Linnéuniversitetet
Kalmar Växjö

Bachelor thesis

# Develop Mobile Hybrid Application with Ionic and Angular.js

*Suitable for further development?*

*Author:* Anton Mösenbacher
*Supervisor:* Anders Haggren
*Examiner:* Anders Haggren
*Supervisor, company:* Erik Stenberg, Softhouse Växjö
*Date:* 2016-05-27
*Course code:* 2DT00E, 15,0 hp
*Topic:* Computer Engineer
*Level:* Bachelor

Department of Technology
Faculty of Technology

# Summary

This report presents a proof- of-concept regarding hybrid mobile application and its development, based on Ionic with the implemented framework Angularj.js for web and mobile application development.

Regarding the background of requests from APEA-Hotworks to Softhouse Växjö, a complementary application for the convenience of how hot work is carried out in industry. Based on a specification from the customer, a hybrid mobile application, using Softhouse Växjö and their Agile approaches to realize this vision.

The report is based on the explanation mentioned above, account for the difference in the production and implementation of basic elements of hybrid and native- components, particularly Android, are implemented. Finally, a conclusion is drawn as to which of them different development methods, hybrid or native, are best suited for further development.

# Sammanfattning

Denna rapport presenterar ett proof-of-concept avseende hybrid mobil applikation och dess utveckling, baserat på Ionic med det implementerade ramverket Angularj.js för web och mobil applikationsutveckling.

Detta mot bakgrunden av ett önskemål från APEA-Hotworks till Softhouse Växjö, om en kompletterande applikation för att underlätta för hur heta arbeten genomförs inom industrin. Baserat på en kravspecifikation från kunden, ska en hybrid applikation, med hjälp av Softhouse Växjö och deras Agila tillvägagångssätt förverkliga denna vision.

Rapporten kommer utifrån förklaringen nämnd ovan, redovisa skillnaden för hur tillverkning och implementation mellan grundläggande element i hybrid och native- miljöer, främst Android, genomförs. Slutligen kommer en slutsats dras om vilket av dem olika utvecklingssätten, hybrid eller native, som lämpar sig bäst för vidareutveckling.

# Abstract

This report presents a proof-of-concept regarding hybrid mobile application development and mobile phones. Regarding the background of a desire to simplify and modernize of how hot work is carried out in in industry.

A conclusion on future developments regarding a hybrid or native options will also be deducted.

*Keywords: native, hybrid, mobile application, future development, Agile, Scrum.*

# Preface

*I want to begin by thanking Henric Westergren with personnel at Softhouse Växjö (2016), for letting me realize my thesis and for all support and help during my time with you. By showing enthusiasm and dedication, you have given me the inspiration to develop myself in programming and web development.*

*Finally, I would also like to thank my family for your near-limitless patience, your understanding, your encouragement and your willingness to support me through, inter alia, proofreading. You have made it possible for me to carry out this work and I am forever grateful!*

# Table of contents

# 1. Introduction

Many would describe the man as lazy in today's digital society, then you are constantly striving to find ever simpler and more flexible solutions to today's complex problems. Personally I would rather describe the man as comfortable, as the todays technology allows for ease of access and flexibility. That was why APEA – Hotworks, contacted Softhouse Växjö (2016), for them in a convenient and easy way to realize a vision of hot work in today's digitized industrial society to become more efficient, easy to use, increment the quality, but also to prevent fires in the workplace. As the permission and checklist of hot work is done manually and follow-up is lacking in several businesses, has APEA - Hotworks with its product found a way to digitize and increase the quality.

During my last year at Linnaeus University in Växjö, where I was studying to be a computer engineer, I did my thesis on Softhouse. The first official meeting between Softhouse and APEA–Hotworks happened on my first day as an intern. Thus I came to be a big part of the process to create a mobile application in line with APEA-Hotworks wishes.

Just as APEA-Hotworks, there are many other companies, but also private stakeholders who´s seeking integrated solutions of consulting firms to develop mobile applications. In addition to web application is native and hybrid the most established and therefore the most common application types. Based on the present finding, bottoms this report to compare two of the most used application types.

The report´s focus is in the area of native and hybrid mobile application, the Agile movement and the Agile methodology, Scrum, in a consulting firm. Based on the report´s background defines the problem area, that is the basis for the report´s problem and its boundaries.

## 1.1. Background

The well-known expression " Rome was not built in a day" indicates that nothing magnificent is done from one day to the other. There is a story behind every technological marvel that reveals years of development, innovation and updates to the modernity that prevails. With this in mind, we can take a step back to briefly see how the application's development curve has been like since the IBM Simon came into the world.

IBM Simon [1], which is said to be the first official smartphone was introduced 1992 and available to consumers August 16 1994. Simon was preloaded with several features, or mobile applications as we say today. Some of the preloaded mobile applications was: Address Book, Calculator, Calendar etc. Since there was no app marketplace where you could purchase

other mobile applications, the number of mobile applications was quite limited at this time.

The next big event that made an impression in world of mobile applications, was when Research in Motion Limited (RIM) [2], released the first BlackBerry in 2002, which later developed to become the first mass-produced smartphone, optimized with an innovative wireless email and with voice and data support.

During this time, small app marketplaces, like Handango, struggled to survive in a small and relatively unknown world for smartphone users. It was not until Apple´s cofounder, Steve Jobs (1955-2011), introduced the first generation iPhone and IOS on his keynote speak in January 9, 2007 [3], as the mobile application industry took off. Steve Jobs created a need for mobile applications, which users themselves, did not know they needed. But it was not until July 2008, when Apple launched the App Store and the new iPhone 3G, as the mobile application sales really took off [4]. A few months later, Google launches their own app marketplace, Android market, or Google Play, as it is now called and HTC releases the first commercially available Android smartphone [5].

This is said to be the start of mobile applications purchasing and it is estimated a total of 44 billion mobile applications during 2016 will be downloaded so far [2].

This overall success would not have been possible if it were not for the underlying technology, such as those different operating systems that began to emerge in the twenty first century, which made it possible for third-party developers to create, but also sell their products in any of those various app marketplaces [6].

Until today, the development of mobile applications has led us to three different types of apps, namely native, web and hybrid [7].

This reports main research area will focus on the construction of a native and hybrid mobile application in a first edition, using web technologies relatively new to the market and investigate different elements between hybrid and native mobile applications for Android.

## 1.2. Purpose and objectives

The purpose with this comparative study aims to give an insight to develop a native and hybrid mobile application in a first stage edition, but also to show differences in how to create different elements with current techniques. The report will also address the Agile Scrum methodology of working at an early stage in a project.

## 1.3. Limitations

The report refers only to treat the technologies and practices that are deemed to have a broad consensus on how to produce a hybrid application of a consulting firm for the modern mobile phones. This means that parts of the report that deals with native and hybrid applications, will only cover the basic knowledge of programming and development. The prototype is the basis for the report's proof-of-concept, is only intended to cover Android based devices, with a focus on the graphical user interface. The implementation made by the techniques and methods with respect to Android, should only be viewed as an alternative solution and not a final approach.

# 2. Theory

In order to understand the report's result and discussion, the section intends to present the necessary and relevant theory in the field of native and hybrid mobile application, the Agile movement, and the Agile methodology, Scrum. This presentation includes mainly only theoretical areas that are directly necessary to understand the big picture, which means that areas that advance has been limited not included in the section.

## 2.1. Applications

In this section gives a short description about those various mobile application alternatives.

### 2.1.1. **Native application**

Native applications are developed specifically for one operating system or platform and are downloaded and installed through app marketplaces. Since the application is all native, they have full access to the device´s functionality like camera, accelerometer, contacts and so on. To develop a native application, you need some of the existing development environment, like Android Studio for Android or xcode7 for iOS. Native applications work even if the device do not have any internet connection. Unfortunately, to develop and maintain a native application that will work on several platforms, has proven to be expensive, since native applications exists in several versions [7].

### 2.1.2. **Web application**

A web application is an application that only exists in the browser and should not be mistaken for genuine native applications. An ordinary web application is built on web technologies, such as HTML, JavaScript and CSS and since it runs in the browser, you need internet connection to access them. A web application has almost the same graphical appearance and feel as a native application, but are limited of accessibility to the device´s core functionality. Web applications are not installed on the device, but can be accessed through bookmarks located on device´s home screen to a specific URL [7].

### 2.1.3. **Hybrid application**

Hybrid applications consists of the best from two worlds, namely native and web applications. Hybrid applications have the graphical appearance, responsiveness and device specific functionality from native, but have maintainability and cross platform technologies from web applications. This

has proven to be a cheaper alternative than native applications, since hybrid does not require several development teams to reach several platforms. Hybrid applications is available in those different app marketplaces [7].

## 2.2. Agile

This section intends to give the reader an introduction to the Agile movement and the Scrum methodology.

### 2.2.1. Brief history

The Agile approach arose in the 1970s, as a result of a fast and dynamically changing software development market, where the industry changed more promptly than they could deliver a finished product to customer, in which requirements, systems and sometimes entire businesses changed during the development period. Until 2000, the waterfall principle was the most used methodology, but was considered inflexible to changes in the development process [8].

### 2.2.2. Agile manifesto

During a meeting in Utah 2001, a group of Extreme Programming pioneers and well known in the Agile community, Agile as a practice, came to shape in what to become today's Agile methodology, the Agile manifesto. [8].

The Agile manifesto is an academic proclamation, consisting of twelve basic principles and four key values of how an iterative, people-centric software development, is to be used daily. By satisfying customer through continuous delivery in an early stage, welcome changes in develop process at any time, motivate individuals with trust and environmental support, and by maintaining regular reflections of the team’s work is just some of the principles in the Agile manifesto [9].

### 2.2.3. Agile software development

Agile software development derived from traditional approaches (waterfall, sequential etc.) where planning, documentation, unrealistic project plans, specific roles and predictability prevent to actually deliver results to customer [10].

Agile software development intends through iterative software development and collaboration between cross-functional teams, quickly give customers access to high-quality and bits of working software as soon as they´re ready. Instead of big pieces of a project at once.

The developers, in return, will receive quick feedback on their work. Rapid feedback and the desire to quickly respond to feedback, is one of the key

characteristics. If the developers are not confident in what customer wants, can they form an approximation of the customer's requirements. If this approximation does not satisfy the customer´s need, the misunderstanding can be resolved quickly, thanks to the feedback from the customer [8].

This approach is based on small, but simple methods that forms the basis for Agile software development. i.e. it focuses on being flexible to adjustments that may occur during the development process, methodical testing of software, simple but applicable code and continuous delivery [11].

### 2.2.4. **Scrum**

Scrum, which is a subset of Agile, is a lightweight process framework for Agile development and has become one of the most used Agile methods [12].

To get necessary knowledge about Scrum, regarding this project, are some important keywords listed below.

#### *2.2.4.1. Sprint*

The foundation of Scrum is called a Sprint. Sprint is a limited iteration cycle of one to four weeks and where the development team is focused on completing defined objectives before the current sprint ends. Each sprint ends with Review Meeting, or Sprint Review. Every iteration in the sprint aims to increase the product value.

#### *2.2.4.2. Sprint Review*

Is a final meeting with the customer, or the customer representatives and others involved in the development of the product in the current sprint. At this meeting demonstrates and discusses any progress or setbacks that might occurred during the sprint workflow. The meeting then ends with the next sprint planning.

#### *2.2.4.3. Product Backlog*

Is a to-do-list where the product owner or representatives of the product, dynamically adds any new functionality to the product that he might want to be prioritized in a higher order to the next sprint. The highest prioritized objectives are later transferred to a Sprint Backlog.

#### *2.2.4.4. Sprint Backlog*

Consists of a fixed amount of product backlog objectives, determined by the Scrum team and the product owner, which has been branched into smaller, prioritized tasks.

#### *2.2.4.5. Scrum Team*

Is a self-organized group of three to nine members that has been optimally arranged by the product owner and the Scrum Master and they are also included. The team itself, is responsibly to assemble all the expertise needed, hence they do not have fixed project roles within the firm. Task allocation and information distribution is determined by the team members themselves, but every member of the scrum team shares the responsibility for the sprint´s outcome and delivering a complete, qualitative product.

#### *2.2.4.6. Scrum Master*

Serves as a coach, fixer, a gate-keeper and ensures that the product owner and the development team has the best conditions to succeed and moving towards established goals. It is also the Scrum Masters responsibility to ensure that everyone keeps themselves informed about what is happening, what has happened and what will happen. By having daily, short and informative meetings, the Scrum Master ensures that everyone in the team keeps themselves updated. These little meetings are called Stand Up. It is also the Scrum Masters responsibility to make sure the whole team is in a good shape.

#### *2.2.4.7. Daily Scrum*

Is a short, informative meeting, located at the same time and place every day, held by the Scrum Team. Every member in the Scrum Team should be able to answer the following questions:

- What have you accomplished since the last meeting?
- What will you accomplish until next meeting?
- Are there any impediments in your way?

By answering these fundamental questions, the team are able to discuss and bring input on how to solve any complications that might occurred during the development. The purpose with Daily Scrum is to synchronize the team progress and redirect jobs to the developer who is more suited,
then you always strive for product increment [13].

To get an overview of how the Agile methodology Scrum works, see figure1.

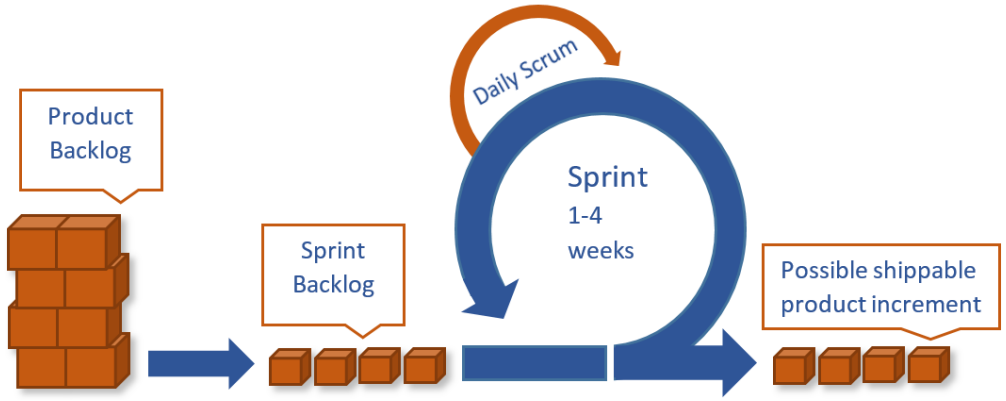



*Figure 1. Flowchart of the Agile methodology, Scrum.*

## 2.3. JavaScript

JavaScript is a dynamic script language (Figure 2), designed to interact with web pages and its data management. JavaScript is also designed to give the user a more fluent experience on web pages, is often found on the client side and is executed by browsers integrated JavaScript engine. Example of when JavaScript is used on web pages:

```
4
5   function testFunction() {
6
7       console.log("Hello World!");
8
9   }
```

*Figure 2. Example of JavaScript programming in Microsoft Visual Studio Code.*

- Automatically format date on web page.
- Route between web pages.
- Change graphic on item when pressed or mouse rollover
- Real time validation on text field and others.
- Capable of performing more advanced functions, such as games in browser.

JavaScript does not only exist in web browsers, but on server side as well, called Node.js, which allows developers to work with connections to databases [14].

### 2.3.1. **Angular.js**

Angular.js is a client-side, JavaScript MVC (Model-View-Controller) framework for adding interactivity to HTML [15]. Angular.js helps you create responsive and dynamical websites, but also helps you write organized JavaScript code which is easy to test. Figure 3 illustrates the principle of Angularj.s. Instead of loading entire webpage again, web server only response with some JSON data, resulting in a web page that feels responsive.

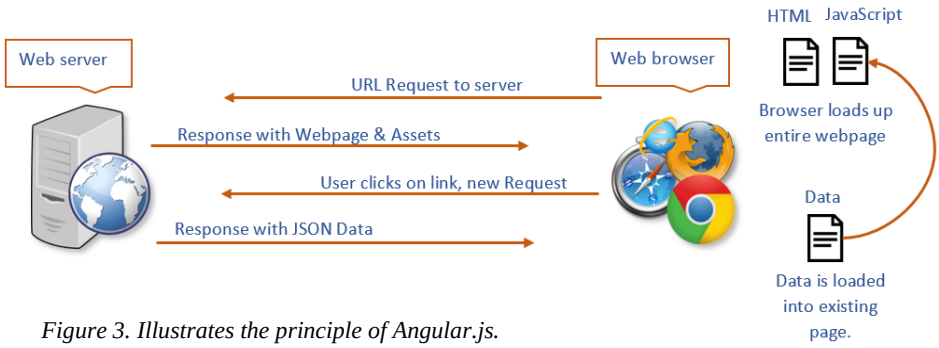



*Figure 3. Illustrates the principle of Angular.js.*

Figure 4, illustrates the principles of MVH architecture and below is a short explanation.

#### *2.3.1.1. Controller*

processes and responds to user interactions, such as button click or typing in a text field.

#### *2.3.1.2. View*

Is the graphical presentation of your processed data, like the graphical presentation on a web page or other medium.

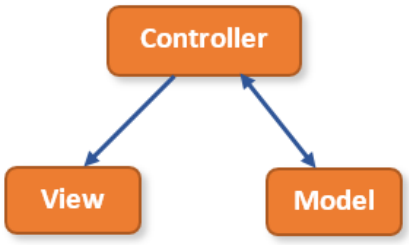



*Figure 4. MVC Architecture.*

#### *2.3.1.3. Model*

Encapsulates the underlying data and business logic. The model is basically plain JavaScript objects.

#### 2.3.1.4. *Two-way data binding*

One key feature to Angular.js is “two-way data binding”, which keeps the model and the view in sync at all times, meaning a change to the model is equal to a change in the view and vice versa. The result of two-way data binding, is that the user gets an application that feels responsive and quickly responds to user interactions.

Figure 5, illustrates two-way data binding in practice.

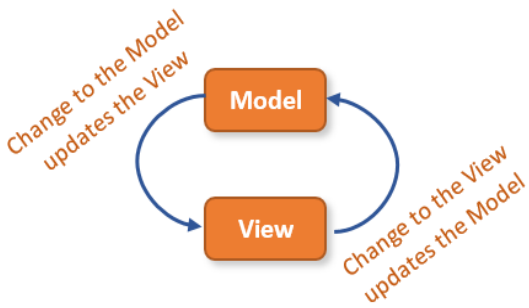



*Figure 5. Two-way data binding.*

#### 2.3.1.5. *Directive*

Directive is one of the most confusing, yet most powerful features of Angular.js, and has become one important part of any application. It adds specific behavior to the HTML DOM-element (via event listeners etc.), which results in you write code that is reusable and testable.

### 2.3.2. **JSLint and JSHint**

JSLint and JSHint is a code quality tool and used in software development for analyzing if JavaScript source code complies with specific syntax. They both works almost the same way as a modern compiler for Java or C# etc. JSLint is used from command prompt and JSHint is used within the JavaScript document.

## 2.4. HTML

HTML is used by browsers and web pages to give structure to its content. The content in HTML is called element, and it can be everything from headers, buttons, paragraphs etc. which later results in a graphical user interface to interact with. HTML is then combined with JavaScript to add functionality to its elements and CSS for styling. The newest version, HTML5, allows implementations like, video playback and drag and drop [16]. Figure 6 illustrates the basics of HTML file.

```
1   <!DOCTYPE html>
2   <html>
3   <head>
4   <title>Page Title</title>
5   </head>
6   <body>
7
8   <h2>My First Heading</h2>
9
10  </body>
11  </html>
```

*Figure 6. Example of HTML programming in Microsoft Visual Studio Code.*

### 2.4.1. **CSS**

CSS stands for Cascading Style Sheet, is an independent CSS-file describing how script language, like HTML and its elements are to be rendered on

different kinds of media, such as papers or screens. Since CSS-file is independent, you are able to use the same file on several documents and it has proven to be very efficient, from maintenance point of view. CSS, HTML and JavaScript forms the basis of today's mobile applications and webpages. Figure 6 illustrates a simple example of how to change color on body and header elements in CSS [17].

```
body {
    background-color: #d0e4fe;
}
h1 {
    color: orange;
    text-align: center;
}
```

*Figure 7. Example of CSS programming in Microsoft Visual Studio Code.*

### 2.4.2. Ionic

Ionic strive to simplify the hybrid mobile application development for devices of today and the future. Thanks to their own powerful framework and by using web technologies like HTML5, CSS and JavaScript, they have managed to create a framework that brings native graphical appearance and feel, and UI interaction of your mobile application to the table [18].

Ionic comes with many basic functions of your mobile application, such as buttons, lists, icons and much more. Ionics standard graphics look similar to iOS7, but because the styling is built in CSS, there is no problem to customize the design at will (Figure 8).

Ionic has made its source code open source for third-party developers, especially web developers, to have the opportunity to simplify and develop new features for future development.


*Figure 8. Illustrates modified Ionic components.*

Currently, Ionic is applying Angular.js to obtain the core functionality of the framework, but you´re able to use ionic without it, hence you are missing out some powerful UI interaction, animations, gestures and alternative concepts [19].

## 2.5. Cordova

The framework makes your hybrid mobile application get the functionality available in today's smartphones. By taking your hybrid mobile application and wrap it in a container, it will act like a bridge between your application and your device-specific functionality, such as Camera, Accelerometer, GPS, Calendar, to mention a few. The result is your web application behaves and feels like an ordinary native mobile application, even though you are running in a web view (Figure 9). Unfortunately, in the current situation has Ionic and Cordova not the potential to reach the same standard as a native application. [20].


*Figure 9. Ionic, Angular.js and Cordova cooperate to receive the device's full potential.*

# 3. Research methodology & Implementation

Since the report covers two different areas, the research methodology will contain two parts. The first part describes a period of internship development of hybrid mobile application takes part in a consulting firm (Softhouse) and which methods is implemented to achieve firm's and customer's goals.

The second part describes which methods are used to develop and then comparing different elements of native and mobile application.

## 3.1. Development environment and software

In order to develop the product for APEA-Hotworks, needed a development environment that handles both HTML, CSS and JavaScript. Microsoft Visual Studio Code offers a complete and easy to use text editor that also supports JSHint and JSLint, which makes it a great development tool for hybrid mobile applications. In order to continuously test their improvements in the application development, mainly used Google Chrome and its Developer Tool (Appendix 1), but there is also the opportunity to test the application in a simulated environment on your own device.

## 3.2. Softhouse

The first step into realize APEA-Hotworks vision, is to design a mobile hybrid application that will span multiple platforms, mainly IOS and Android. By using technologies and methodologies that is currently up to date, in an upward and modern consulting firm in software development.

Softhouse has been commissioned from the customer, APEA-Hotworks, to develop a hybrid mobile application, in order to rationalize and simplify how hot work is carried out in the industry. The first step in realizing their vision began March 16, 2016, when the customer featured product for Softhouse during the introduction meeting.
This meeting was to include the basics from the customer's requirements, which later came to design how the Agile methodology Scrum, came to be carried out during the development process. Together with customer determines who should act scrum master, sprint backlog is discussed and prioritized, and the length of the sprint is stated.
The customer provided a graphical template for further development and ambition to continue with the current software. This template became an illustration of how their first edition might look like, but was developmentally also a good opportunity for further development.

There were given no further instructions on other implementations or specific wishes of software than the current implemented software. which

means that other implementations to the product that requires other software of any kind were implemented without customer approval.

### 3.2.1. Approach

Initial weeks came to be a learning process of how Scrum works in practice, how the implemented software is related to each other, but also to build a foundation in how web programming works, since it is a crucial part of the project. During these first weeks, was planned along with the scrum master and the customer, a sprint backlog that was tailored to the level of knowledge that was available. The sprint backlog was later to become more progressive, when the workflow became more efficient over time. If the customer against all odds was not satisfied with how the product evolved over the last sprint, there is the possibility to customize and prioritize tasks to the next sprint from the customer's new requirements, since the Agile movement welcomes changes.

The product development progressed on a weekly basis according to the scrum methodology and where try and error was an everyday routine. Since I was stationed at Softhouse offices in Växjö, there were personnel expertise directly available in my scrum team, which meant the ability to discuss any problems or solutions never was distant. This resulted in the development of the product could gradually proceed, accordance with the continuous delivery. Much of the time was also spent to find information and tutorials on the internet about the various problems that arose during the development process.

## 3.3. Survey

The investigation part will mainly illustrate the comparison of different elements in hybrid and native mobile applications for Android. Since some of the parts, about developing hybrid mobile application is mentioned above, it will not be mentioned in this section.

## 3.4. Development environment

This section describes the different methods and technologies for developing a native mobile application.

### 3.4.1. Native development environment

Regarding this project, the survey will be conducted with Android Studio, which has been developed by Google since 2013, so it is relatively new to Android development market. Android studio comes with a well-designed built-in IDE (Integrated Development Environment) based on intellij IDEA, well documented website, but even an emulator, which thanks to the latest update (Android Studio 2.1.1, May 18th, 2016) has become much simpler to

use than its predecessor. Android Studio offers a graphical environment for developing GUI, which makes it quite convenient to use. By unlocking development option on your Android device, it is also possible to directly download your application on the device for further testing.

### 3.4.2. Native development programming language

For developing native mobile applications, Android Studio will be used with Java, although it is possible to program applications with other programming languages in Android Studio.

### 3.4.3. Hybrid development

The same techniques, methodologies and environments that were applied in Softhouse Växjö to develop hybrid mobile applications, will also permeate the following tests in order to fulfill the survey.

## 3.5. Survey methodology

The report´s comparative study intends to compare several different elements on native and hybrid applications, each representing methodology and techniques behind creation of different elements in standard version (No third-party software developers).
The report will later recognize them biggest differences between those different approaches and from that draw a conclusion that reveal which option might hold for future development. The report will only focus on comparing elements, based on their own conclusions, which seems to be used most frequently in today's mobile applications, such as buttons, lists, keyboards etc.

## 3.6. Test device

All tests will be using Samsung Galaxy S6, Android version 6.0.1, to conduct the tests on native and hybrid mobile applications.

# 4. Results and analysis

This section will show the results of these different approaches for how to make various elements in them both application types, native and hybrid. The degree of difficulty will gradually increase, start with the simplest and finish with the more complex element to be implemented.

## 4.1. Graphical Implementations Differences

Since there are several completely different ways to manufacture mobile applications, it is not entirely surprising that there are several implementation ways as well. Both cases have proven to be easy to grasp. Android Studio with its graphical GUI tool, which allows drag and drop but even implement components in xml version. Ionic offers their own Ionic Creator, which also has drag and drop functionality and HTML version. Ass figures 11-12 illustrates, just by implementing a button in the native application requires some lines of code, while the hybrid only have a few. In this example, the buttons do the same thing, calling a function that generates a random number.

```
9       <button style="float: bottom;" class="button button-block button-stable;"
10              ng-click="getRandom()">
11              Push Button
12      </button>
```

*Figure 10. Button from Ionic, implemented with HTML.*

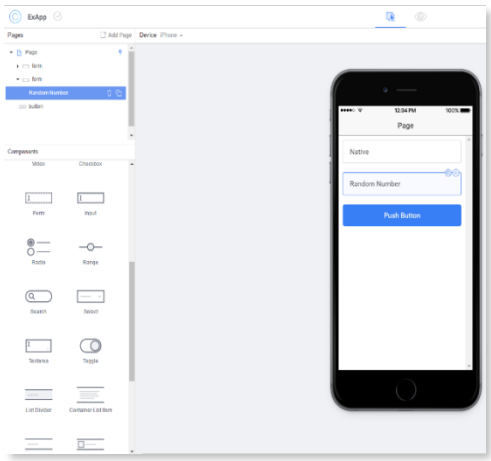


*Figure 11. Ionic Creator offers drag and drop functionality for creating hybrid applications.*

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Push button"
    android:id="@+id/randomButton"
    android:clickable="false"
    android:onClick="buttonOnClick"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
```

*Figure 12. Example of button implementation with xml in Android studio.*

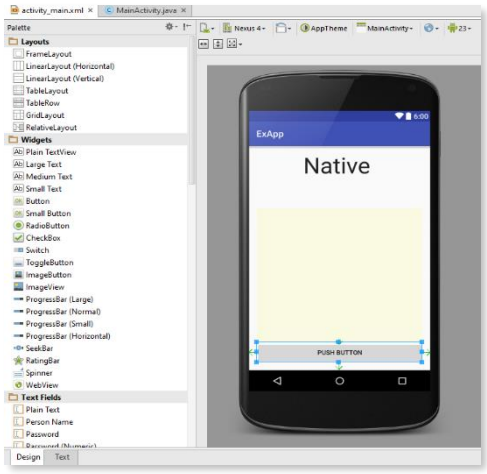



*Figure 13. Graphical development environment in Android Studio.*

Both ways have their advantages and disadvantages. One advantage with hybrid mobile development is that you do not need to download the application to your phone to test your changes, as you can use Google Chrome Development tool (Appendix 1), although the possibility exists. While Android Studio either have to open up the emulator, or download the application to your phone. Both ways with Android Studio has proven to be time consuming and the emulator requires some resources from your computer. The advantage with Android Studio, compared to the hybrid mobile application development with Ionic Creator, is that you are constantly updated on Android's latest feature and components.

Both development options offer unlimited possibilities for styling of components. Thanks to third-party developers, there are also components and software that advances your application even more, both to native and hybrid.

## 4.2. Code implementations differences

Since the applications are created with different techniques, depending on the implementation of the hybrid application, you can according to old -style web development add logic and elements in a single file, which also operates today, but in large programs can become confusing and even to complex. With Ionic and Angular.js, you are more or less forced to separate the elements with the logic, then you usually are using the templates where the code is already structured. This might be difficult to grasp in the beginning, but when building large projects, it will definitely help you develop structured code.  Android Studio offers empty templates as well and thanks to the power of Object Oriented programming, you can structure your code as you please. But compared to the other technique, you are not forced to it, which might result in complex code when your application grows in size.

## 4.3. Test 1: buttons

Buttons is one of the key elements in applications and comes in various shapes and sizes. When interacted with, user expects a particular action will transpire. Both types come with a wide variety of buttons, such as radio buttons, checkboxes, sliders etc.

### 4.3.1. **Graphical differences**

Figure 14 and Figure 15, illustrates a button implemented in both hybrid and native mobile applications. The graphical user interface differs very little,

and with a little more CSS adjustment, you can certainly get them to look the same. Android button´s has something called long click, which also can be implemented in Ionic button´s.

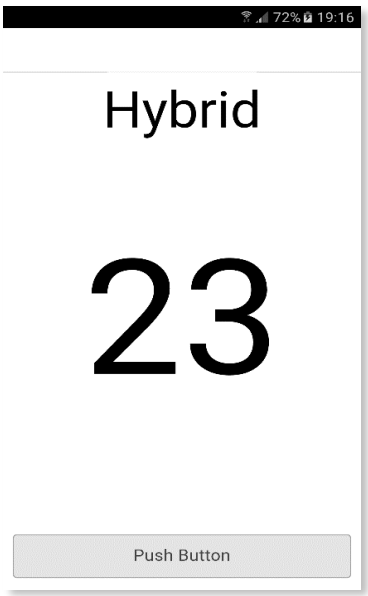



*Figure 15. Graphical illustration of button implemented in hybrid application*

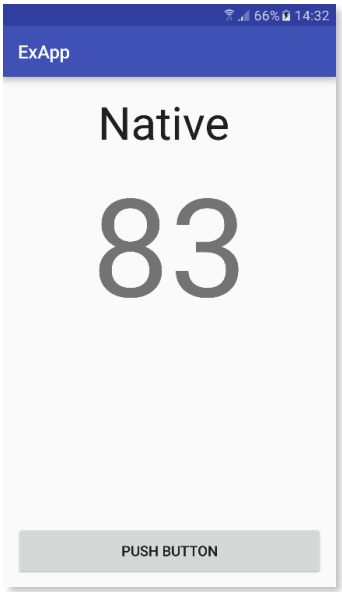



*Figure 14. Graphical illustration of button implemented in native application*

## 4.4. Test 2: List

Lists allows users on a representative and easy way present different types of data in a scrollable view. The ability to dynamically add and remove data are now a thing we take for granted in various applications. In both hybrid and native exists the ability to add lists in popups and others.

### 4.4.1. **Graphical differences**

Graphically, it's nothing that distinguishes them significantly. Should I go into details, have the hybrid application what looks to be a list constructed with labels, making the list do not go all the way to the edges. Worth mentioning is that no styling or other modification has been made on the list views, but this is the default of the respective frameworks. Both approaches were found to not have any problems with rendering the list when it was filled with over 40 items, but both delivered a responsive and smooth experience. In the current situation provides Android more modification to their lists, when you have the ability to do much more modifications to how each row should look like.

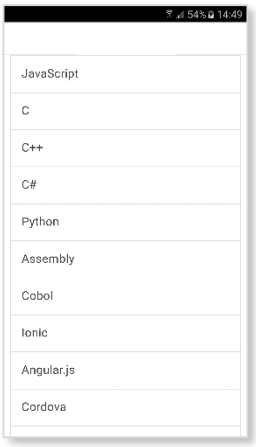



*Figure 17. List view in hybrid application.*

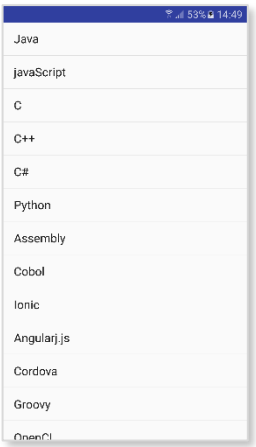



*Figure 16. List view in native application*

### 4.4.2. Implementations

Both methods have their own characteristics on how to implement a list and to populate data. Both types retrieve data from a source, which may be a complete data source, possibly an array or other container, or add to the list dynamically.

Android uses the adapter, which is the link between the source and your List View, to populate your list. While Ionic uses a Angular.js directive called ng -repeat. Technically, the principles are the same and both methods are simple to use and understand.

## 4.5. Test 3: Text field & keyboard

Text fields allows the user to enter text into your application. The text itself may not only consist of letters, but also numbers, date, email, password and other characters and can be completed in single line or multi-line. Every time the user presses a text field, the cursor moves and the internal keyboard will appear. It is also possible to limit the keyboard to mere numbers, letters, or modify content or styling according to their own needs.

### 4.5.1. Graphical differences

Both the hybrid and the native mobile application uses the inbuilt keyboard and its features, like cut, copy, share etc. They main differences is the hybrid applications lack of two buttons and the letters hinted in almost each number. The text field in Figure 19, have been styled in CSS to look like the native text field in Figure 18. Current figures are limited to numerical values.

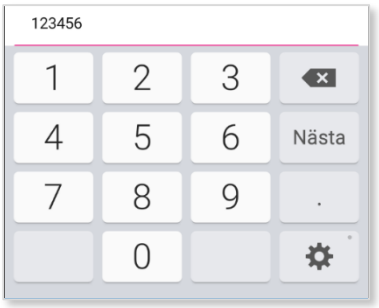



*Figure 19. Default numeric keyboard on hybrid application.*

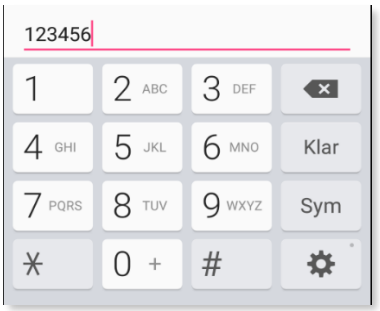



*Figure 18. Default numeric keyboard on native application.*

The similarities between the two keyboards becomes more similar to each other when you change to plain text input, as illustrated in figure 20-21. If you are developing hybrid applications with Ionic and Cordova, there is no reason why you should not use the Ionics keyboard plugin to really make use of your device's functionality and modification opportunities.

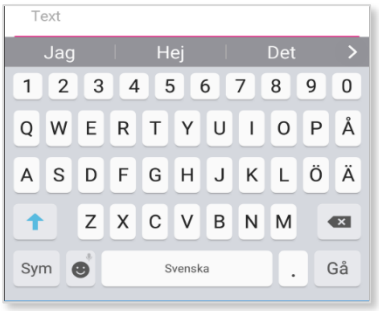



*Figure 21. Default text keyboard on hybrid application*

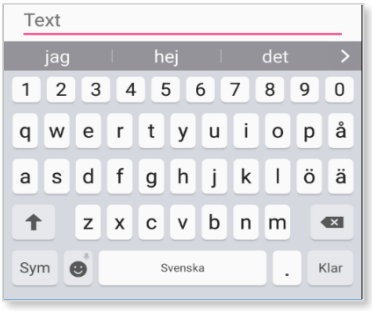



*Figure 20. Default text keyboard on native application*

In addition to the above, there is nothing more that separates them visually.

### 4.5.2. Features

Native applications and especially Android Studio offers much more modifications to their text field modifications, than Ionic do in the current situation. An important feature that currently is difficult to implement in Ionic is AutoComplete, which more or less require some form of third-party developed software, while it is available in the Android Studio. When it comes to further modifications on the text field, unfortunately, Ionic has not quite managed to implement all the features that are available in native applications. But then, web development is on the rise, it will not be long before Ionic, more or less, has the same features as native.

As for the keyboard, you can only restrict the type of input. To modify as you like, requires a complete framework from a third-party developer or

Ionic Keyboard Plugin, compared to Android Studio where you can change the layout without additional frameworks.

### 4.5.3. Text field & Two-way data binding

The hybrid mobile application development has a great preamp, if you use Angular.js as an easy and convenient way to get real-time validation of your input. This is thanks to a Angularj.js directive ng -model and two-way data binding. Android currently has two-way data binding, but is in beta stage and is complex to implement into your application [21]. Real-time validation means in this case that input is validated while the user types, and not to validate input after the user performing another event.

## 4.6. Result Softhouse

My work on Softhouse Växjö resulted in a first stage version of a hybrid mobile application that came to be part of APEA-Hotworks product. The result is an application that feels responsive and looks and feels like a native application, but maybe the most important thing, user-friendly. Unfortunately, much time went to learn various programming languages, technologies and methods. Figures 22-27 illustrates the result.

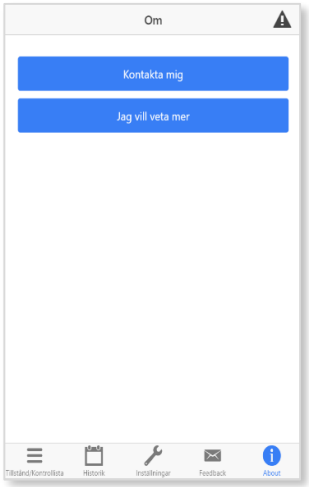



*Figure 24. About-tab.*

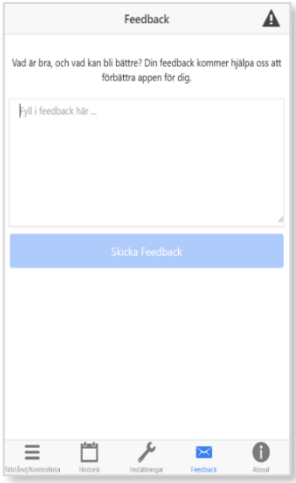



*Figure 24. Feedback-tab, send feedback about the product.*

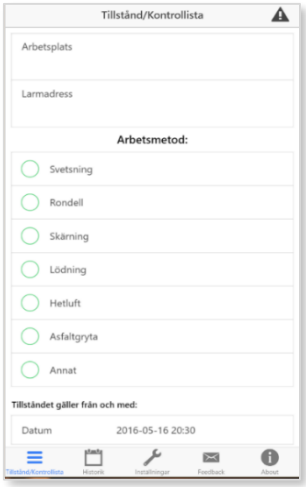



*Figure 24. Main view in the application.*

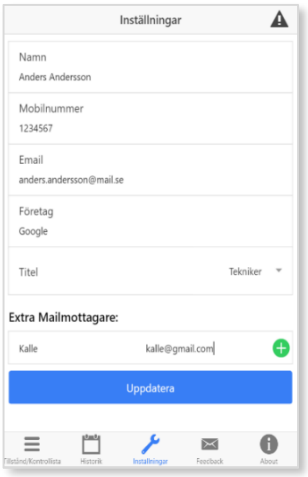



*Figure 27. Settings-tab.*


*Figure 27. Show modal when report being send.*

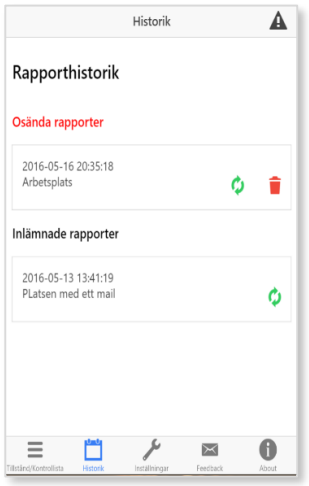



*Figure 27. Send and unsent reports shows in history-tab.*

## 4.7. Further development

When there is demand for a mobile application that more or less requires the device's basic functionality such as camera, GPS, Accelerometer etc., but also more advanced device-specific features which may require much performance, then native is right way to go. Native applications are currently more modifiable and you get the full performance of the device. The downside to native is that it is expensive to develop, as a native application usually occurs in several versions and requires specialized platform developers. For a company that has a different platform -based applications, this can prove to be expensive, since it requires multiple development teams. Another drawback of the native is that you also have to upload every single update of the application to any app marketplaces.

When the need for an application does not require the entire device´s full potential, functionality and performance, then the hybrid application is the right choice. With a hybrid application, you do not really get the performance of a native or its functionality, but you get an application that will pay off in the long run and almost feels like a native application. Not only do you not need to upload every update to the app marketplace when it is done automatically in the Web view of your application, without the development costs of a hybrid application is lower than the native. The reason the cost is lower, it only requires a development team to reach multiple platforms.

Both applications tend to be more complex the more advanced functions and bigger it becomes, but both can be built with the structured code because of their different construction techniques. Android and Java, which is an Object

Oriented programming language and Ionic, thanks to its structure with Angular.js offers easy structured code. in the current situation has native a small advantage, but it is only a matter of time before hybrid application development is beyond the native application development.

# 6. Discussion and conclusion

The report's actual results, which are the result of my work at Softhouse Växjö. I think it reflects all those many hours that have been spent on trying to learn them different methods and techniques used for developing a hybrid mobile application. When it comes to the survey, I would like to go deeper into both the development of both native and hybrid manufacturing, but when the time was not sufficient and loss of work, I think, despite what happened the result is acceptable. I am, after all that has occurred, pleased about my result. What I have done is to prove to others and not just myself that I can learn a new programming language, and even get a result that I think is good, considering how short the process still has been. To get an even better result, I might have been able to create an application on a programming language that for me to be known before, or get a few weeks to learn a new programming language.

When web development made great progress in the introduction of HTML5, it reduced the distance between the native and hybrid application development. Once Ionic and Angular.js was introduced, hybrid caught almost up, but it will, unfortunately, not quite live up to their full potential available in native. If they ever do it? I believe that native mobile application belongs to a larger company with the financial resources. As the costs of having specialized development team within a given platform can be expensive in the long run. I think hybrid is an option that will, more or less, replace the native, thanks to it cross platforms availability. But before that, the next revolutionary thing in web development must be introduced before the hybrid mobile application development even catch up, or even bypass.

# 7. References


[1]. D. Aamoth. (Aug 18th, 2014). *First Smartphone Turn 20: Fun Facts About Simon* [Online]. Tech Gadgets, http://time.com/3137005/first-smartphone-ibm-simon/(May 9th, 2016).

[2]. AppSchopper blog team, no name. (Dec 6th, 2012). *History – Mystery of Mobile Application Development Revealed Here.* [Online]. Appschopper, http://www.appschopper.com/blog/history-mystery-of-mobile-application-development-revealed-here/(May 9th, 2016).

[3]. P. Cohen. (Jan 9th ,2007). *Macworld Expo Keynote Live Update.* [Online]. Macworld, http://www.macworld.com/article/1054764/liveupdate.html (May 9th, 2016).

[4]. P. Ahrnstedt. (July 14th, 2008). *iPhone App Store Downloads Top 10 Million in First Weekend*. [Online]. Apple Press Info, http://www.apple.com/se/pr/library/2008/07/14iPhone-App-Store-Downloads-Top-10-Million-in-First-Weekend.html (May 9th, 2016).

[5]. A. Dobie, R. Holly, J. Hildenbrand. (Oct 21th, 2015). *ANDROID´S EARLY DAYS, The T-Mobile G1 arrives.* [Online]. Androidcentral, http://www.androidcentral.com/androids-early-days (May 9th, 2016).

[6]. D. Reagle. (Nov 30th, 2012). *Mobile Operating Systems and App Marketplaces.* [Online]. Sitepoint, *http://www.sitepoint.com/the-advancements-in-mobile-design-and-how-it-has-developed-into-a-strong-industry/ (May 10th, 2016).*

[7]. R. Budiu. (Sep 14th, 2013). *Mobile: Native Apps, Web Apps, and Hybrid Apps.* [Online]. Nielsen Norman Group, https://www.nngroup.com/articles/mobile-native-apps/ (*May 10th, 2016*).

[8]. P. Varhol. (Aug 26th, 2015). *To agility and beyond: The history-and legacy-of agile development.* [Online]. TechBeacon, http://techbeacon.com/agility-beyond-history%E2%80%94-legacy%E2%80%94-agile-development (*May 10th, 2016*).

[9]. M. Rouse. (Sep 9th, 2011). *Agile Manifesto.* [Online]. TechTarget, http://searchcio.techtarget.com/definition/Agile-Manifesto (May 11th, 2016).

[10]. D. Bishop, A. Deokar, "*Toward an Understanding of Preference for Agile Software Development Methods from a Personality Theory Perspective",* 47th Hawaii International Conference on System Science, Hawaii, 2014, pp.2. (May 11th, 2016).

[11]. M. Rouse. (Feb 15th, 2007). *Agile software development (ASD).* [Online]. TechTarget, http://searchsoftwarequality.techtarget.com/definition/agile-software-development (May 11th, 2016).

[12]. K. Thompson. (2010). *What Is Agile? What Is Scrum?* [Online]. cPrime, *https://www.cprime.com/resources/what-is-agile-what-is-scrum/* (May 12th, 2016).

[13]. Softhouse Agile Coach Team, no name. (Feb 6th, 2014). *SCRUM in five minutes* [Brochure]. Softhouse, https://issuu.com/softhouse/docs/scrum_5min_eng_131210 (May 13th, 2016).

[14]. M. Rouse. (Sep 21, 2005). *JavaScript.* [Online]. TechTarget, http://searchsoa.techtarget.com/definition/JavaScript (May 13th, 2016).

[15]. D. Lau. (Sep 5th, 2013). *10 Reasons Why You Should Use Angular JS* [Online]. Sitepoint, *http://www.sitepoint.com/10-reasons-use-Angular.js/ (May 16th, 2016).*

[16]. M. Rouse. (Jul, 2010). *HTML File Format.* [Online]. WhatIs, *http://whatis.techtarget.com/fileformat/HTML-A-Web-page (*May 14th, 2016*).*

[17]. J. Kyrnin. (Dec 10th, 2014). *What Is CSS?* [Online]. About tech, http://webdesign.about.com/od/beginningcss/a/aa021607.htm (*May 16th, 2016*)

[18]. A. Bradley. (Oct 28th, 2013). *Where does the ionic framework fit in?* [Online], http://blog.ionic.io/where-does-the-ionic-framework-fit-in/ (May 15th, 2016).

[19]. Ionic, (May 12th, 2016). Chapter 1, *All About Ionic*, [Guide], http://ionicframework.com/docs/guide/preface.html (May 17th, 2016).

[20]. P. Virinchy, (Jul 27th, 2014). *What is Cordova and how does it work?* [Online], http://scn.sap.com/community/developer-center/mobility-platform/blog/2014/07/27/what-is-cordova-and-how-does-it-work (May 22t, 2016).

[21]. F. Collini, (Jul 22, 2015). *Two-way Android Data Binding – How to use two-way Data Binding to manage a layout.* [Online], https://medium.com/@fabioCollini/android-data-binding-f9f9d3afc761#.cwu566atn (May 24th, 2016).

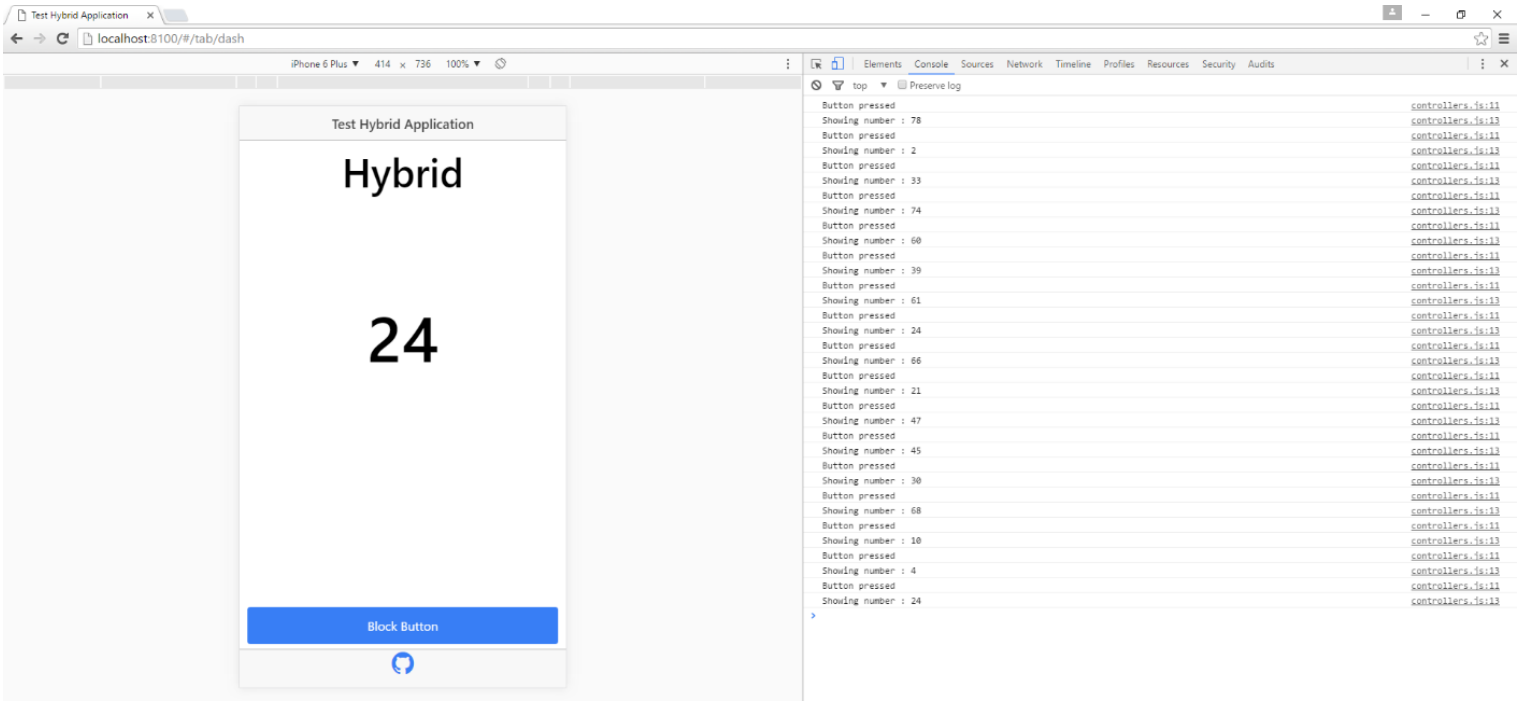
# 5. Appendices

APPENDIX 1 Development environment Google Chrome

# APPENDIX 1 Development environment Google Chrome

Illustrates the development environment in Google Chrome.

When running your application, Ionic will start a local server on your computer and build a local emulator on your default web browser which you can interact with.



Lnu.se

Linnéuniversitetet
Kalmar Växjö

Fakulteten för teknik
391 82 Kalmar | 351 95 Växjö
Tel 0772-28 80 00
teknik@lnu.se
Lnu.se/fakulteten-for-teknik