, which is a new version of EthXpert software, was equipped by collaborative real-time feature. Due to this new feature, we determined two different roles for users, editor and viewer. The user, who wants to create a project, should sign-in to the system first, if s/he has signed up before. Otherwise, the user should create account to be able to login to the system from sign up page first. Then, the user can create a project on the homepage after sign-in procedure. The user can see all the projects, which were created by him/her or invited to as an editor, on this page (homepage was shown in figure 39). S/he can invite other users to the project as editors by inserting their email address in invitation dialog (figure 55 shows invitation dialog). The user can also give other users the URL address of the project which is available on invitation dialog. In this case, the user, who has the role of viewer of the project, can add the URL address of the project to the address bar of his/her browser and only being able to view the project. However, there is no sign-in, sign up, homepage (user page), and invitation tab in Ethxpert software.

2. Description tab: Users should write a description of the project in this page. Since ColLab application is equipped by collaborative functionality, we have two different roles, an editor and a viewer, to defining users in ColLab. In this case, figure 41 shows description page for editors who are members of the project. We create a pad and open connection with Etherpad-Lite on description tab in ColLab. All project members can edit the page synchronously and see what other members are writing. Revision control, which is the management of changes to the document, is one of the functionalities in this page (which was shown in figure 42). If the user is only viewer of the project, s/he cannot edit the page so that the interface which is shown to the viewer is different (figure 43 shows the description page for the viewer) from what is shown to editor. Users cannot see what other members are doing.

3. Network tab: This page also has some collaborative real-time features in ColLab application. Since we do not have message passing with NodeJS yet, we simulate this functionality to inform users with perceptible information. While two users want
to add, edit and delete the same stakeholder or interest at the same time, a message will appear on the top of the network page (figure 46 shows the network tab in CoLLab). For instance, imagine two users want to delete a stakeholder at the same time. A message will be shown on the top of the network page for the user, who presses the OK button of delete dialog just some millisecond after the first user. The message shows that the stakeholder was deleted by another user. There are some differences on network tab between interface design of EthXpert and interface design of CoLLab. For example, by clicking the head of the arrows, a dialog is shown up, which represents the interests’ names of the stakeholder beside the content of the cell (the interest’s cell which is toward the stakeholder in relation tab). The arrow class on Network tab will be improved in next version of CoLLab which will be discussed in section 6. As you can see in the figure 56, the shape of stakeholder as an organization is rectangular and the non-organization is shown as circle. However, we decided to have only one shape to represent stakeholder on network tab in CoLLab to keep simplicity and avoid complexity.

Figure 56: Network page of EthXpert software.

4. Relation tab: When a user adds stakeholders and interests on network page, a table in relation page is built which is the same procedure for both CoLLab and EthXpert. There is also collaborative functionality by using Etherpad-Lite on relation tab in CoLLab. Whenever a user clicks on one of the cell of the table, a connection to Etherpad-Lite will be opened (figure 48 shows relation tab in CoLLab). If two users work on the same cell simultaneously, they can see what they are both writing such as description tab. If users work on different cells of each other, they need to refresh the page to be able to see what other users have done which will be improved in future. There is a checkbox on the top of each cell on the table. If
the user wants to access the comment of the cell on conclusion page, s/he should mark the checkbox. You can see the relation tab in EthXpert in figure 57. If user attempts to write on the cell, s/he needs to double click on the cell and a dialog will pop-up. Then, the user can write on the cell and mark the related checkbox as important, if s/he wants to have it on the conclusion page.

5. Evaluation tab: When a user adds stakeholders and interests, a table is built only with rows and no columns on evaluation tab, which is the same for both ColLab and EthXpert system. Then the user needs to add options to create the columns on the table. The collaborative issue on this page is the same as what we have mentioned for relation tab in ColLab. The editor view of evaluation tab in ColLab was shown in figure 50. A user can insert the name of the option in the related text box and press enter. An option will be created as a column and the user can write on the cells by clicking them. The procedure of adding fork option to the table is almost the same for both EthXpert and ColLab. Even though, we do not show the comments of the fork options’ parent on the cells; the user can add it to the related cells if s/he wants to do so. The reason is because S/he should decide if it is needed. The evaluation of EthXpert software is shown in figure 58.
<table>
<thead>
<tr>
<th><strong>Designer</strong></th>
<th><strong>Social Status</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Not to Design</strong></td>
<td><strong>What are the risks and opportunities in this?</strong></td>
</tr>
<tr>
<td><strong>Previous statements ( iterative) argument</strong></td>
<td><strong>Maybe he could be critized by his colleagues.</strong></td>
</tr>
<tr>
<td><strong>Statements about designer's interest</strong></td>
<td><strong>Social Status</strong></td>
</tr>
<tr>
<td>&quot;Social Status&quot;</td>
<td><strong>Unchanged</strong></td>
</tr>
<tr>
<td><strong>Unchanged</strong></td>
<td><strong>Unchanged</strong></td>
</tr>
<tr>
<td><strong>More than the public admire his design.</strong></td>
<td><strong>Unchanged</strong></td>
</tr>
<tr>
<td><strong>Unchanged</strong></td>
<td><strong>Unchanged</strong></td>
</tr>
<tr>
<td><strong>Statements where designer is the target</strong></td>
<td><strong>Unchanged</strong></td>
</tr>
<tr>
<td><strong>Unchanged</strong></td>
<td><strong>The better the design, the more addiction he/she will have.</strong></td>
</tr>
<tr>
<td><strong>Unchanged</strong></td>
<td><strong>The more he/she offers to the public, the more addiction he/she will have.</strong></td>
</tr>
<tr>
<td><strong>Unchanged</strong></td>
<td><strong>He/She offer more jobs to the public.</strong></td>
</tr>
<tr>
<td><strong>Unchanged</strong></td>
<td><strong>It will lose money in case of a good design.</strong></td>
</tr>
<tr>
<td><strong>Unchanged</strong></td>
<td><strong>If the design is good, there will be negative impact on real life.</strong></td>
</tr>
<tr>
<td><strong>Unchanged</strong></td>
<td><strong>The probability of winning is high if the websites is good.</strong></td>
</tr>
<tr>
<td><strong>Unchanged</strong></td>
<td><strong>It depends on how attractive the design is.</strong></td>
</tr>
<tr>
<td><strong>Unchanged</strong></td>
<td><strong>Unchanged</strong></td>
</tr>
<tr>
<td><strong>Unchanged</strong></td>
<td><strong>Increase Economy If Designer designs good website</strong></td>
</tr>
<tr>
<td><strong>Unchanged</strong></td>
<td><strong>If the website is good, the hits will increase.</strong></td>
</tr>
</tbody>
</table>

**Figure 58:** Evaluation page of EthXpert software.

**Figure 59:** Dialog to add text to the cell on the evaluation page in EthXpert

---

6. **Conclusion tab:** The conclusion tab in CoLLab was shown in figure 53 in section 4.1. Users can write their conclusion of the whole project by considering the important comments (the cells which the user marked on relation and evaluation tabs as important comments) on this page. Since there is an open pad on this page, users can write on the page simultaneously, which is the same as description page. They
have also access to important comments of both relation and evaluation tabs at the bottom of the page. The user can see each comment by clicking show option and turn it to hide by another click (to keep the page clear if s/he wants). We decided to have separate tables to show important comments of the relation and evaluation tabs. That is because of, its asset to the user recognition and reference to the cell of the table easily. The conclusion tab of EthXpert software is shown in figure 60. Users can write the conclusion of the project on this page regarding to the important comments of both relation and evaluation tabs which are accessible at the bottom of the page.

![Figure 60: Conclusion page of EthXpert software.](image)

- The problems we have encountered while implementing ColLab application:

  1. Opening and closing the connection of Etherpad-Lite for each cell: Since collaborative real-time issue is one of the main aspects of our project, we have started to apply Etherpad-Lite functionalities in our program. We used a jQuery plugin, newEtherpad.js, to create an iframe, to make a pad, and open a connection with Etherpad-Lite. Since we intended to have iframe for each cell of the table, we created pads by using pad() method of newEtherpad.js for each cell. We also get the content of the pads by using
getContents of newEtherpad.js, to show the content of the pads for each cell. Whenever a cell was clicked, a user could write on the pad; when s/he unclicked the cell (by clicking another cell on the table) the content of the pad was shown. However, the connections with Etherpad-Lite (with NodeJS server) was open for each cell of the table continuously. It put a lot of loads on the server and decreased the speed of connection with the server considerably. In this case, we have decided to show data (content of the pad on related cell) directly from database and an open connection with Etherpad Lite (NodeJS server) once a cell clicked. Therefore, we could find an efficient solution for our problem.

2. Problem with getting data back from server in Network tab: Since network tab, is almost a pure client side, we had lots of difficulties to get data back from the server. We have tried different methods to pass data from PHP to javaScript. At least we could send data back by Ajax’s success() function from the server on the network tab. We have also used this functionality on invitation tab. In this case, when user invites another user by his/her email address on invitation tab, the user-id is checked with success function of Ajax if it has already existed.

3. Error-handling if two users modify any stakeholder/interest simultaneously in the network tab: This error occurred during testing the system. For example, when two users deleted the same interest of a stakeholder at the same time, the error took place during the process. Since we did not have collaborative real time functionality, such as a message passing on network tab, we have simulated this functionality in an efficient way. In order to solve this problem, we check if the stakeholder or interest exists in the database by Ajax with success function. If the user first did the action on the same element as another user wants to, the message would appear on the top of network tab. This message informs another user in letting him/her know that s/he cannot do his/her action such as deleting the interest that was performed by another user.

- What are the advantages of ColLab against EthXpert?

1. ColLab is a web-based application so that a user does not need to download and to use further application. However, a user has to download EthXpert software in order to apply it.

2. Everything is stored on the web which means it is accessible across the world.

3. It is a collaborative tool: Group members can participate at any time across the world. Therefore, they can contribute to do the project synchronously and asynchronously from different places. There is also versioning control for ColLab on description and conclusion tabs. Then, if something goes wrong, it is possible to change it to acceptable and stable situation. ColLab also integrated with Chat functionality which is accessible on description and
conclusion tabs. It has a user friendly interface and all projects are easily available to open without any need to load an xml file.

4. It was integrated with user Role (editor/viewer): The user can invite other people as an editor of his/her project. Otherwise, the user can give only the URL address, which is accessible on invitation tab, to people as viewers of the project to review the analysis.

5. Capability to add more functionalities: The features, which are available for web-based applications, are not accessible for standalone software. For instance, adding Mail notification feature to ColLab in order to inform a user whenever his/her group member modifies the analysis is one of the suggested functionalities.

- What are the disadvantages of ColLab against EthXpert?

1. Since ColLab is a web-based application, it cannot function without internet connectivity.
2. The arrow class of EthXpert on network tab has functionalities which is not implemented for ColLab. It will be improved in the new version of ColLab as it is discussed in section 6 (Future work).
7. Future work

The first version of ColLab is successfully implemented. We considered implementing ColLab as simple as possible because we did not want to overwhelm and confuse students by defining complicated functionalities. We did not want even adding simple extra pages in order to modify user information such as change first name or last name of students because the first aim of implementing ColLab was implementing a web-based application that enables students solving their assignments collaboratively. By adding extra tabs/view there is risk to confuse students.

We should consider that our target users in ColLab are not just limited to the students. We really want to offer ColLab to professional decision makers and to achieve this goal we should add more functionalities into ColLab. If we think about the real target users then the list that is presented below is the continuation of functionalities that should be considered in the next versions of ColLab:

- In the current version of ColLab users should refresh Network, Relations and Evaluation tabs to see the changes which are made by other group members. We have tried implementing a queue that remembers all the changes which are made by other members by sending a message. Each cell in the queue consisted of a message. Each message consisted of project number, the object that is modified or created and the modification itself. We tried to use shared memory in PHP but since the size of each message may be different and there is huge risk to put a large message in a smaller dedicated cell then we stopped implementing this functionality since there is risk that a large message placed out of the memory and it causes segmentation fault. In this case ColLab has crashed and there was no option to return to the previous state. The solution for real time updates is implementing a network stream. By implementing the network stream clients are able to notify whenever an object gets changed by group members and also it is possible to send changes which are made by a user to other group members. It is possible to use NodeJS API since Etherpad-Lite is based on it and all the functionalities are available in order to implement the network stream.

- In both Relations and Evaluation tabs we have used tables. Each cell in these tables is a pad and the content of each pad is gathered from the database. In other words in order to draw these tables, we have to send n queries where n is the total number of cells in each table. If it is possible to use two views in the database in the way that one view draws Relations table and the other one draws Evaluation table then we just need to send just two queries to database to fetch the data that is available in the database. If this method is possible to implement then drawing of Relations and Evaluation tabs becomes faster and the objects which are stored in the project session object become lighter.

- Email notification is one of the important functionality that should be implemented in the next version of ColLab. As it was mentioned in chapter 5 we wanted to implement this functionality but we needed university SMTP password that it did not available for us. Email notification functionality is very simple to implement and it...
allows users to get registration information such as username and password via email whenever they registered themselves in ColLab. It is also useful for those users who forgot their password in the way that it allows users to reset their passwords. The other usefulness of implementing email notification users is that whenever a project member invites other users they get an email that notifies them whether accept the invitation or refuse it.

- Admin control panel is another functionality that can be helpful for administration proposes. It allows admin to add and remove users and projects in ColLab.
- When we consider about the target users of ColLab, it is required to add user control panel tab in ColLab. By implementing this tab users are able to modify their information such as first name, last name, password etc. It can also improve in the way that enables users to search all available projects in ColLab and add other projects to their favorite list in order to follow them as a viewer.
- ColLab is safe enough in both security and access control perspectives. To ensure that no one can be able to access other projects if the author of the project does not like it then the project ID that is sent by GET method and it is available by edit the URL in order to access as viewer to other projects the project ID should be encrypted in the URL. It is also important to use other encryption/decryption methods than MD5 that is responsible for storing passwords in the database because MD5 is not safe enough and the algorithm of encryption/decryption is found recently.
- According to the quotes that we got from qualitative analysis many students do not like zoom functionality. They have complained about hovering function that resizes the cells in Relations and Evaluation tables. This functionality should be either removed or redesigned.
- The arrow class in Network tab should be improved in the way that it should allow users to use it as a single column in Relations table in order to write their statements.
- The tables which are used in Relations and Evaluation tabs are static. Those should be redesigned as dynamic tables. By having dynamic tables users are able to sort table elements and modifying table sizes. The column name of tables should also be visible when users scroll down the page as students mentioned in the survey.
8. Summary and Conclusions

In this report we have started discussing about the definition of ethics and we have also mentioned why ethical decision making is important by discussing about two scenarios. We have discussed about the aim of this thesis that is based on implementing a collaborative web-based application that is useful for ethical decision making. As it was mentioned the idea of ColLab is based on EthXpert system.

In order to implement ColLab we did a research about available APIs which bring collaborative functionalities to other applications. We compare some of them and chose the best one that is compatible with our needs. The research should give a broad overview about the other functionalities which are available in other APIs. We talked also about the differences between drawing tools which are compatible with the ideas of ColLab. Canvas in HTML5 was elected as our drawing tool and Etherpad-Lite was elected as our collaborative API.

In chapter 4 we started to discuss about server and client side technologies which are used in ColLab. Then we discussed about client side technologies which are used in ColLab in more detail i.e. why they have used, how they have used and how these technologies implement functionalities in ColLab and how they are connected to other parts. After discussing about client side technologies we have started discussing about the server side technologies since ColLab is based on both server side and client side technologies. All tables in the database with their attributes are fully discussed with numerous examples which show how ColLab collect required data from the database. All the examples are discussed with real database queries which are performed by simple queries. After discussing about the database we have started discussing about NodeJS and Etherpad-Lite which how they are connected and perform their jobs in ColLab, how Etherpad-Lite is configured and which plugins are necessary in order to connect Etherpad-Lite with ColLab. Since PHP is the programming language that ColLab is based on it, we discussed about all the classes and their properties which are available in ColLab. We compared also about why PDO is the best alternative as database extension in ColLab. In this chapter we divided ColLab into some modules and discuss about each module and its objects in detail. At the MVC section we collect all the modules which were discussed earlier and showed why they implemented in that way. We tried to make a connection between this report and the person who reads this report that classes and objects are implemented like the ways which are implemented in ColLab and at the MVC section we tried to prove the correctness of the ideas which are behind ColLab. We enforced us to explain some objects and classes in details since the relation between entities are important otherwise it becomes very hard for readers to get all the concepts and relation between all the modules which are implemented in ColLab.

In chapter 5 we show the results of the project and we analyze the data that we got from the questionnaire that is 19 students answered to it. The analysis is performed based on both qualitative and quantitative methods.
Chapter 6 consists of comparison between ColLab and EthXpert and we also discussed about the problems which are faced during implementation of ColLab. In this chapter it was mentioned how we solve the problems.

We explained about the desired functionalities which either those are requested by users who answer to the questionnaire or our ideas which make ColLab as efficient application as possible in chapter 7.

The first step of implementing ColLab is finished. We know by implementing new technologies it is possible to redesign some parts of ColLab e.g. we should eliminate storing huge amount of data in the session objects and performing many queries to draw Relations and Evaluation tables. Some of the functionalities in ColLab should either redesign or removed like zoom functionality by hovering mouse pointer over the cells in Relations and Evaluation Tables. In the next version of CoLab it is required to implement message passing over a network stream in order to bring the real collaborative environment to ColLab.

ColLab will grow and we hope someday in future it gets used widely for making ethical decision and it is our goal of implementing ColLab.
9. References


http://www.w3schools.com/jquery/jquery_ajax.asp


[29] jQuery API. [Online; accessed June 2012]
http://api.jquery.com/Query.ajax/

http://en.wikipedia.org/wiki/JQuery

http://jquery.com/


http://api.jquery.com/delegate/

[34] Delwin Campbell, Are you using CSS3 Appropriately, October 2011. [Online; accessed July 2012]
http://sixrevisions.com/css/using-css3-appropriately/

http://webdesign.about.com/od/html5tags/a/html5-canva
tutorial.htm


http://docs.jquery.com/UI/Dialog#events

[38] Jay Blanchard, Getting the most out of jQuery selectors, July 2011. [Online; accessed July 2012]
http://www.netmagazine.com/tutorials/getting-most-out-jquery-selector


http://www.kineticjs.com/

http://simonsarris.com/blog/225-canvas-selecting-resizing-shape
http://simonsarris.com/blog/140-canvas-moving-selectable-shapes

http://www.it.uu.se/edu/course/homepage/itetikorg/vt09/EthXpert.pdf
10. Appendix

10.1 Questionnaire

1. The given assignment was easy to understand.
2. ColLab functioned without problems.
3. I had no difficulty learning how to work in ColLab.
4. ColLab made me bored with the given assignment.
5. ColLab was convenient to use.
6. ColLab increased my motivation to complete the assignment.
7. ColLab provided good support for solving the assignment.
8. Sometimes I was annoyed with ColLab.
9. ColLab made it easy to collaborate with the other group members.
10. ColLab required too many mouse clicks.
11. I felt that ColLab restricted my creativity.
12. ColLab functioned in a predictable way.
13. I felt socially included when working with ColLab.
14. I would rather have used another application (other than ColLab) to do the assignment.
15. I had a good overview over my work in ColLab.
16. My experience is that ColLab is easy to work in.
17. There were too many repetitive tasks when working in ColLab.
18. The "Network" view was very valuable for solving the assignment.
19. The "Relations" view was very valuable for solving the assignment.
20. The "Evaluations" view was very valuable for solving the assignment.
21. The "Conclusions" view was very valuable for solving the assignment.
22. The chat was very valuable for solving the assignment.
23. I think ColLab is useful for ethical decision making.
24. Is there anything in ColLab that you think has to be improved?
25. Is there anything in ColLab that you think is functioning particularly well?
26. How did you get instructions for how to work with the assignment?
   ● The demo project.
   ● The screencast.
   ● The lecture.
   ● Friends/group members.
27. Other comments

10.2 Quantitative analysis

Quantitative data from questionnaire:
| Standard | Average | Median | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 |
| Easy to understand | Without problems | Learning | Boring | Convenient to use | Motivational | Good support | Annoying | Collaboration | Mouse clicks | Creativity restriction | Predictable | Involving | Another application | Overview | Easy to work | Repetitive tasks | Network tab | Relations tab | Evaluations tab | Conclusions tab | Chat usefulness | CoLLab usefulness | Demo project | Screen cast | Lecture | Friends/group members |
Correlation between answers:

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 3 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 4 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 5 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 6 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 7 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 8 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 9 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 10|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 11|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 12|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 13|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 14|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 15|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 16|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 17|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 18|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 19|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 20|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 21|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 22|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 23|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 24|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 25|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 26|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 27|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |