Självständigt arbete på grundnivå

Independent degree project  first cycle

Dateknik
Computer Engineering

A Web- and App- Based Interaction System Bridging the Gap between Patients and Doctors

Zehua Li
Dan Luo
A Web- and App- Based Interaction System Bridging the Gap between Patients and Doctors
Zehua Li, Dan Luo

2014-06-20

MID SWEDEN UNIVERSITY
The Department of Information and Communication Systems (IKS)

Examiner: Ulf Jennehag, ulf.jennehag@miun.se
Supervisor: Tingting Zhang, tingting.zhang@miun.se
Author: Dan Luo, dalu1303@student.miun.se
         Zehua Li, zeli1300@student.miun.se
Degree programme: Bachelor of Science in Engineering, 180 credits
Main field of study: Computer Engineering
Semester, year: Second semester, 2014
Abstract

In modern society, people are lack of physical exercise and have a poor eating habit. A sedentary lifestyle gives rise to an increase of diseases and death rate among European Union. Compared to the general population, people with Serious Mental Illness(SMI) suffer more from poor health condition. Thus the project, as a part of the LIFEHOPE.EU, is established to ameliorate to this circumstances through encouraging people with SMI to regularly attend physical exercises and eat a balanced diet. However, most patients with SMI who participate in the program live in remote areas. It is inconvenient to hold face-to-face meetings with either doctors or group members. Therefore, it is necessary to develop a system to solve this problem. The system contains an Android application for patients and an administrative website for doctors to keep contact with patients in order to monitor and assist patients' recovery. In this paper, the system is based on Apache Web Server and Android. How the functions of this system are created and realized will be elaborately introduced and illustrated in the paper. In addition, CakePHP framework applies in website development. After design and implementation, tests are run on both the Android app and the website to evaluate security and performance. The LIFEHOPE.EU program has been satisfied with the final outcome and decided to put it into use.

Keywords: Apache, Android, CakePHP, e-health
Acknowledgments

At first, we would like to thank Professor Tingting Zhang, our supervisor, who offers us the precious opportunity to participate in this project. She patiently gives us suggestions about the project work and paper writing.

In addition, we want to express my sincere gratitude to Professor Rönngren Ylv, Professor Björk Annett and Professor Kristiansen Lisbeth as well. Thanks them to come up with the original idea of the project. The outcome of this project has gradually become desirable through their patient direction to us.

Finally, we wish to voice our deep acknowledgment to all those who have helped and supported us.
# Table of Contents

Abstract .............................................................................................................. iii
Acknowledgments .............................................................................................. iv
Terminology ...................................................................................................... vii

## 1 Introduction .............................................................................................. 1
1.1 Background and problem motivation ..................................................... 1
1.2 Overall aim ............................................................................................... 2
1.3 Scope ........................................................................................................ 2
1.4 Concrete and verifiable goals .................................................................. 2
1.5 Outline ....................................................................................................... 3
1.6 Contributions ........................................................................................... 3

## 2 Theory ........................................................................................................ 4
2.1 Apache HTTP Server ................................................................................ 4
2.2 PHP .......................................................................................................... 4
2.3 CakePHP ................................................................................................... 4
2.4 Android ..................................................................................................... 4
2.5 DBMS ........................................................................................................ 5
2.5.1 MySql .................................................................................................. 6
2.5.2 SQLite .................................................................................................. 6
2.6 XML .......................................................................................................... 6
2.7 UML .......................................................................................................... 6
2.8 Cipher ....................................................................................................... 6

## 3 Methodology .............................................................................................. 7
3.1 RUP .......................................................................................................... 7
3.1.1 The concept of RUP ........................................................................... 7
3.1.2 Six Best practices ............................................................................... 7
3.1.3 Process structure ............................................................................... 8
3.2 Software prototyping .............................................................................. 8
3.3 MVC ......................................................................................................... 9
3.4 DAO .......................................................................................................... 10
3.5 Black-box testing ................................................................................... 10

## 4 Design and Implementation ...................................................................... 12
4.1 Requirement analysis ............................................................................. 12
4.1.1 Feasibility Analysis .......................................................................... 12
4.1.2 System use case model ...................................................................... 12
4.1.3 Prototype ........................................................................................... 16
4.1.4 Requirements Specification ............................................................... 18
4.2 Design ....................................................................................................... 18
4.2.1 System Architecture ......................................................................... 18
4.2.2 Functionality structure ...................................................................... 20
4.2.3 Flowchart of system ......................................................................... 23
A Web- and App- Based Interaction System Bridging the Gap between Patients and Doctors
Zehua Li, Dan Luo

Terminology

Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMI</td>
<td>Serious Mental Illness</td>
</tr>
<tr>
<td>MVC</td>
<td>Model-View-Controller</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>DAO</td>
<td>Data Access Object</td>
</tr>
</tbody>
</table>
1 Introduction

The public health burden of a sedentary lifestyle has been recognized widely. A sedentary lifestyle leads to about 3.5% of the disease burden and up to 10% of deaths in European Union. The economic costs on medical treatment related to physical inactivity are huge. Doing physical exercises at a moderate level could avoid health risks and financial costs. Compared to the general population, People with Serious Mental Illness (SMI) suffer more from disease burden. Moreover, their high death rate is mainly due to a sedentary lifestyle. [1]

This project is a part of the LIFEHOPE.EU program. The aim of the program is to reduce physical inactivity people with SMI by means of improving their eating habits and the level of physical exercises. The primary objective of LIFEHOPE.EU is to establish a significant body of evidence with the potential to improve the level of physical activities and the realization of adopted healthy eating habits among people suffering from SMI in EU. This will be achieved through joint participation of clients and care providers, education of relatives and workshops on lifestyle focusing at physical activity and diet habits. The secondary objective is to evaluate the effects of physical activities and eating habits on social health, mental well-being and symptom reduction related to SMI. [1]

1.1 Background and problem motivation

In LIFEHOPE.EU, many of the patients, people with SMI, live in remote areas and some of them are aged, which leads to inconvenient attendance to face-to-face meeting with their doctors or group members. Considering the fact that they are not accompanied with their doctors or nurses in most of the time, when they are overwhelmed by stress or depression, they only could rely on themselves, which is a tough task for them. Moreover, patients' statistics expect to be collected so that the doctors could have a comprehensive view on their situation and more accurate instructions could be given back to the patients individually.

Thus, a system which contains an Android application for patients and an administrative website based on Apache web server and CakePHP framework for doctors is required. The purpose of the system is to make a bridge between doctors and patients in order to monitor and assist patients' recovery.

Our project develops a system involves such functions as stress management, physical exercises encouragement, improvement of self-accomplishment and feedback. For the patients in the project are people with SMI, if the operation is too trivial or too complex, perhaps it is too difficult for them to handle. Therefore, the operation of Android need to be available on one application and easy to navigate.
A Web- and App- Based Interaction System Bridging the Gap between Patients and Doctors
Zehua Li, Dan Luo
2014-6-20

1.2 Overall aim
The overall aim of this project work is to develop an Android APP for patients’ daily assistance and a website which stores patients’ information in MySQL database and displays them on the web pages. The website serves as an administrative platform for doctors.

1.3 Scope
The project work has its focus on the LIFEHOPE.EU program required functions of the Android application and the website. The App concentrates on UI design. The website puts emphasis on security. The final outcome will be an android application performs a friendly user experiences and updates data from the website at configured intervals and an administrative website adopts some security algorithm to ensure the safety of data transformation.

1.4 Concrete and verifiable goals
The purpose of the project work is to develop a system by Apache web server and Android. The entire developing procedure is divided into several steps and concrete goals are set as follows.

For website:

(1) Use that framework to develop a website based on Apache web server to fulfill such administrative needs for doctors as manipulating patients’ basic information, uploading multimedia which alleviate stress and depression (refers to measure and book functions in the system), encouraging more physical activities (stands for sports schedule part), improving patients' self-accomplishment by setting weekly goal on patients' own desires (refers to recipe) and view patients' feedback of their feelings on the Android application and the recovery process.

(2) Optimize the website by providing managing groups and the code to improve user experiences and work efficiency.

(3) Evaluate the security and loading time of the website.

For Android application:

(1) Use Android software development kit to develop an Android application to meet such requirements as encouraging the patients to attend physical activities (stands for sports schedule functions), providing assistance when nervous or depressed (refers to measure and book in our system), giving spirit to finish goals in order to enable the patients be more confident and arouse self-accomplishment (refers to recipe part), recording users' daily physical exercises (refers to walk steps which runs in the background), writing their feelings about the Android application and setting their individual preferences (stands for settings).

(2) Evaluate the responding time of the Android application.
1.5 **Outline**

The structure of this project is as below:

Chapter 1 introduces the background and motivation of this project work, and both overall goals and concrete goals. Chapter 2 introduces several terms and technologies related to the project. Chapter 3 presents the methodology used in the overall developing process. Chapter 4 stresses on the design patterns and details in design. Chapter 5 displays the outcome of this project. Chapter 6 draws a conclusion and pointed out the direction of future work.

1.6 **Contributions**

The project work is split into halves, the website and the Android application. The website is conducted by Zehua Li and the app is conducted by Dan Luo. The prototype of the system is made in the course DT107G, applied computer engineering.
2 Theory

2.1 Apache HTTP Server

Apache HTTP Server, generally referred to as Apache, is an Apache Software Foundation open source web server that can run on most computer operating systems, including Unix, OpenVMS, OS X, Microsoft Windows, OS/2, TPF, Solaris, FreeBSD, Linux, Novell NetWare and eComStation. [2] It is widely used due to its multi-platform and safety. Apache is the most popular Web server software. It is fast, reliable and can be extended through a simple API.

2.2 PHP

PHP is a server-side scripting language for web development. PHP can also be used in a general-purpose programming. PHP was created by Rasmus Lerdorf in 1995. PHP originally represent for Personal Home Page, [3] it now stands for Hypertext Preprocessor. [4] PHP is especially suitable for server-side web development. [5] PHP is the most popular web programming language. Up to January 2013, PHP was installed on above 240 million websites and 2.1 million web servers. [6] For feasibility and practice, PHP is a reasonable and suitable choice.

2.3 CakePHP

CakePHP is an open-source web application framework, written in PHP, which follows the MVC approach. Convention over configuration, Model-View-Controller, Active Record, Association Data Mapping, and Front Controller are software patterns used in CakePHP. [7]

CakePHP contains following features: build quickly, batteries included, no configuration, clean MVC conventions, friendly license and secure. It is easy and convenient to use and manipulate. [8] So we choose CakePHP to accomplish this project.

2.4 Android

Android is an operating system mainly for smart phones and tablet computers. It is created by Android, Inc. and bought by Google in 2005. The first smart phone running Android to public is the HTC Dream on October 22, 2008. [9]

Android is based on Linux kernel which is a Unix-like OS (Operation System). Linux kernel is outstanding in free and open source software.
The following diagram is the architecture of Android system:

![Android System Architecture](image)

**Figure 2.1 Android System Architecture**

Android OS can be roughly split into five components and four layers. All the applications can be found and installed in applications layer. The higher-level services which permitted developers to utilize are offered by the application framework layer in Java classes. Android routine layer provides Dalvik Virtual Machine which is as critical as Virtual Machine in Java and a set of core libraries. There are a set of libraries in the libraries layer including SQLite database for storage and sharing of application data, WebKit(an open-source Web browser engine), SSL for Internet Security and etc. The basic system functionality as process management, memory management, device management and etc. is offered by the bottom layer, Linux Kernel. [10] The kernel also handles all the things that Linux is good at such as networking and a vast array of device drivers.

### 2.5 DBMS

DBMS is the abbreviation of Database management systems. DBMSs are specially designed software applications that interact with the user, the database itself and other applications to capture and analyze data. It is a software system designed to enable the definition, creation, querying, update, and administration of databases. [11] In our project we use two kinds of DBMSs including MySQL and SQLite.
2.5.1 **MySQL**

MySql is a popular choice of database for use in web applications. Free-soft-
ware-open source projects often use MySQL to satisfy the requirements of a
full-featured database management system. [12] For commercial use, several
paid editions are available, and offer additional functionality in this project, we
choose to use the free edition because its functionality is enough.

2.5.2 **SQLite**

SQLite is a relational database management system. Compared to other data-
bases management systems, SQLite is an integral part not a separate process. For
its small size, SQLite is very suitable for embedded systems like Google's An-
droid. [13] It is also suitable for desktop operating systems. SQLite has bind-
ings for a large number of programming languages, such as VB, Basic, C, C++,
C# etc.

2.6 **XML**

XML (Extensible Markup Language) is a markup language designed to carry
data. XML is self-descriptive and can be understood by both human and com-
puter. XML tags are needed to be defined by developers. [14] XML is formu-
lated in the XML 1.0 Specification which was made by W3C. [15]

2.7 **UML**

The Unified Modeling Language (UML) is a modeling language. It is designed
to provide a standard way to display the design clearly and visually. [16] In our
project, we will use some UML diagrams to describe the functionalities of the
system and the activities of the system.

2.8 **Cipher**

In cryptography, a cipher is an algorithm which used for performing encryption
or decryption—a series of well-defined steps that can be followed as a proced-
ure. To encipher or encode stands for to convert information from plain text
into cipher or code. [17]
3 Methodology

The first key step for a project development is using proper methodologies and choosing appropriate corresponding methods. This chapter will describe the methods adopted and the developing process of this project.

3.1 RUP

3.1.1 The concept of RUP
RUP is the abbreviation of Rational Unified Process. RUP is an adaptable software development process framework invented by a division of IBM, Rational Software Corp. [18] RUP is designed for effective project management. RUP enables users to select and deploy the components specifically meet the needs of the current project. [19]

3.1.2 Six Best practices
RUP describes that effective development has six best practice as following: [20]

- Develop software iteratively
- Manage requirements
- Use architectures based on components
- Model visually
- Verify quality
- Control changes to software
3.1.3 Process structure

![Rational Unified Process structure](image)

Figure 3.1 Rational Unified Process structure

The Rational Unified Process structure is displayed as figure 3.1. It has two dimensions, organization along time and content. The horizontal axis is time spent in inception, elaboration, construction and transition in a dynamic aspect. The vertical axis stands for activities, artifacts, workers and workflows in a static aspect. Activities, artifacts, workers and workflows are represented for who, how, what, when. [20] The core process workflows has two kinds, engineering and supporting. In engineering workflow, business modeling, requirements, analysis & design, implementation, test and development workflows are included. Project management, configuration and change management and environment workflows are contained in supporting workflows.

3.2 Software prototyping

The original objectives of a prototype is to enable the users to evaluate the proposal of the developers for the design of the final product by actually trying out instead of by evaluating the design based on descriptions. Prototyping can also be applied by end users and it is the crucial factor in the commercial relationship between developers and their clients. [21]

Prototyping contains several advantages as follows. First of all, valuable feedback can be get back to software designer and implementer early in the project. Secondly, the client can ensure whether the software matches the software specification. Thirdly, the software engineer can estimate the accuracy of initial project and keep on schedule. [22]

In the beginning of our project, software prototyping methodology was used to analyze the requirements. After having several meetings with the staff from the
LIFEHOPE.EU program, the prototype was made to specific the requirement. Details will be mentioned in the next chapter.

3.3 MVC

Model–view–controller (MVC) is a software architectural pattern with dividing the code into three interconnected parts.

As the Figure 3.2 describes, the model layer implements the business logic of the application. It is used to obtain the data and convert it into a meaningful concept for the application. The model layer includes processing, validation, association or other mission related to data processing. The view layer is used to present the modeled data. When the Model layer returns data, the view uses the data to generate a HTML page. The controller layer is used to process the request from the users. It is used to deal with the request with the help of both the View and Model layer. [23]

The benefits of the MVC is as follow:

Firstly, attach multiple views to a model to provide different presentations (view/model decoupling)

Secondly, MVC change the way a view responds to user input without changing its visual presentation (view/controller decoupling). [24] The web application is developed by using a MVC developed framework called CAKEPHP.

3.4 DAO

DAO which is the abbreviation of Data Access Object is used to abstract and encapsulate all access to the data source. It takes charges of the connection with
the data source to obtain and store data. It isolates all details of data storage from the rest of the application. [25] DAO allows ripple effects from possible changes to the persistence mechanism to be confined to a specific area.

In our android client, the android activities access the data source which include XML files and SQLite database which modified through the DAO interface.

3.5 Black-box testing
Software testing used to evaluate the quality of the product or service is maintaining during the entire implementation phase. The system testing begins after all the function modules are finished. [26]

Black-box testing is a method of software testing which can be implemented to every virtual level of software testing: unit, integration, system and acceptance. It examines the functionality of an application without peering into its internal structures or workings. Typically it comprises most if not all higher level testing, but can also dominate unit testing.
### Table 3.1 Test case of doctor login

<table>
<thead>
<tr>
<th>Project name</th>
<th>System based on Apache Web Server and Android client for LIFEHOPE.EU program</th>
<th>Test case ID</th>
<th>0001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function module</td>
<td>Login</td>
<td>Author</td>
<td>Zehua Li</td>
</tr>
<tr>
<td>Function Features</td>
<td>User Authentication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testing aim</td>
<td>allowing legitimate login, to prevent unauthorized access</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testing data</td>
<td>username=admin password=LZHlzh523588</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Procedure</td>
<td>Description</td>
<td>Data</td>
<td>Expected results</td>
</tr>
<tr>
<td>1</td>
<td>Input username and password and click login button</td>
<td>username=3333 password=555555</td>
<td>Prompt “Invalid login”</td>
</tr>
<tr>
<td>2</td>
<td>Input username and password and click login button</td>
<td>username=admin password=LZHlzh523588</td>
<td>Login success</td>
</tr>
</tbody>
</table>

Table 3.1 shows an example test case which is used to test the login function module. From this table, the login test case is passed.
4 Design and Implementation

In this section, first the process of requirement analysis is conducted. Then how the Android application and the administrative website are designed and implemented will be presented in detail. At the end of this chapter, part of the Web page and android activities will be navigated.

4.1 Requirement analysis

The requirement analysis consists of two parts: feasibility analysis, and performance requirement.

4.1.1 Feasibility Analysis

The feasibility study is a preliminary comprehensive and integrated research on the project development and the identification.

For the system construction is a big investment, wide-range and complex systems engineering. So a sufficient feasibility study must be done to ensure precise investment. The information system is an integrated project, thus the system obtains convincing proof of its viability. The purpose of the feasibility study is to use the lowest cost, in the shortest possible time to determine which problem can be solved. it is not intended to solve the problem, but to consider whether it is worthwhile to solve the problem. The feasibility contains two aspects, technical feasibility and economic feasibility.

This system is based on Apache HTTP Server and android client. The programming languages include PHP and Java. Also, the required functions are not difficult to implement.

The project involves only hundreds people, so the system has no special requirements for computer hardware. The LIFEHOFE.EU already have a domain and a web server which supports the PHP and MySQL. To save investment, open source technique is used. For it is free and economical.

In summary, technical and economic are feasible. So is the development of this system.

4.1.2 System use case model

A use case diagram is a graphic representation which can clarify how different groups of users interact with system. The system refers to something being operated or developed. It uses UML (Unified Modeling Language), planning overall requirements, validating a hardware design, testing and debugging a soft-
ware product under development, creating an online help reference, or performing a consumer-service-oriented task are all included in system objectives. [27]

Table 4.1 meanings explanation of use case model

<table>
<thead>
<tr>
<th>Character</th>
<th>Name</th>
<th>Actor</th>
<th>Association</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1 illustrates the meanings of every component in use case model.

Figure 4.1 use case diagram of doctor

Figure 4.1 shows the use cases of the doctor. After the doctor logs in the web application with the username and password assigned by the administrator, he or
she can manage the patient and the recipe and the schedule. Management includes four basic functions which are create, read, update, delete. After the doctor creates a recipe, measure, schedule which can be sent to the patients who use the username and password to obtain the data from the web application through the android device. Additionally, the doctor can view patients feedback.

![Use Case Diagram of Patient](image)

**Figure 4.2 the use case diagram of patient**

In this part, we will introduce the how the android application works. Figure 4.2 shows in the use cases of the android application. After the patients get the username and password, they can get the data from the web application and set the preference to decide if they want to accept the schedules which the doctor sends to him as well. If not, they can manage their own schedule.
4.1.3 Prototype

In this chapter, the outcome will be displayed with several screen shoots of the website and the app.

Figure 4.3 main page of prototype web application

Figure 4.3 shows the web page of managing patients. As the screen shoot shows, the functions of this website provided to the users contains the main page, manage patients, new patient, manage recipes, new recipe, manage schedules, new schedule, manage group, new group. (Group is designed for doctors to manage patients in a more efficient way.).
Figure 4.4 measures management page

Figure 4.4 shows the measures management page. One measure has several steps. After the measure is sent to group, the relation between the group and measure will be shown in this page.

Figure 4.5 main page of the android prototype

This is the main menu that can be seen after login successfully. Users can touch sports schedule, recipe, and setting buttons to launch the activities.
4.1.4 Requirements Specification

In the initial stages of our project, we did a rigorous requirement analysis. At first we do not have a detail document of the requirements. All we had is a rough view. After several meetings the prototype was made, the staff from LIFEHOPE.EU program and we analysis the requirement through the prototype together. Finally, the requirement specification was finished.

Website: doctors can manage patients, including add, delete, modify. Patients were divided into different groups. Each patient has his or her own unique schedule and recipe which means that a weekly task desire to finished by the patient. Each group can have the same measures used when facing nervous and depressed and books about a health lifestyle. For example, some relaxing video measures can help appease the patients. Doctors can design tips and assigned to different groups.

Android: the patient's android device can automatically receive schedule, recipe, book and measure. Measures are videos or audios which will consume very large data flow. Hence, only when the WIFI is available these measures can be downloaded. If the patient completes the recipe on time, the app automatic feedback to the site, so that the doctor can see the completion time, in order to arrange next week’s recipe more properly. The patient also can write some feedback and send to the doctor.

To adapt to the actual work of managing the system, performance of the system requirements are as follows:

1. Security requirements: use of the system to authenticate the user needs only legitimate users can use the system.

2. Responding time requirements: The website’s page load time should be less than 3s. The android’s responding time should be less than 1s.

4.2 Design

4.2.1 System Architecture

The system architecture will be illustrated as follows.
Figure 4.6 Overall system architecture

The overall system architecture is shown in Figure 4.6. The system can be divided into three parts: android client, web server and database server. The android client sends request to web server and then the web server responses the result which is returned from the database server.

Figure 4.7 Web Server Architecture

Figure 4.7 shows the architecture of the web server. The web server provides two interfaces. The first is authentication which is used to authenticate the android client user request. If the android client send the correct username and password, it will get the ID or it will get the failed message. The second is data request interface. After the android client gets the ID, client can get data from the web server thorough data request interface.
Figure 4.8 overall design of the android application

Figure 4.8 shows the overall system design of the android application. The activity call the data access object to process the data and then provide the data to user. The data received from the web application all are inserted into the SQLite database by using DAO interface.

4.2.2 Functionality structure

Figure 4.9 functionality structure of website

Figure 4.9 shows the functionality of the website. The basic functionality includes managing patient’s information, managing schedules, managing recipes, managing measures and managing feedback.

Manage patient’s information: The doctors could manipulating patients’ information by adding, deleting, altering and searching. Every patient has a unique
username. All the input data should be validated, for example, phone numbers should be consists of numbers not includes alphabets. If the data are invalid, the website will alert the doctors and request validate data.

Manage Schedule: To encourage patients to do physical exercises, the doctors could make a plan for them. Schedules could be set once or every month or every week or every day. Begin date is set in a schedule. If this schedule hasn't begun, it will not be seen by patients in Android application.

Manage Recipe: In LIFEHOPE.EU program, the doctors weekly have a meeting with their patients and ask what he or she wants to accomplish in the next week. It is a weekly goal which LIFEHOPE.EU called it as recipe. Every week, each patient only have one recipe. The doctors add the new recipe related to the patient. If the web server receive an information that it has been finished, SQL database in web server will be accordingly updated and the doctors could see it through manage recipe. The finishing time may assist the doctors to determine which kind of recipe is more suitable for this patient.

Manage Measures: Stress management is offered in the system. Video and audio files can be uploaded and combine in a stress measure. When the patients are overwhelmed by nervousness or depression, these measures will help them to overcome these feelings.

Manage Feedback: This function enables the doctors view patients different kinds of feedback on measure, recipe and schedule. With the help of feedback, doctors could made more accurate instructions.

Figure 4.10 describes the functional diagram of android client. Patients, can look up schedule made by his or her doctor and weekly task and follow measures to relax. They can also write feedback about their feelings of recipe, schedule, stress measure or other.

Sports schedule: Patients could set their preference to whether accept schedule made by their doctors or not. If not, they could manage their own schedule.
Recipe: Patients would receive a firework as encouragement when they finish recipe. If the status is accidentally changed to finished, they could change it to unfinished.

Measure: When nervousness or depressing, measure may cheer them up. After a measure selected, it will automatically and sequentially open media files in this measure.

Book: In the Book section, books written by LIFEHOPE.EU program about healthy life habits Book are available which could let the patients get to know these knowledge and live in a fit life.

Feedback: Comments on the Android application will be sent through feedback and then to the web server. These could be read on web application by doctors.

![Android Service Diagram](image)

Figure 4.11 android app service

Figure 4.11 shows the functions of the service. The service of the android application is running in the background. Periodically interacting with the web application to get the data and then use the DAO interface to insert the data into the SQLite database. If there are any changes in the today’s schedule, the alarm will be updated. Between 1pm and 3pm, it will notify the user if the user has not finished. The service also posts data which includes the feedback data and the recipe state data to the web server.

4.2.3 Flowchart of system

Figure 4.12 flowchart of the website

Figure 4.12 shows the flowchart of this system. After the doctor opened the site, he or she needs to enter a user name and the password. If the username and password are correct, the user can enter the management interface. At this time, doctors can manage all the information. Just click on logout, user will be able to exit the website system.
First of all, the user already has an account number and password from the doctor. Figure 4.13 shows the flowchart of the android app. When the user open the app, user need to input the username and password. If the username and password is not correct, they will get error prompt and can retry to input username and password. If login successful, the user can start to use this app. When clicking Exit, the user exits the android app.

Figure 4.13 flowchart of the app
4.2.4 Database Design

Figure 4.14 shows the main database structure of the web application. Since the data is dynamic, the input data collected from doctors must be stored where it can be read and manipulated. In the web application, the back-end data stored in MYSQL which is a relational database management system (RDBMS) for an organized collection of data is provided. PHP scripts are used to establish a connection with the database, and then use the SQL statement to create, update, read, and delete records. From the figure 4.9 it is shown that the doctor can have many measures, recipes, groups, schedules, and patients. A patient belongs to a group and can have many recipes and schedules. The group can have many measures and patients.

Table 4.2 doctors

<table>
<thead>
<tr>
<th>Table: doctors</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>username</td>
</tr>
<tr>
<td>passwaord</td>
</tr>
</tbody>
</table>

As shown in Table 4.2 doctors is used to store the profile of doctors. The username and password are used to verify when the doctor logins on the administrative website.
A Web- and App- Based Interaction System Bridging the Gap between Patients and Doctors
Zehua Li, Dan Luo

Table 4.3 patients

<table>
<thead>
<tr>
<th>Table: patients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>username</td>
</tr>
<tr>
<td>Password</td>
</tr>
<tr>
<td>groupid</td>
</tr>
</tbody>
</table>

As shown in Table 4.3, table patients is used to store the profile of patients. The username and password are used to verify when the patient login on the android device.

Table 4.4 recipes

<table>
<thead>
<tr>
<th>Table: recipes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Week</td>
</tr>
<tr>
<td>Content</td>
</tr>
<tr>
<td>Success</td>
</tr>
<tr>
<td>PatientId</td>
</tr>
</tbody>
</table>

As shown in Table 4.4, table recipes is used to store the profile of patients. The week stands for which week of this year. The content describes the task of the week. The success is used to indicate whether the patient finish this week’s task.

Table 4.5 schedules

<table>
<thead>
<tr>
<th>Table: schedules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Begin</td>
</tr>
<tr>
<td>End</td>
</tr>
<tr>
<td>Begintime</td>
</tr>
<tr>
<td>Endtime</td>
</tr>
<tr>
<td>PatientId</td>
</tr>
</tbody>
</table>

As shown in Table 4.5, table schedules is used to store the information of schedules. The field begin and end stands for the valid date. Begin time and end time stands the time of the day.
Table 4.6 steps

<table>
<thead>
<tr>
<th>Table:steps</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Int</td>
<td>Primary Key</td>
</tr>
<tr>
<td>Name</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>Url</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>Measureid</td>
<td>Int</td>
<td>Foreign Key</td>
</tr>
</tbody>
</table>

As shown in Table 4.6, table steps is used to store the information of upload file. The field url stands for the path of the upload file.

Table 4.7 comments

<table>
<thead>
<tr>
<th>Table:comments</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Int</td>
<td>Primary Key</td>
</tr>
<tr>
<td>Content</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>Patientid</td>
<td>Int</td>
<td>Foreign Key</td>
</tr>
</tbody>
</table>

As shown in Table 4.7, table comments is used to store the information of feedback. The field content stands for the feedback.
Figure 4.15 database design of android application

Figure 4.15 shows the database structure of the android application. The comment, schedule, measure and step have the same fields with the tables of web application.

Table 4.8 alarm

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Int</td>
<td>Primary Key</td>
</tr>
<tr>
<td>Time</td>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>Scheduleid</td>
<td>Int</td>
<td>Foreign Key</td>
</tr>
</tbody>
</table>

A schedule generates two alarms one is fired in the beginning of the schedule, the other is fired in the end of the schedule. As the Table 4.8 showed, the time store the exact time when the app should remind the user of the schedule.

4.3 Implementation

During this phase, we focuses on program and application development task. Its tasks are programming and application development task. Our main working is coding.
4.3.1 Naming convention

Convention allows other developers to understand and help much more easily for its uniform system development. CakePHP's convention is concentrated from years of best practices and web development experience.[28]

The requirements of controller class name are plural, CamelCased, and end in Controller, for examples, PeopleController and LatestArticlesController. The default CakePHP first execute is the index() method.[28] Furthermore, Changing the visibility of controller methods in CakePHP is allowed by prefixing controller method names with underscores.

Generally, file names match the class names. If a class in CakePHP named MyNifflyClass, the file should be correspondingly named as MyNifftyClass.php. [28] Each file would be located in the appropriate folder in your app folder.

Model class names are singular and CamelCased, for instance, Person, BigPerson, and ReallyBigPerson. Table names which should match to CakePHP models are plural and underlined. The table names for the examples above respectively are big_people, and really_big_people.[28] The utility library Inflector can be used to check the singular/plural of words.

View template file are named after the controller functions which they display. The basic structure is like this, /app/View/Controller/underscored_function_name.ctp. For example, the getReady() function in PeopleController class will look for a view template in the path /app/View/People/get_ready.ctp. Through these convention, CakePHP knows that a request which is to http://example.com/people/ maps to a call on the index() function of the PeopleController. Person model is available in PeopleController automatically and renders to a file. [28]

4.3.2 Plugin and open source

Uploader is a CakePHP plugin that will validate and upload files through the model layer. Provides support for image transformation and remote storage transportation. Uploader makes it easier to upload file and associate the database record just need some configurations. There are two kinds configuration. The first is called AttachmentBehavior. In the AttachmentBehavior, you can configure where the file uploads to, which database field association with the information (including file size, file final upload path, file mime type) of the upload file. The second is called FileValidationBehavior. FileValidationBehavior defines certain rules to validate the file. For example the file size cannot be more than 20MB. The file mime type have to be video or audio. The file extension must be included in the extension list. Uploader supports file deletion when a database record is deleted or updated. [29]

Vitamio is an open multimedia framework for Android and iOS. Vitamio is the most popular choice among developers, because of its simple, clean and powerful API. The types that Vitamio can play includes 720p/1080p HD mp4, flv,
rmvb, rm, mkv, avi, m4v, mov, ts, tp and many other video formats in Android and iOS. [30] It is easy to use as a default MediaPlayer in Android but accompanied with some fancy capabilities.

### 4.3.3 Privacy

Nowadays SQL injection is a very popular attack method. SQL injection is a code injection technique. It is used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution. [31]

In order to forbid SQL injection, we use following methods.

In most case, we use the methods provided by CakePHP Model::find() and Model::save(). CakePHP handles SQL escaping on all parameters to Model::find() and Model::save(). [32] In the rare case when we need to construct SQL by hand using user input we use Prepared Statements.

The parameters to prepared statements don't need to be quoted; the driver automatically handles this. [33] If an application exclusively uses prepared statements, the developer can be sure that no SQL injection will occur.

![Cipher encryption and decryption diagram](image)

**Figure 4.16 Cipher encryption and decryption**

As the Figure 4.16 shows, the patient's private information will be stored into the database. A database server which is rented from a company called citynetwork is used. Although the server agents are advertising their servers are absolutely secure, in order to prevent the leak of private information, it is still necessary to take an approach to encrypt the private information.
As the Figure 4.17 shows the private information will first be encrypted by cipher and then be stored into the database. The original data only can be seen by doctors.
4.4 **Web page and android activities**

Figure 4.18 activities of the Android app

Figure 4.18 illustrates the main activities on Android application. After the user successfully login. Due to the fact that our Android users are people with serious mental illness, buttons which refer to recipe, settings, measure, book and sports schedule are placed in the main menu. Every button contains words and related pictures to ensure users understand to the meaning. After clicking each button, the Android will navigate into corresponding activities.
Figure 4.19 presents how a doctor create a measure. In this case, the doctor create measure name depression which has two steps. When the patient feels depression or nervousness, he or she can use this measure.
Figure 4.20 the measure display in the android application

When patients click a measure, the app will display the video or audio files of this measure. Figure 4.20 shows an example of a measure named depression. This measure can be used when a patient feels depressed. The patient follows the steps to make themselves better. At first, the patient takes the first step to do deep breath. After the deep breath video is over, a funny video will be offered. Hopefully, the patient can get help from the measure.
A Web- and App- Based Interaction System Bridging the Gap between Patients and Doctors
Zehua Li, Dan Luo

2014-6-20

Figure 4.21 patient’s information page

Figure 4.21 shows the patient’s related information page. From this page, the doctors can look up and manage patient’s recipe, schedule and comment.

Figure 4.22 Recipe and Schedule adding page

Figure 4.22 shows the pages how doctors create a new recipe and a new sports schedule for a patient.
Figure 4.23 the schedule activities

Figure 4.23 displays the schedule information activity after clicking sports schedule button on the main menu. When the schedule begins or ends, the android device will remind the patient by vibrating and ringing. The vibrating and ringing will be disappeared if the patient clicks the “close the alarm” button. If the patient does not close alarm, the alarm will be closed automatically after 1 minutes.

Figure 4.24 the recipe activities

Figure 4.24 illustrates how the Android application reminds the patient of finishing weekly recipe. If the patient clicks the “Yes” button, the application will show the reward activity which is applause sounds and a firework animation. If
the patient clicks the “No” button, the app will encourage the patient by toasting an encouraging sentence.

Figure 4.25 the feedback activity

Figure 4.25 displays the feedback page. The patient can write four kinds feedback contain "measure" type, “recipe” type, “schedule” type and “other” type. The physician can improve the measure, sports schedule, recipe according to the patient’s feedback. The “other” type feedback refers to the user experience feedback of this application which is useful for developers to upgrade the app.
Figure 4.26 Book activities

Figure 4.26 shows how the Android application displays the book section after the patient click the “book” button. After clicking the book name, the book will be presented in a web view which can be zoomed in and out so that no matter how small the screen is, the book can be viewed very clearly.
5 Results

5.1 Test environment

Table 5.1 web server test environment

<table>
<thead>
<tr>
<th>Operate system</th>
<th>Microsoft Windows 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Frequency</td>
<td>2.40GHz</td>
</tr>
<tr>
<td>Memory Information</td>
<td>1.93 GB</td>
</tr>
<tr>
<td>Web Server</td>
<td>Apache/2.2.22(Win32)PHP/5.39</td>
</tr>
<tr>
<td>Server domain</td>
<td>192.168.0.3</td>
</tr>
<tr>
<td>Server IP</td>
<td>192.168.0.3</td>
</tr>
<tr>
<td>Server Port</td>
<td>80</td>
</tr>
</tbody>
</table>

Table 5.2 android test environment

<table>
<thead>
<tr>
<th>Operate system</th>
<th>Android 4.0, Android 2.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Frequency</td>
<td>1.2GHZ</td>
</tr>
<tr>
<td>CPU pattern</td>
<td>ARMv7 Processor rev 10</td>
</tr>
<tr>
<td>RAM</td>
<td>694.1M</td>
</tr>
</tbody>
</table>

Table 5.1 shows the hardware and software configuration of the test environment. Table 5.2 describes the test environment of the android test.
5.1 Security evaluation

![Figure 5.1 Common high risk vulnerabilities](image1.png)

![Figure 5.2 vulnerabilities of our web application](image2.png)

In order to evaluate the security of our web application, the Acunetix is used to scanner the web application. In the Figure 5.1 and Figure 5.2, the red vulnerability is high risk and yellow vulnerability is medium risk and blue vulnerability is low risk and green information is safe.

The Figure 5.1 shows the high risk vulnerabilities of a sample web application which is scanned by Acunetix. [34] The security aim of this project is to protect the patients’ private information. Usually, attacker may take advantage of several high risk vulnerabilities to get the access to privacy.

Firstly, The most common approach is use the SQL injection. The arbitrary SQL statements can be executed by an attacker on the vulnerable system. The sensitive information may be exposed to the attacker. [35]

The second is to use the unprotected phpMyAdmin interface. PhpMyAdmin is an application that provides a web-based interface for the administration of MySQL databases. The password of the initial MySQL root account is empty.
So, attacker can connect to MySQL server by using the root account without password. Then the attacker can get all the sensitive information. [36]

The third is Script source code disclosure. An attacker can collect sensitive information which include database connection strings or application logic by analysing the code source. If an attacker get the database connection, the attacker can access the database, and then get sensitive information. [37] This information can be used to launch further attacks. We could analyse the source code of this script and solve the problem.

The Figure 5.2 shows the scanning result of the web application. Comparing with the sample scanning result shown in the Figure 5.1, it can be concluded that the web application cannot only prevent the SQL injection, unprotected phpMyAdmin interface and script source code disclosure but also prevent other common vulnerabilities. The main Vulnerability is that the User credentials are transmitted over an unencrypted channel. A third party may be able to read the user credentials by intercepting an unencrypted HTTP connection. This problem can be fixed by using an encrypted connection (HTTPS).

The patient who using the android client only can only get access to his or her own data. Only the care-givers can view data of all patients under their watch.

5.2 Responding time evaluation

To achieve a reasonable user experience, a pleasant responding time is required. It is expected that the responding time less than 1s on the Android application and load time on the web pages is less than 2s.

5.2.1 Android responding time evaluation

<table>
<thead>
<tr>
<th>Schedule (ms)</th>
<th>Recipe (ms)</th>
<th>Measure (ms)</th>
<th>Measure open files(ms)</th>
<th>Book (ms)</th>
<th>Book open files(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>508</td>
<td>383</td>
<td>133</td>
<td>570</td>
<td>72</td>
</tr>
<tr>
<td>2</td>
<td>117</td>
<td>109</td>
<td>101</td>
<td>578</td>
<td>101</td>
</tr>
<tr>
<td>3</td>
<td>109</td>
<td>125</td>
<td>148</td>
<td>352</td>
<td>94</td>
</tr>
<tr>
<td>4</td>
<td>109</td>
<td>242</td>
<td>110</td>
<td>368</td>
<td>101</td>
</tr>
<tr>
<td>5</td>
<td>110</td>
<td>109</td>
<td>117</td>
<td>344</td>
<td>103</td>
</tr>
<tr>
<td>Average (ms)</td>
<td>190.6</td>
<td>193.6</td>
<td>121.8</td>
<td>448.4</td>
<td>94.2</td>
</tr>
</tbody>
</table>

Figure 5.3 logcat
The responding time is tested by using the logcat which is a tool of the eclipse. Through using the Log.v() method, to log the beginning time when the click button event and the end time when the new activity finish initial. Figure 5.3 shows one example test of the responding time. In this example, the beginning time is 01:23:27:138, the end time is 01:23:27:325. So, the responding time of this example is 187ms.

Table 5.3 displays the android application responding time. None of these average responding times is more than 1 second. Therefore, the responding time requirement is satisfied.

### 5.2.2 Web application responding time evaluation

#### Table 5.4 evaluate of web load time

<table>
<thead>
<tr>
<th>URL</th>
<th>Average(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Main Page</td>
<td>0.8</td>
</tr>
<tr>
<td>2 Patients</td>
<td>1.60</td>
</tr>
<tr>
<td>3 Patients/view/Web average time33</td>
<td>1.54</td>
</tr>
<tr>
<td>4 Comments/Index/33</td>
<td>0.66</td>
</tr>
<tr>
<td>5 Books</td>
<td>0.89</td>
</tr>
<tr>
<td>6 Recipes/add/</td>
<td>0.58</td>
</tr>
<tr>
<td>7 Schedules/add</td>
<td>0.52</td>
</tr>
<tr>
<td>8 Groups</td>
<td>0.69</td>
</tr>
<tr>
<td>9 /groups/view</td>
<td>0.69</td>
</tr>
</tbody>
</table>

In order to evaluate the web page load time of the web application, a tool called WebWait is adopted which is used to benchmark website or test the speed of web connection. WebWait pulling down the entire website into the browser which means it takes into account Ajax/JavaScript processing and image loading. So, the timing of WebWait is more accurate than other tools which often ignore the Ajax/JavaScript processing and image loading. [38] Table 5.4 shows the average web page load time of the web application. All the average load time of them are less than 2s. Thus, the web page load time satisfy the requirements.
6 Conclusions

As a part of LIFEHOPE.EU program, this project are able to make the program more convenient for those people living in remote areas with SMI. It is difficult to establish a bridge between patients and doctors. Thus we develop a system to solve this problem.

The project targets at developing a system included an Android application for the patients and an administrative website based on Apache web server for doctors to assist patients’ recovery and doctors’ treatment. Through strictly following RUP method step by step, the system has been carried out and fulfill the requirements. To evaluate the security of the system, Acunetix Web Vulnerability Scanner was implemented to scan the vulnerabilities of the web application. Additionally, evaluations were conducted on responding time of both Android application and web pages to make sure it meet demands.

Through this system, doctors can set schedule for patient, manage patient's information, send measures to patients, send recipe to patient and check patient's feedback. Moreover, patients could choose to accept the schedule sent by doctor or manage their own schedule. The system fairly meets the demand. The interface of this system is user-friendly and easy to operate by both doctors and patients.

The LIFEHOPE.EU program has been satisfied with the final outcome and decided to put it into use. They are going to install the Android application on patients’ Android devices. And then the system of this project will benefit both the patients and doctors participated in LIFEHOPE.EU.

From the aspect of ethical issues, the outcome system of this project assists LIFEHOPE.EU program research. To prevent the misuse of the system we have created and the privacy of the users, the important and private information are encrypted.

The future work will continue to addressing our efforts on security protection, user interface optimization and system function perfection. In the security protection aspect, the user credentials will be transmitted over an encrypted channel by adopting HTTPS. In view of user interface optimization, the interface will be more user-friendly and rewarding animation will be more vivid so that patients can gain more confidence and self-accomplishment from it. Moreover, after this project being put into service for a while, the LIFEHOPE.EU program will send back how this system is and which functions are looking forward to improve.
A Web- and App- Based Interaction System Bridging the Gap between Patients and Doctors  
Zehua Li, Dan Luo  
2014-6-20

References

Description of the intervention and results.  
Accepted March 2014 in Journal of Psychiatric and Mental Health Nursing.

[2] Information about the Apache Web Server, ntchosting.com  
http://www.ntchosting.com/apache-web-server.html  
Retrieved May, 2014

Retrieved May, 2014

Retrieved May, 2014

http://en.wikipedia.org/wiki/Php  
Retrieved May, 2014

Retrieved May, 2014

http://en.wikipedia.org/wiki/Cakephp  
Retrieved May, 2014

http://cakephp.org/  
Retrieved May, 2014

Retrieved May, 2014
A Web- and App- Based Interaction System Bridging the Gap between Patients and Doctors
Zehua Li, Dan Luo

Retrieved May, 2014

Retrieved May, 2014

http://en.wikipedia.org/wiki/MySQL
Retrieved May, 2014

http://en.wikipedia.org/wiki/SQLite
Retrieved May, 2014

[14] "XML and Semantic Web W3C Standards Timeline". 2012-02-04

Retrieved May, 2014


http://en.wikipedia.org/wiki/Cipher
Retrieved May, 2014

[18] “IBM Acquires Rational”
Retrieved May, 2014

Retrieved May, 2014

http://www.ibm.com/developerworks/rational/library/content/03July/00/1251/1251_bestpractices_TP026B.pdf
Retrieved May,2014

A Web- and App- Based Interaction System Bridging the Gap between Patients and Doctors
Zehua Li, Dan Luo 2014-6-20


[23] CakePHP, “Understanding Model-View-Controller”
Retrieved May, 2014

[24] “What are the benefits of MVC?”, 9 December 2008 by Ian Davis
Retrieved May, 2014

[25] “Data access object”
http://www.oracle.com/technetwork/java/dataaccessobject-138824.html
Retrieved May, 2014


[27] “What is use case diagram?”
http://whatis.techtarget.com/definition/use-case-diagram
Retrieved May, 2014

[28] “CakePHP Convention”
Retrieved May, 2014

[29] “code->uploader”
http://milesj.me/code/cakephp/uploader
Retrieved May, 2014

https://www.vitamio.org/en/
Retrieved May, 2014

[31] Microsoft. "SQL Injection".

[32] “Data Sanitization”
Retrieved May, 2014

44
A Web- and App- Based Interaction System Bridging the Gap between Patients and Doctors
Zehua Li, Dan Luo

[33] “Prepared statements”
http://se1.php.net/pdo.prepared-statements
Retrieved May, 2014

[34] “Why choose acunetix”
http://www.acunetix.com/why-choose-acunetix/
Retrieved May, 2014

[35] “sql injection”
http://www.acunetix.com/websitesecurity/sql-injection/
Retrieved May, 2014

[36] “Unprotected phpMyAdmin interface”
http://www.acunetix.com/vulnerabilities/unprotected-phpmyadmin-in/
Retrieved May, 2014

[37] “Script source code disclosure”
http://www.acunetix.com/vulnerabilities/script-source-code-disclo/
Retrieved May, 2014

[38] “WebWait”
http://webwait.com/
Retrieved May, 2014