Usability and User Experience in Mobile App Frameworks
Subjective, but not Objective, Differences between a Hybrid and a Native Mobile Application

Lina Andersson
Abstract

Usability and User Experience in Mobile App Frameworks

Lina Andersson

The native framework has been seen, in both research and industry settings, as the best framework for mobile development in many aspects, if it is compared to the hybrid framework. Often the technical differences are of interest and the usability and user experience aspects have not been fully covered.

In this paper usability is tested by looking at quantitative data from user tests and the user experience is tested by using qualitative data from interviews. The results from 20 users tests show that the usability is the same between the frameworks, while the user experience points towards a favor for the native framework due to its more familiar look and feel.

Thus, there is a subjective, but not objective, difference between the frameworks.
Contents

1 Introduction 4
   1.1 Purpose 4
   1.2 Limitations 5

2 Background 5
   2.1 Caspeco 5
   2.2 Technology 5
      2.2.1 Native Applications 5
      2.2.2 React Native 7
      2.2.3 Hybrid Applications 7
      2.2.4 React.js 7
      2.2.5 Apache Cordova 8
   2.3 Usability 8
      2.3.1 Effectiveness 9
      2.3.2 Efficiency 9
      2.3.3 Satisfaction 9
      2.3.4 Limitations 9
   2.4 Critique Against Small Screens 9
   2.5 User Experience 10

3 Theory 11
   3.1 Online Articles 11
   3.2 Current Industry Knowledge 11
   3.3 Research Articles 12
   3.4 Hypothesis and Null-hypothesis 14

4 Methods 15
   4.1 Prestudy Interviews 15
   4.2 Development of Test Applications 15
      4.2.1 Design Phase 15
      4.2.2 Implementation Phase 15
   4.3 Measures 16
      4.3.1 Quantitative Research Design 17
      4.3.2 Qualitative Research Design 17
      4.3.3 Mixed Method Research Design 17
   4.4 Usability measurements 18
   4.5 Interviews 19
   4.6 Test session 19
      4.6.1 Tasks 19
      4.6.2 User Groups 21
      4.6.3 Test Site and Equipment 21
      4.6.4 Procedure 21

5 Prestudy tests 22

6 Data collection 22

7 Data analysis 24
   7.1 Analysis of Quantitative Data 24
      7.1.1 Effectiveness 24
      7.1.2 Efficiency 26
      7.1.3 Satisfaction 28
   7.2 Analysis of Qualitative Data 28
   7.3 Results 29
      7.3.1 Demographics 29
      7.3.2 Usability 29
      7.3.3 User Experience 32

8 Discussion 35
   8.1 Usability 35
   8.2 User Experience 36
   8.3 Contributing Factors 37

9 Conclusion 38

10 Future Work 39

References 39
1 Introduction

The mobile phone has become part of everyday life for a lot of people all around the world. In 2016, around 4.6 billion people owned a mobile phone, which is about 62.9% of the world population. However, the popularity of the device has not reached its peak, at least according to Statistas forecast which expects the number of mobile phone users to increase to about 67% by 2019 [1].

More than 50% of all mobile phone users are expected to be smartphone users by 2018 [1] and with so many people using smartphones there are also a lot of companies and independent developers who want to build applications for those users [2] [3].

However, when starting a project with the goal to develop a mobile application it is a lot that needs to be decided. Everything from who the potential users are, to what platforms the application is going to be used on and what colors to use in the design are of interest. One of the decisions that needs to be made is what kind of technical framework is going to be used for the development of the application and the user interface. This decision might not be easy since each framework has its advantages and disadvantages.

When developing an application today there are three types that are most commonly used; native, web and hybrid applications. Both native and hybrid applications are possible to download from app stores like Google play or App store, which is not possible with a web application that is entered through URL searches. Web and hybrid applications are similar in the way that they both are rendered in a webview, which is not the case with a native application [4]. One advantage with web applications is that there is only one application that needs to be developed, and then it can be used on several platforms. This contrasts with the native applications that needs to be developed separately for each platform [5]. The hybrid application is just like the web application and one application can thus work on all different platforms [4].

Besides the differences that are mentioned above, the usability and the user experience of the application itself might also be affected by the choice of framework. This is for instance mentioned in articles online [6] [7] [8] [9], but also mentioned in scientific research articles [10] [5]. For a company the usability and the user experience of the application should be of high importance, since a bad usability or user experience might lead to the users migrating to other services [11].

Some popular applications that have a lot of users are Instagram, Facebook, Skype and Airbnb, which all are native applications built with a native framework called React Native [12]. These native applications all have over 300 million users, but there are also applications that do not have just as many users that is out on the market as well [13]. Some examples of this is for instance McDonald’s in Turkey, but also Untappd and Diesel who all have made hybrid applications [14]. But what is it that makes the native applications attract more users than the hybrid applications? Are they really better in terms of usability and user experiences or is there a significant difference at all? These are the overall research questions investigated in this paper.

1.1 Purpose

This study aims at gaining a better understanding of the differences in usability and user experience between different frameworks. The current knowledge points towards a better usability and user experience in native applications compared to hybrid applications, but what actually makes the difference between these two frameworks is currently not explored in depth in terms of usability and user experience. For the mobile development industry it is important to know what actually makes the difference, if there is any, and thus the purpose of this study is to look further into that area of interest.
1.2 Limitations

The focus is on the user interface and the user’s experience of it. Thus, this study did not look at other parts of the system that might have been affected by the choice of framework. This means for instance that questions like how fast a page is rendered were not studied, but instead the study focused on how fast the users completed their task.

In the study, the tests were only done on mobile smart phones, which means that devices with larger screens such as tablets or desktop screens were not included.

The study focused on the differences in usability and user experience between the different frameworks, which means that the specific application used to do this test was not evaluated on its usability since that was not the aim of this study.

Another limitation is that a pure web application was not studied and used to compare with the native and hybrid applications. A study of that kind would bring more data and information about the different frameworks in regards to the usability and user experience, but due to several reasons this was not done. One of the reason for this decision was that the focus in this study was purely on the applications that users download from specific app stores and thus can experience as the same kind of application even though they are built differently. Another reason was that the development time for both the native and web frameworks would extend beyond the time limits of this project.

Both hybrid and native applications can be studied on various platforms, however this study did only look at the differences in usability and user experience between hybrid and native applications on iOS, thus excluding for instance Android and Windows phones.

2 Background

2.1 Caspeco

Caspeco is a company that offers business management solutions for hotel and restaurant businesses. The company has its office in Uppsala with around 45 employees [15]. Currently they have a hybrid application built with React.js, but they are considering to replace it with a native application made with React Native. Due to that possible change questions have been raised about how much the company would benefit from changing from a hybrid to a native application in terms of usability and user experience.

2.2 Technology

The Nielsen Norman group is a company that focuses on user experience, both in research, consulting and training within that field and it does so in a evidence based way [16]. Raluca Budi works at the company as Director of research and also has a degree as a doctor of philosophy in computer science from Carnegie Mellon University [17]. More than 20 articles and conference presentations within the field of human-computer interaction have been written by her, but besides that she also has written a lot of articles concerning for instance usability, user experience design and user interface principles for the Nielsen Norman group. The usability and user experience of mobile devices are also something that she has written about [16] and the definitions she uses of what a native application and hybrid application is are used in this report and is also explained more in detail below. An overview of the applications are also possible to see in Figure 1.

2.2.1 Native Applications

A native application is developed specifically for the platform that it is used on [4]. This means that for instance Android and Apple have different frameworks for applications that are used on their particular plat-
Andersson, p. 6

Figure 1: Overview of how the different frameworks works

The native applications are all downloaded through an app store application such as Google play on Android products or App Store on Apple’s products and can then be accessed through an icon on the home screen. Since the native applications are made specifically for one platform it is also possible to access features from that platform and use them in the application, this includes for instance the camera, the contacts stored on the phone and the GPS. The native application can also make notifications pop up on a user’s mobile and the application can run when a connection to the Internet is not available [4].

Regarding the user interface the native applications take advantage of the systems built in features and can thus create a familiar look and feel for the users [5]. The frameworks often has Human-Computer Interaction (HCI) guidelines that the developing teams should follow when making an application for a specific platform, which contributes to a consistency among the applications on that particular platform [10].

A clearer definition of what look and feel actually is might be needed here, and even though it is not defined by law, it is however argued that "look" refers to an application’s visual appearance while "feel" on the other hand refers to the applications interactions with the user and how it performs in those moments [18].

As mentioned earlier the native applications are developed specifically for each different platform and thus uses different frameworks to do so. For instance is Objective-C or Swift used to make native applications for platforms that has iOS as operative system [19]. In contrast to Apple, Android applications uses Java as their framework while Windows uses C# for their native applications [9].

However, new frameworks have emerged and now there are so called cross-platform frameworks on the market that solve the problem of having to build separate applications for each platform. Two of these cross-platform frameworks are Xamarin and React Native [20] [21], where the later one is going
to be used in this study.

### 2.2.2 React Native

In 2013 React Native was started as a part of a hackathon at Facebook where the goal was to build a framework that would make it possible for them to go from two development teams, one for Android and one for iOS, to one team. They wanted to speed up the development time for the applications built by the company and had an idea about making a cross-platform framework with JavaScript to solve that problem. In 2015 the React Native code was released as an open-source project on Github and had in 2016 about 650 people who contributed and 5,800 commits to the code base. Even though the framework is used and developed by Facebook, around 30% of the people contributing to the project are not hired by the company [22].

The framework is used in thousands of applications according to Facebook and the companies that uses it comes in all different sizes. Some example of the companies that have applications built with the framework is, besides Instagram and Facebook, Skype, Tesla, Walmart and UberEats [21].

When building an application with React Native both JavaScript and React are used, but it is also possible to add code from the native frameworks if needed. This means that if you are building an application for an Android phone it is possible to add Java code into the React Native code if there are certain features that are more easily or only possible to do that way [21].

### 2.2.3 Hybrid Applications

A hybrid application is, as its name suggest, a mix of both a native and a web application. Just as the native application it is possible to get the hybrid application through an app store like Google play or App store. It is possible to use the platforms built in camera, GPS and other device features in a hybrid application, just like the native application. However, there are some differences between the hybrid and the native application. For instance, the hybrid application is rendered in a webview, just as a web application.

There are a number of different advantages and disadvantages with a hybrid application that influences if this kind of framework is going to be used or not. Some chooses to build hybrid applications instead of native application since this can be more effective in terms of time and money spent on the development. One example of this is that if there already exists a website that code can be reused and transformed into a hybrid version. By choosing a hybrid application framework it is also possible to save time and money on the development on one other way as well. The way in which a company can benefit from choosing this type of framework is because a hybrid application can be used on all kind of platforms, and thus there is no need for development and maintenance on several different applications. As a difference to the web application, that is not possible to get through an app store, a hybrid application is possible to download through these kind of stores, thus give a presence on an attractive market [4].

Some companies that have chosen to make hybrid applications are for instance McDonald’s in Turkey, Untappd and Diesel [14].

### 2.2.4 React.js

Facebook has also built React.js which is a library that, just as React Native, is built with JavaScript. The purpose with this library is to make it as easy as possible for developers to make user interfaces that are interactive. Another feature of the library is that it makes no assumptions on what other technology that is used in the same stack [23] and, as it is meant to work for big platforms such as Facebook it is also scalable. It is an open-source project that started in 2013 and challenged how web development worked at that time, something that was taken with both cheers and dissatisfaction by the web development community [24].

The split reactions from the community...
comes from the fact that the library did not only challenge but also made a shift within the web development community; suddenly both web experts and novices could make interactive, large-scale websites. The library solved a problem that was not unique to Facebook as a big technology company, but a problem that was noticed by other companies as well as among hobby programmers: it was hard to build a large-scale web project that could handle data that changes over time and still kept it maintainable. To solve this problem React.js was created and since 2013 the library has grown. In November 2017 the project had 1128 contributors and 9373 commits on Github [24] [23].

2.2.5 Apache Cordova

To be able to transform the code written with React.js to a hybrid application a framework for the transition is needed, which in this case was Apache Cordova. It is an open-source mobile development framework that uses a webview to package the HTML-code into a native container, making it possible to distribute the application on app stores and to access device Application Programming Interfaces (API) [25], see Figure 1.

2.3 Usability

The definition of usability can vary depending on who is using the definition. For instance, Mills et al published an article in 1986 on the topic stating that many companies looked at the number of errors in the code and documentation, but missed the fact that the product might be difficult to use even though those errors were to be fixed. Some companies had started doing usability tests, but what exactly usability was and how it was tested varied. According to the authors it is important to think of for instance who the users are, what to measure, when and what to test but the usability concept was also concerned with the ease in which a user can learn to use the system [26].

Another person who also looked at the usability and its definition is Shackel, who suggested that usability should be determined upon how easy it is to use but also its effectiveness, which means that the user’s performance should also be measured in some sense. According to Shackel the usability of a product should be measured within certain conditions as well, for instance there should be some predetermined users that have trained and received support in using the product and conducts these usability tests within a certain context and with specific goals [27].

Krug says that if a product should have a good usability it means that the product works well and that a person that is even below average in both skills set and previous knowledge can use the product. He also points out something that is usable should not take or appear to take a lot of time since that reduces the change for the product to even be used in the first place. The initial use of the product is also something important within the usability concept according to Krug, who says that the users needs to be able to understand how to use a product and what it should be used for without having to put any kind of thinking into it [28].

There is however one definition that is widely used among researchers [29] and which is also going to be used in this study. It is the usability definition set by the International Organization for Standardization (ISO). The ISO has a goal of developing international standards to make sure that services, products and systems will be developed with a focus on good quality, safety and reliability. The standard is set by experts from the particular field or fields that are affected by the proposed standard and those experts shares knowledge and develops these standards together over time [30].

In the ISO 9241-11 the usability concept is defined as a measure that makes it possible to design and evaluate a visual display terminal, something that is done with the aim of making it possible for the users of the systems to reach a certain goal and to meet specific needs in a
specified context of use. In the standard it is a focus on the performance and the satisfaction of the users that uses the system which is measured in terms of effectiveness, efficiency and satisfaction. In other words the usability concept is focusing on to which extent the intended users reaches one or several goals, how much effort it takes to achieve that goal and how acceptable the product is to the users [31].

What is important to remember is that the context of use is central to the usability definition as well, since a product might for instance be satisfying to use in one context, like at home, but not particularly acceptable or satisfying to use on a train. The context does not have to be just a physical environment but can also be the tasks, the equipment or the social environment [31].

The users are also of concern since a product for one user might have a higher usability than it might have for another group of users [31]. One example of this is a study by Chadwick-Dias, McNulty & Tullis where they compared the usability of a product between different age groups. The study found that a lot of factors contributed to a lower usability among older adults, for instance social, cognitive and physical factors but also the previous experiences with computers [32].

To be able to measure usability three categories of measurements are of interest: effectiveness, efficiency and satisfaction.

2.3.1 Effectiveness

Effectiveness is defined by the ISO as the part of the usability concept that is concerned with the specified goals that the users has when they use a product, more exact it is concerned with to what extent the users can reach a certain goal and how accurate their achievement of that goal was [31].

2.3.2 Efficiency

Efficiency is concerned with the relationship between how much of and how correct a specific goal was meet and how much effort that was put into reaching that particular goal [31]. This means that if there was a lot of effort that the user had to put in, but could not reach the intended the goal at all, the efficiency is low, while the efficiency is high if the user puts little effort in it but completes the goal in an accurate way.

2.3.3 Satisfaction

Satisfaction is a measurements of the user’s attitudes towards the product that is of interest, the positive attitudes are of interest as well as the discomfort that also might be present when a user uses the product [31].

2.3.4 Limitations

Since the focus on this study is going to be on the definition set by IOS on what usability is, other aspects of usability will not be studied more in depth. This means that usability aspects such as ease of use, learnability and initial understanding of the product were not examined in terms of usability in this particular study.

2.4 Critique Against Small Screens

The study by Ryan & Gonsalves also bring up aspects of how the interaction can differ between the use of an application on a big screen, like on the desktop, compared to the smaller screens of a mobile phone, which is important to mention in this context as well. Even though the mobile application frameworks are discussed in this study with its advantages and disadvantages, mobiles still has its own downsides compared to bigger screens. For instance more complex information is difficult to show on a smaller screen without the risk of losing valuable parts of the information [33]. This is for instance mentioned by Chae & Kim who saw that user changed their own perception and how they navigated on sites depending on the size of the screen and how the information was structured. The users
scrolled a lot more on the small screen products due to the fact that all the information was not visible. Another result of the study showed that the horizontal scrolling also affected the users cognitive workload. On a small screen the users more often pressed the button to go back and forward, but on the other hand this lead to a funneling, meaning that less information was shown to the user at each page, giving the user a more focused used of the cognitive resources at hand. One important take on the results of this study is that the complexity of the task was of high importance due to the usability issues raising with the use of smaller screens [33].

The problems with small screens is also mentioned by Albers & Kim who says that the smaller screens cannot fit the same amount of information, which leads to a reduction in the amount of information presented on those screen. This can however lead to a distortion of the information presented and also block the user from seeing relationships in the information, something that would be visible on bigger screens [34].

Thus it is obvious that no medium can by itself solve all problems that we have since they all have their own advantages and disadvantages. This also means that each medium comes with its own challenges for the cognitive workload, leading to a development of different skills depending on what medium is used. Greenfield mentions for instance that the Internet seems to enhance visual skills rather than a deeper understanding of the information, such as doing more critical thinking, analysis, reflection and also in using the imagination [35].

2.5 User Experience

Some argue that usability does not capture all aspects of a product since it is mainly focused on tasks and the concept has long been work related, thus ignoring other usage of digital artifacts. This has led to the rise of user experience design, which also captures other aspects of a product than the traditional usability definition do. Some examples of this is for instance that a products affective, experiential or aesthetic values are of interest [36].

It is the experience that a user has with an interactive system that is of interest within this concept, which is mentioned by Law et. al as something that is highly dependent upon the context in which this interaction take place and that the experiences can be unique for each user. However, Law et al. also mentions that the experience can be seen as different components that together makes up the experience, the components can for instance be motivation, fun, trust or hedonic.

Usability and user experience design is not two separate things, but are interconnected. As an example Law et. al says that it is the usability that sets up all the conditions in an interactive system so that it is possible for a user to smile when using it, but it is the user experience design that actually makes the user smile. Due to this, user experience design is for some seen just as a more fuzzy extra version of usability that focuses on qualities that the core usability concept does not take into account [37].

In a study by Bargas and Hornbæk 51 publications from 2005 until 2009 was reviewed with the goal of study how research within the area was conducted. When they looked at the dimensions of experience within user experience research they found that the three most studied dimensions were emotions and affect with 25%, enjoyment with 17%, aesthetics with 15% and hedonic qualities with 14% [38].

The user experience is interesting to look at in this study as well as usability since it gives more information and aspects of the use of a product than just one of the concepts would do alone. From the usability aspect information about performance can be measured while the experience, emotions and thoughts of the differences between the frameworks were collected through the focus on the user experience. In that way a bigger picture of how the differences between the frameworks are facilitated is possible to study.
3 Theory

3.1 Online Articles

By searching online for articles written on the topic there is a lot of different sources claiming that the usability and the user experience is better on a native application compared to a hybrid or web application [6] [7] [8] [9].

Saccomani mentions in one article a few advantages of choosing a native application when it comes to the usability and user experience, for instance he points out that the native application is much faster than both the web and hybrid applications but also offers a better responsive experience. The native application also has functionalities that are more easy accessed, which for instance is the camera, but the swipe gesture is also mentioned as something that is better with that framework. By choosing a native framework the application also has to match the HCI guidelines set up for that particular framework, which makes the applications more familiar for the users since most applications on their mobile will have similar UI and UX design [9].

Hybrid applications are also mentioned by Saccomani who says that they are good to use if you want to build an application fast and see if it will be liked or not by the intended users. The hybrid application still has access to platform features such as the camera, but on the other hand the performance is worse compared to the native application. Another disadvantage is that the user experience will not be as good as on the native application since the familiar UX and UI for each specific platform will be lost. The loss of familiarity can be frustrating for many iOS and Android users since they according to Saccomani are very loyal to the particular platform that they have chosen to use [9].

In another article Marketing & Growth Hacking Publication says that the native application offers a more reliable user experience compared to both web applications and hybrid applications, what is meant by reliable is however not specified any more in that particular article. Other features such as a better performance and more features that can be accessed is also mentioned as advantages while the articles emphases the disadvantage of the hybrid application limited speed when loading the page [7]. This is also mentioned by Menon who says that the native application has a high speed, access to platform specific features and UX/UI while the hybrid application has a medium speed and moderate access to platform specific features and UX/UI [8].

The choice between hybrid and native application development is also mentioned by Abed, who sees the familiar and consistent UI and UX within the native application as one important part of the user experience since it makes it possible for the user the faster learn and use the application. The expectations of the users is of high importance as well, since the users will expect the application to be fast, responsive and easy to use as soon as they have downloaded it and opens it. If the users finds the user experience poor 92% of the users will do something about it, from never using it again to giving it a bad review. Since the user experience is something that is affected by the framework according to Abed, the absolute best choice to get it right is to choose the native application framework. The reason for this is that even though a hybrid application is being designed with a great user experience it is almost impossible to make both iOS and Android users satisfied with the trade offs made from the familiar look and feel of their particular platform. The guidelines between these two platforms are a too great gap to bridge in one design and thus trade offs are made [6].

The articles mentioned are not of a scientific kind, but are written by companies and professionals within the field and can thus contribute with knowledge from the industry.

3.2 Current Industry Knowledge

To get a better understanding of how much the industry knows about the differences in
usability and user experience between hybrid and native frameworks semi-structured interviews were conducted. One of the interviews were with Sara Ingmars, a developer working at a technology company called Valtech. Regarding the hybrid frameworks she mentions positive aspects such as the possibility to develop and maintain one application for all different platforms, reducing both time and costs for development. However, the native framework is better when it comes to using graphical elements and interactions that are already standardized in each platform. One example that Ingmars brings up is the Windows phone where a panorama view is used and implemented on the platform, something that might be hard to mimic in a web based hybrid application, but easy to take advantage of in a native application. Other differences could be noticed in the interaction as well, for instance there is a quicker response on native applications due to a default delay that comes with the web and hybrid applications. However, this could be reduced by using certain libraries designed to fix this problem [39].

In an interview with Johan Nenzén, lead android developer at tv.nu, he says that the hybrid framework is slower and not as responsive as the native framework. It can also be hard to adapt the applications to the different screen sizes of mobile phones in a hybrid application, something that is easier done in a native framework. Nenzén also mention that the hybrid frameworks does not use native components, which makes them look and act differently than what the user is used to and the users can find it harder to navigate in the application compared to a native one. Some examples that he mentions is that the animations and the buttons can differ from the standard designs used on the platform. Nenzén also mention that the hybrid frameworks does not use native components, which makes them look and act differently than what the user is used to and the users can find it harder to navigate in the application compared to a native one. Some examples that he mentions is that the animations and the buttons can differ from the standard designs used on the platform. In a more historical perspective Nenzén also mentions that a lot of companies started using hybrid applications a couple of years ago, but due to the lower quality of the hybrid applications companies like Facebook started using native applications instead. This means that it can be hard to go back to hybrid again, mostly due to the fact that the platforms are so different and it is important to be able to utilize the platform to its full potential [40].

In the third interview with Sven Jonsson, developer at North Kingdom, aspects such as performance is brought up as an important aspect that might affect the applications. He says that the native framework has a better performance, even though this is highly dependent on what type of hybrid framework is used. The design guidelines are also different on Android and iOS which means that two different designs should be made on native applications, but that is not always the case according to Jonsson.

Jonsson also mentions that the iOS applications tends to be better made and includes less bugs since most time is spent developing and testing that version of the application, making the Android version less worked through. This can also be seen in application designs where applications look as if they are made for iOS, even though they are made for both platforms. One example of this is for instance when the back button is forgotten but there is instead a visible arrow to go back, which is a design made for an iOS application. On Android there is a physical button that is supposed to be used to go back, but that is neglected in some cases to make the applications look alike on the different platforms, or they might just be forgotten [41].

### 3.3 Research Articles

In one study Holzinger, Treitler and Slany focused on a project at a company where the goal was to build user interfaces on a number of different mobile platforms. In this study the authors say that developing an application with HTML5, CSS3 and JavaScript makes it possible to only develop one application since all mobile devices today has the capability to run those applications. While developing a native application means that you have to make several applications to be present on different platforms. However, The native application has access to the platforms features
such as the camera, the performance is overall better, a place in the app store and the application will have a familiar look-and-feel for the user. In the end the authors of the study says that the choice on what type of framework to choose is dependent upon a number of factors. Some of these factors is for instance the intended users, the features in the application and the programming skills of the development team [5]. Even though this study did not come to a conclusion about which framework that is better for the usability the articles still mentions important aspects of the frameworks that might influence the usability. For instance the familiar look-and-feel of the native application might affect the users feeling of satisfaction with the application.

However, the familiarity and recognizable part of an application might not just come from internal HCI guidelines for the different platforms, which is mentioned by Joorabchi, Mesbah & Kruchten [10] in their article called “Real challenges in mobile app development”. In that article they studied the challenges that the development team faces when developing apps for different platforms. It is a qualitative study and the results shows that the fragmented market for applications came with several difficulties, including a dilemma between using consistency in the app or consistency with the HCI guidelines for each mobile platform and operating system. For the company it is ideal to develop one application that looks the same across all different platforms, but on the other hand the developers have to adapt to the guidelines set for the different platforms. The latter is good for the end users, since it makes it easier for them to interact with several applications, while the first example is something that is more appreciated by developers that do not have to create and design two different applications due to the platform differences [10]. Joorabchi, Mesbah & Kruchten studies the developers of mobile applications, which only gives the developers perspective and thoughts about the mobile development market and not the users perspective. Still it raises interesting questions relevant in this study since the HCI guidelines within each platform or within the company might affect the usability and user experience of an application.

Another study that also emphasizes the importance of consistency is a study made by Mirkovic, Kaufman & Ruland. They studied the usability of an application for illness management for cancer patient. The results showed a number of design issues and violations against usability principles. One of the findings showed that many patients relied on their previous experiences when creating a mental model of the application. This lead to the users requesting help in situations when the system did not match the expectations that the user had on the mobile device or the web version of the system, which means that the match between the real world and the system was not fully supported. In the study the authors recommend the development of a native mobile application since the features of that application should be adapted to resemble the standards for that particular device and thus support the user’s mental models of the application [42]. The user’s previous experience can be seen as important in this particular study and might also affect the usability, which might affect the results in this study as well.

Some of the differences in the user experience is mentioned by Serrano, Hernantes & Gallardo who besides pointing out that the native applications gives a more responsive user interface as well as a better experience also mentions some more exact interactions that is of importance. For instance, he mentions that the possibility for the user to open the application fast is important as well as the native application’s specific hand gestures that also influences the user experience [43]. This might be interesting in this study since it can influence the result of for instance the usability tests, but also works as input to the tasks that is used in the tests.

A study looking at the difference between platforms is done by Ryan & Gonsalves who did a study where they compared four dif-
different application types: PC web based, PC device based, mobile web based and mobile device based. The results showed that the usability of the mobile web based application was the lowest of all the different types studied. One reason for that is the fact that the web based application did not take advantage of the location context or client-side application code. In the study it was also seen that the PC device based application had a lower performance and the lowest subjective liking of the users. The researchers were surprised by that result, but found a couple of reasons for it. For instance, familiarity with the web applications on PC was one important factor since that made the user prefer that kind of application and also criticizing the device specific PC version due to the layout being different from what they were used to [44]. Once again, the familiarity with the application is important and thus affects the application’s usability and user experiences which might also be reflected in the results in this study. Even though the study was made over 10 years ago the results is still interesting, especially since the usability differences between web and native applications on mobile was mention and the results from this study might indicate if the choice of frameworks still matters and in that case how much it matters today.

The null-hypothesis in this study is that there is not any differences between the frameworks in terms of usability and user experience. The hypothesis have been divided into narrower and more easily measured sections that is linked to the tasks that were tested. To clarify the narrower hypothesizes some definitions are needed. Orientation is in this study defined as the direction which the user is traveling on the screen, for instance vertically or horizontally. The hierarchy is defined as something that takes several steps, or levels, to go through in an application.

- Task 1 - A hierarchical problem where the user has to take several steps to complete the task.
  - H0 - There is no difference in the usability and user experience when the user solves hierarchical problems.
  - H1 - The native application has a better usability and user experience when the user solves hierarchical problems.

- Task 2 - An orientation and hierarchical problem where the user has to swipe in a horizontal direction to solve one task that takes several steps to complete.
  - H0 - There is no difference in the usability when the user solves orientation problems on the horizontal axis and hierarchical problems.
  - H1 - The native application has a better usability when the user solves horizontal orientation problems and hierarchical problems.

- Task 3 - An orientation and hierarchical problem where the user has to swipe in a vertical direction to solve several tasks that takes several steps to complete.
  - H0 - There is no difference in the usability when the user solves orientation problems on the horizontal axis and hierarchical problems.

3.4 Hypothesis and Null-hypothesis

Based on the current knowledge from previously mentioned online articles, scientific articles, books and in the interview conducted with an industry company working with the frameworks it becomes evident that the native framework is seen as the best option in terms of usability and user experience. The hypothesis is that the native application is better in terms of usability and user experience due to users familiarity with the native frameworks visible appearance and interaction, but also due to the native being accessed directly and not through a web-view.
- H1- The native application has a better usability when the user solves vertical orientation problems and hierarchical problems.

- Task 4 - An orientation problem where the user has to swipe in a vertical direction to solve one task.

- H0 - There is no difference in the usability when the user solves orientation problems on the vertical axis.

- H1- The native application has a better usability when the user solves vertical orientation problems.

When comparing the frameworks there is aspects of the interaction that is not possible to measure but that might influence the results. For instance the users own experience of the applications look and feel that might lead to users interpret the hybrid application as less satisfying due to slower response in the buttons. The differences in the look and feel might be something that affects the users experience of the application and the satisfaction questionnaire.

4 Methods

4.1 Prestudy Interviews

To be able to explore a phenomena a qualitative research method can be used, which for instance includes semi-structured interviews where questions like ”why” and details about the area of interest can be further explored [45]. This was done before the study started when three developers were interviewed about their experience from working with hybrid and native applications. The interviews were conducted through phone calls and video chats and the questions asked were centered around the experiences they have with the different frameworks, what the advantages and disadvantages of the frameworks are and what the future for the industry might be.

4.2 Development of Test Applications

The goal was to create one design that worked in both frameworks and that also included the tasks, which can be found in Chapter 4.6.1. The design was only made for iOS to limit the scope since tests and development would increase drastically more in time if yet another version was going to be tested.

4.2.1 Design Phase

First sketches were done with pen and paper to put down the ideas and after some iterations a first digital design was made in black and white to look closer at the structure. Feedback was given from Caspeco’s UX/UI-designer Aino-Maria Kumpulainen to further improve the design. Color was added and after a few more iterations the design was set with minor changes occurring during the implementation phase. The rough draft of the development of the design can be seen in Figure 2, Figure 3 and Figure 4.

4.2.2 Implementation Phase

When the design was finished the implementation of it began. First the native application was developed and when that was finished the hybrid version was developed. During the last weeks of development bug testing were con-
ducted on both applications to make sure that they worked correctly. A comparison between the applications to adjust the styling to be as close to each other as possible were also made. The final result of the restaurants screen can be seen in Figure 5 and Figure 6.

4.3 Measures

In this study several methods have been used to gather data to be able to answer the research question and the methods are presented below. A scenario-based approach has also been used in this study, this means that stories are written about how a person should act or do. By using this approach it is possible to see concrete and detailed design issues that can occur when the users does and action and uses an application. This will give a coherent and concrete understanding of the situation, which contrasts to a list of requirements that just are functional goals [46]. Thus this method made it possible to explore the actions that the users do and how these actions might differ between different frameworks. The scenarios are however called tasks in this paper since the goal is for the user to finish those tasks, even though they are em-
4.3.1 Quantitative Research Design

When doing quantitative research the goal is to see if there is some kind of relationship between the two variables of interest: the dependent and independent variable. The goal is thus to get measurements of these to be able to tell if changes in the independent data affects the dependent data. This is often done with a lot of participants to get statistically valid data from the studies and thus give information that for instance could be used on a part or the whole population [47]. In this study the independent variable was the two frameworks studied, while the dependent variables were the usability measurements that are further described under point 4.4.

4.3.2 Qualitative Research Design

In a qualitative study more focused on exploring a phenomena and asking questions that explores this, like "what", "why" and "who", and is thus not focused on counting how many participants that for instance did a certain thing. This means that the overview is not always the biggest interest, instead it is the details and the richness off the data that is of importance. The exploration of the chosen phenomena is often done through the participant’s perspective, or at least uses them as a starting point in their research. There are several methods that can be used in qualitative studies, whereas one commonly used are semi-structured interviews [45]. In this study a qualitative method was used to explore more of how the user experience is affected by the different frameworks.

4.3.3 Mixed Method Research Design

By using both qualitative and quantitative method it is possible to combine the result to get information about the same phenomena or to get information about different views on
how to look at it. The combination of different methods are called mixed method research design [48] and was also used in this study. As mentioned earlier the usability and the user experience are both concepts that looks at how the user interacts with a product in a given context, however by using results from a quantitative measurement and a qualitative measurement the goal was to get a greater insight into the differences that the users experiences.

4.4 Usability measurements

Effectiveness, efficiency and satisfaction can be used to measure usability, which is mentioned buy ISO 9241-11. These measurements should be made in a certain context with specific goals by a group that is also specified and has a particular goal to reach [31]. In this study the effectiveness, efficiency and satisfaction were studied and compared between the different frameworks to see how the usability differed between them. The data gathered about the usability was made numerical and thus gave data to the quantitative part of the study.

Effectiveness

Effectiveness is concerned with the completeness but also the accuracy in which users reach the goal that has been set out to them [31]. In this study this meant that the number of finished tasks were registered as true or false, depending on if they finishes the tasks as they were set out in the task descriptions or not. By using the average finishing rate the effectiveness was possible to compare between the different frameworks.

The effectiveness was also measured in terms of how many errors the users did on their way towards their goal. An error could for instance be that a user pushed the wrong button, made a wrong spelling or clicked on something that was not even possible to click on. The average number of errors was later compared between the different frameworks.

<table>
<thead>
<tr>
<th>Nr</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I thought the app was unnecessarily hard to use</td>
</tr>
<tr>
<td>2</td>
<td>I thought the app was easy to use</td>
</tr>
<tr>
<td>3</td>
<td>I thought the app was inconsistent</td>
</tr>
<tr>
<td>4</td>
<td>I felt confident using this app</td>
</tr>
<tr>
<td>5</td>
<td>I thought the system was very difficult to use</td>
</tr>
</tbody>
</table>

Table 1: Satisfaction Questions

Efficiency

Completeness and accuracy is also of interest when measuring efficiency, but here it is the resources that it takes to reach a specific goal that is measured [31]. The resources were measured as time spent on each task and the number of clicks needed to finish or quit a task. The average of these measurements of efficiency were later compared to see the differences between the frameworks.

Satisfaction

Positive attitudes from the users and also their feeling of discomfort when using a product is what is of interest to know when looking at satisfaction [31]. A standardized post-study usability questionnaire called System Usability Scale (SUS) was used to measure the satisfaction of the users. SUS has proven to be a reliable scale for measuring users satisfaction both in the industry and within research, which has led over 5 000 references [49]. To better fit the purpose of this study and the nature of the system the test was modified. For instance the questions were remake so they were directed towards the application instead of a system and some questions were also removed from the test since they would not bring useful information for this particular study, the questions used can be found in Table 1. The usability questionnaire was handed to the users after they had finished...
the task on one platform and thus give insights into the user’s thoughts and opinions about that particular framework. A Likert-scale was used where the users can give answers ranging from 1, which means that they strongly disagree with the statement given, to 5, which means that they strongly agree with the statement.

4.5 Interviews

A method that is usually used in qualitative studies are interviews since the method is a good way to get information about people’s experiences and opinions of a certain phenomenon [45]. Interviews can be structured in different ways, some have standardized questions that are to be answered in a certain order without extra questions, while others are totally free of rules. However, the most commonly used structure in interviews in qualitative research is semi-structured interviews. This approach means that the researcher has prepared some questions or themes that works as a guide for the interview but new questions are also welcome, making it a more flexible interview form. The advantage of using this method is that the researcher can set the agenda on what to talk about, but the interviewee can still give spontaneous reactions that the researcher can dig deeper into if it is of interest [50].

Critique has been raised against this method since it relies too much on the individual participants and their experiences and thoughts. One other critic is that the researchers themselves might influence the participant’s answers depending on aspects such as how the questions are formulated. The method is however legitimized by most researchers in the area and gives individual insights in detail that is of interest within research [50].

Interviews were used in this study since the method gives details about a topic and can give insights into the users thoughts and opinions, something that is not possible to collected through the quantitative method used.

<table>
<thead>
<tr>
<th>Interview Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
</tr>
<tr>
<td>Overview</td>
</tr>
<tr>
<td>Look</td>
</tr>
<tr>
<td>Feel</td>
</tr>
<tr>
<td>Familiarity</td>
</tr>
<tr>
<td>Performance</td>
</tr>
<tr>
<td>Frustration</td>
</tr>
<tr>
<td>Enjoyment</td>
</tr>
<tr>
<td>Other</td>
</tr>
</tbody>
</table>

Table 2: Interview Categories

As mentioned the user experience approach is focused on values and the users own thought about the use of a product, something that was possible to explore more through this method.

The questions asked during the test were focused on user experience differences between the frameworks, but also more specific questions were asked. The questions were divided into eight categories which can be seen in Table 2. The categories were chosen from Chapter 2 in this report and thus it was possible to explore more of the factors that are mentioned as possible differences in experience between the frameworks. Additional questions were asked about the feeling of using the application since that gives the user an opportunity to explore and think more about the different ways of interacting with it.

4.6 Test session

4.6.1 Tasks

From the background and theory given there are some aspects of the usability and user experience that have been mentioned and that also was studied closer. It has been mentioned that the native framework has a more familiar look and feel to it [5] and that the hand gesture movements are more well integrated [9] compared to applications made with the hybrid framework. To tests this four tasks
were chosen where these particular areas were tested.

The tasks are the following:

1. Task 1 - Log in
   
   • "You and your friends have not eaten in several hours and now plans on going out for some food. You do not know where you want to go and wants to scan the options that are close to you before choosing. You downloaded a new application called "Yumi" a few weeks ago and suggest to your friend that you can look through that app for inspiration. You open the application and log in with the username 1234 and the password 1234.”
   
   Task: Log in to the application

2. Task 2 - Book one restaurant
   
   • "A few favorite restaurants have been saved in the application since earlier and from the list of favorite restaurants you and your friend decides to go to the asian restaurant "Miss Voon" and books a table there for two persons next Wednesday at 21.00. You book it with the name Kim, and with the phone number 12345 and with the email address kim@gmail.com”
   
   Task: Find the restaurant "Miss Voon" in the favorites list and book a table.

3. Task 3 - Book several restaurants
   
   • "You have two friends coming to town this weekend and they both want to explore the city’s restaurants so they ask you to book dinner tables for them on both Friday and Saturday night. For Friday evening you decides to go to “Plock” and books a table for three at 20.00. You book the table with the name Kim, with the telephone number 12345 and with the email address kim@gmail.com. For Saturday evening you decides to book a table for three at the Italian restaurant “Il Forno”. You do this with the same name, phone number and email address as you did for the Friday booking”.
   
   Task: Book one table on Friday and one on Saturday.

4. Task 4 - Searching for an email address
   
   • "You want to send an email to the company behind the Yumi app and tell them what you think of it, but to do that you first need to find the email address”.
   
   Task: Find the email address

Task 1 is formulated so that the user can see how it is to enter the applications and if there is any differences there. This task also includes the user putting in data in the system, for instance username and password, something that will take several steps to complete making this task focused on the hierarchal procedures that can occur in applications.

Task 2 and 3 contains an element of scrolling in different directions, task two for horizontal scrolling and task 3 for vertical scrolling. This occurs when the user swipes through the different restaurants that can be picked. This is to make it possible to compare the hand gesture and see if there is any differences but also to make it possible to study the orientation hypothesis about hand gestures used in the different direction. Task 2 and 3 is also focused on the hierarchical procedures since there are a lot of steps needed to complete the task of booking one or two restaurants.

In task 4 it is possible to look at a scrolling motion in the vertical direction, where the goal is to find the phone number on the bottom of the main page.

After each task the word 'Task' is written as a summary to what the intended goal of the task is but also to clarify for the user when
<table>
<thead>
<tr>
<th>User number</th>
<th>Hybrid</th>
<th>Native</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test 1</td>
<td>Test 2</td>
</tr>
<tr>
<td>2</td>
<td>Test 2</td>
<td>Test 1</td>
</tr>
<tr>
<td>3</td>
<td>Test 1</td>
<td>Test 2</td>
</tr>
<tr>
<td>4</td>
<td>Test 2</td>
<td>Test 1</td>
</tr>
<tr>
<td>5</td>
<td>Test 1</td>
<td>Test 2</td>
</tr>
<tr>
<td>6</td>
<td>Test 2</td>
<td>Test 1</td>
</tr>
</tbody>
</table>

Table 3: Test order

she or he has reached the finish line and see
the task as completed.

4.6.2 User Groups

In Sweden 78% of the population uses mobile
phones, and 65% uses them on a daily basis, whereas people between 16 and 25 years of age
uses their mobile phone 40 hours a week on
average. The mobile phones are used among
both adults and children [51], but to limit the
age span this study focuses on users who are
12 years and older.

Another limitation of the users who partic-
ipate in the study is their experience of the
use of mobile application, as mentioned many
uses their mobile phone on a daily basis and it
is those persons which were used in this study
as well. This is done to make sure that a basic
understanding of how a mobile phone works is
present and to avoid errors caused by inexpe-
rience in device usage. However, this means
that there are differences in the users experi-
ence in mobile application usage, but a basic
knowledge will still be present.

The users were divided into two groups; the
group that starts with completing tasks in the
hybrid framework and the group that starts
with completing tasks with the native frame-
work. The reason for this to avoid the learning
affects that comes with using an applica-
tion. Which framework that each user started
with was thus based on what the previous user
had tested, but made opposite.

The order in which the tests were conducted
can be seen in Figure 3

4.6.3 Test Site and Equipment

When doing usability tests on mobile phones
it is possible to do those in for instance a lab-
atory or out in the field, this can however
affect the results of the study. In a study
by Duh, Tan & Chen [52] they found that
many more types of usability problems were
found when a user test was done in the field
compared to the laboratory. A problem with
the laboratory tests were that some problems
were only found when used in a context, and
thus missed in the laboratory tests [52].

Due to the risk of missing some usability
problems laboratory tests were avoided in this
study. The test was conducted with one user
at a time in a setting where the user could sit
down and test the applications without be-
ing disturbed by others. The user was given
an Iphone 7 to use during the session and a
screen recording software was used during the
test session to capture what happened on the
screen to be able to count clicks and detect er-
rors. To record the number of clicks the users
did on the screen another mobile phone, an
Iphone, was used to record the users hands,
thus making it possible to see what was hap-
pened on the phone from two separate views.
Besides the user, the test leader were also
present in the room to guide the user through
the test.

4.6.4 Procedure

When the user entered the room the user was
first given a consent form informing the user
about how the data was going to be used and
what the purpose of the study was. When
that paper was signed the user was given a
demographic form to fill in with information
about their age, familiarity and usage of mo-
bile phones and the brand of the phone they
have for the moment. After that the user was
informed of how to start and end a task, which
was by flipping a paper, and then given the
Iphone 7 and the first task to read through.
After the first task the user got the next one
and so on. During the tasks the user finger
movements were filmed and the screen was
Table 4: Test order & Task order

<table>
<thead>
<tr>
<th>User number</th>
<th>Test order</th>
<th>Task order</th>
<th>Hybrid</th>
<th>Native</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test 1</td>
<td>Test 2</td>
<td>S1</td>
<td>S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S2</td>
<td>S2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S3</td>
<td>S3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S4</td>
<td>S4</td>
</tr>
<tr>
<td>2</td>
<td>Test 2</td>
<td>Test 1</td>
<td>S1</td>
<td>S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S3</td>
<td>S3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S4</td>
<td>S4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S2</td>
<td>S2</td>
</tr>
<tr>
<td>3</td>
<td>Test 1</td>
<td>Test 2</td>
<td>S1</td>
<td>S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S4</td>
<td>S4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S2</td>
<td>S2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S3</td>
<td>S3</td>
</tr>
</tbody>
</table>

recorded as well.

When the first four tasks was completed the users were asked to fill in the post-study usability form about satisfaction and after that they performed the same tasks, and in the same order as in the first round, but on the framework that the particular user had not been testing before. The tests were conducted exactly as on the first framework test where the users read the tasks and then flipped the paper when they started and reached the finish line. The post-test usability form about satisfaction was also filled in after this test session was done.

As mentioned, the frameworks were tested in a stratified way with one user starting with the hybrid framework and then testing the native after that, and the next user testing the frameworks in the opposite order. The tasks were also tested in a stratified way where the first task always was the same, since it is about logging in to the application, while to other tasks were in different orders depending on previous user test. In Table 3 it is possible to see the order in which the frameworks were tested, and in Table 4 it is possible to see the tasks for the first three users.

After the tests were done on both frame-works the users had to answer a few interview questions. The answers were recorded through a mobile phone for later transcription. The questions asked were about their experience of the use of the two frameworks and their view on the differences between them. At the same time as the questions were being asked, the users had the opportunity to test the applications and switch between them freely to further investigate the applications without having to focus on a task. After the interview the test was completed.

An overview of the procedure can be seen in Figure 7 and a picture from one of the user test session can be seen in Figure 8, where the phone, recording setup and start/stop-button is visible.

5 Prestudy tests

Before the real tests begun, three prestudy tests were conducted to be able to detect flaws in the test and improve the method. The prestudy tests revealed that the best method for starting and stopping each task was to hit the start and stop-button instead of turning it around since it was a faster and more convenient way of interaction. It was also evident that the users had problems understanding the core of the tasks in some cases and thus "task" was added as a conclusion of the main task. Another change made to the tasks were that the key words where underlined so it was easier for the users to see for instance time, place and the number of participants. When the changes were made and tested in the prestudy, the real tests began.

6 Data collection

As mentioned the data was collected through several different methods. To gather quantitative data screen recording, filming and survey answers were used to gather information about the effectiveness, efficiency and satisfaction.
The effectiveness was measured in terms of completeness, which meant that the users had to reach the goal of each task. The goal for task 1 was to log in to the applications with the right username and password. The goal for task 2 was to book a table for two persons at next Wednesday at 21.00 at Miss Voon by scrolling through the favorites list. The goal for task 3 was to book a table for three persons at 20.00 at Plock on the next Friday and to book a table for three persons at Il Forno on Saturday on whatever time the user wanted to. Both in task 2 and 3 the user had to book the tables with the right information as well, which was under the name Kim, with the email address Kim@gmail.com and with the phone number 12345. The goal for task 4 was for the user to find the email address.

The effectiveness was also measured by the number of errors made during the task, which for instance was when a user clicked in on Miss Voon from the close by list instead of the favorite list in task 2 but also misspellings or if they booked a table with the wrong phone number.

The efficiency was measured by the time
it took in seconds for the user to complete a task. The start time was when the user pushed the start button specifically made for the tests and the stop time was when the user hit the button once again. If the user forgot to push the button to start or stop the start time used instead was when the user first clicked on the screen to start the task or when the user made its last click to finish the task.

Efficiency was also measured by counting the number of clicks the users made during the time the task was executed. This could for instance be when the user was writing something, hitting a button or when they were scrolling. When they were scrolling a click was only counted when the finger touched the screen and then lifted again, if the user for instance held down the finger continuously and toggled up and down on the screen this was only counted as one click since the finger was not lifted from the screen.

To get information about the satisfaction the users had to fill in a satisfaction questionnaire after all the four tasks were finished. The statements could be answered on a Likert-scale from one to five, where one means that the user disagrees on the statement and five means that the user agrees on the statement. Question one, three and five are negatively formulated and if the user answers one on that one it is a low satisfaction grade for the application. Statement two and four are positively formulated which means that if the user answers one on that the application gets a high satisfaction grade. However, the grades for question two and four was later converted to match the scale of the other question, which means that if a user gave an application a grade of 2 on question 2 it was later converted to a 4. This made it easier to compare the result and see the overall trend.

Semi-structured interviews was used to gather qualitative data. The user had to answer questions that was formulated from previous research and industry knowledge about the usability and user experience differences between the frameworks that can be seen in Table 2. During the interviews the user’s voice were recorded from the phone and those answers were then coded into a Google Sheets where the information later was categorized in answers such as positive, neutral and negative to further investigate the material.

7 Data analysis

7.1 Analysis of Quantitative Data

When the quantitative data was gathered it was analyzed using a statistical program called SPSS. Through that program, data such as mean, maximum value, minimum value and standard deviation could be gathered and used for the purpose of this study.

The raw data about each of the usability measurements can be seen below and are listed here to get a better understanding of the raw data and the result of this study. On the X-axis of each of the diagram presented below the number stands for each participant, thus number 10 is left blank since that test could not be analyzed. On the Y-axis the different measurements could be found, for instance completeness time in seconds or number of clicks.

7.1.1 Effectiveness

Effectiveness was measured by completes, which either could be coded as 1 = completed or as 0 = not completed. The overall completeness distribution for each user can be seen for both frameworks in Figure 9. In the figure it is possible to see that most users could complete all the four tasks and that some users could only complete three. One user only complete two out of the four tasks.

When the highest possible value, or close to the highest, is reached in a measurement it is called the ceiling effect. This is a sign of a measurement not being able to accurately measure what was intended to measure or that it has not been possible to capture the variation that might occur [53]. In the case of the measurements of completeness the ceiling
The effect is present, since almost all users completeness the tasks and thus it is hard to see an actual difference in this measurement between the frameworks.

The distribution of the number of errors made by each users on each task can be seen in Figure 10, Figure 11, Figure 12 and Figure 13. There it is possible to see that there were only a total of 5 errors made on task 1, three on hybrid and two on native and most errors were made by users later on in the test session. A lot of errors were made in task 3 in both frameworks.

The range between the minimum and maximum number of errors made by the users can be seen in Table 5, where it is possible to see that the error range between the frameworks are quite similar.

<table>
<thead>
<tr>
<th>Task</th>
<th>Native</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>0 - 1</td>
<td>0 - 2</td>
</tr>
<tr>
<td>T2</td>
<td>0 - 4</td>
<td>0 - 2</td>
</tr>
<tr>
<td>T3</td>
<td>1 - 7</td>
<td>0 - 6</td>
</tr>
<tr>
<td>T4</td>
<td>0 - 5</td>
<td>0 - 5</td>
</tr>
</tbody>
</table>

Table 5: Range Tasks - Errors
7.1.2 Efficiency

The efficiency was measured in terms of how many seconds it took for a user to complete a task and the distribution of the time it took for a user to complete a task can be seen in Figure 14, Figure 15, Figure 16 and Figure 17. There it is possible to see that the average time spent on task 2 was in many cases quite similar, which is around 50 seconds. However, on task 4 the difference were bigger. Many users completed the task under ten seconds, but for some users it took almost 80 seconds to complete the task.

The minimum completeness time and the maximum completeness time for each task can be seen in Table 6, where it is for instance possible to see that the minimum completeness time was quiet similar between the two frameworks, however the maximum completeness time varied a lot on all task except task 1. For instance, on task 2 the difference be-
The distribution of the number of clicks each user did for each task can be seen in Figure 18, Figure 19, Figure 20 and Figure 21. The number of clicks for task 1 was for almost all users around 13, even if some users had both a lot more, for instance user 15 who made around 30 clicks and user 14 who made almost no clicks. For number 14 the number of clicks was low due to a miss click that led the user straight pass the sign in procedure and in to the application.

The minimum and maximum number of clicks for each task can be seen in Table 7, where the difference is quite similar on task 2 and task 4. The difference is a little bit bigger between the minimum number of clicks on task 3 on native compared to hybrid and the minimum and maximum value is also lower on native in task 1, however the low minimum number on task 1 is an error and is thus somewhat misleading.

<table>
<thead>
<tr>
<th>Task</th>
<th>Native</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>6 - 37</td>
<td>7 - 32</td>
</tr>
<tr>
<td>T2</td>
<td>33 - 154</td>
<td>26 - 73</td>
</tr>
<tr>
<td>T3</td>
<td>63 - 233</td>
<td>62 - 141</td>
</tr>
<tr>
<td>T4</td>
<td>4 - 38</td>
<td>3 - 77</td>
</tr>
</tbody>
</table>

Table 6: Range Tasks - Time

<table>
<thead>
<tr>
<th>Task</th>
<th>Native</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>1 - 16</td>
<td>13 - 29</td>
</tr>
<tr>
<td>T2</td>
<td>39 - 67</td>
<td>36 - 53</td>
</tr>
<tr>
<td>T3</td>
<td>52 - 138</td>
<td>87 - 136</td>
</tr>
<tr>
<td>T4</td>
<td>3 - 36</td>
<td>3 - 42</td>
</tr>
</tbody>
</table>

Table 7: Range Tasks - Click
7.1.3 Satisfaction

The results from the satisfaction questionnaire is not possible to divide on each task since the questionnaire was answered after all the tasks were completed, however the distribution of the answers can be seen per question instead. This can be seen in Figure 22, Figure 23, Figure 24, Figure 25 and Figure 26. The questions could be answered on a Likert-scale from one to five, whereas five means that the user agrees with the statement. Overall the users were very satisfied, with most answers being around 4.

7.2 Analysis of Qualitative Data

From the answers in the code scheme, trends were looked for in the answers. The positive, negative and neutral comments about the differences in the frameworks were also gathered
to see the difference in the user’s opinions. Some of the feedback mentioned by the users were however removed from the analysis since it was not relevant for the comparison between the frameworks. One example of this is comments saying that neither of the applications had a search field but also that the calendar in the hybrid application was in an American format while the one in the native version was in a European format. This made many of the users confused since they were not accustomed to the American format, however this is something that could be fixed by including another calendar and thus is nothing that is relevant for the comparison between the frameworks per say.

### 7.3 Results

#### 7.3.1 Demographics

21 people participated in the study, however one test (user 10) was not analyzed since the whole test was not recorded and thus the information could not be used. This means that 20 tests were used and analyzed and the user’s age spanned from 12 to 58 years, but most of the participants were in their 20’s. More information about the age of the users can be seen in Table 8. The distribution of the age can be seen in Figure 27, where it is evident that this study lacks user data from user within the age span of 30-40 years old and from those who are +60 years old.

To grade how often the mobile was used throughout a day the users could answer a form with three alternatives; 1. The whole day, 2. Partly throughout the day and 3. Once a day or more seldom. This was done to be able to get an understanding of the users experience in phone usage. The results showed that 55 % of the users used their phones partly during the day. No one used it less often than once a day (Table 10). This meant that the median was 2 in this case, and the mean 1.55, which indicates that the users on average rated their own use of the phone to be slightly more than partly throughout the day (Table 9).

#### 7.3.2 Usability

**Effectiveness**

Effectiveness was measured in terms of the average completeness rate and the average number of errors. The results showed that the completeness rate on average was lower for the native application on task 1 and 2, while the completeness rate was on average lower on the hybrid on task 3 and 4. This can be seen in Table 11 and in Figure 28.

<table>
<thead>
<tr>
<th>User Age</th>
<th>Mean</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30,6</td>
<td>24,50</td>
</tr>
</tbody>
</table>

Table 8: Demographics - User Age

<table>
<thead>
<tr>
<th>Mobile Usage</th>
<th>Mean</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.55</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 9: Demographics - Mobile Usage Percentage

<table>
<thead>
<tr>
<th>Mobile Usage Percentage</th>
<th>Answers</th>
<th>Frequency</th>
<th>Valid Percent</th>
<th>Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>9</td>
<td>45.0</td>
<td>45.0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>55.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td>100.0</td>
<td></td>
</tr>
</tbody>
</table>

Table 10: Demographics - Mobile Usage Percentage

![Bar Chart Age](image_url)

Figure 27: Age
When the average of the completeness for all tasks were compared the results showed that the average completion on native was 0.91 while the average completion on hybrid was 0.94. This means that the completion rate was higher on hybrid compared to native, which can be seen in Table 12 and in Figure 29.

The average number of errors were lower on hybrid in task 2, 3 and 4 while it was lower on native in task 1, which can be seen in Table 13 and in Figure 30.

When the average number of errors were gathered for all the tasks in each framework the results showed that the average number of errors on native were 1.28 while the average number of errors on hybrid were 1.05. This means that the average number of errors were lower on hybrid, which can be seen in Table 14 and in Figure 31.

<table>
<thead>
<tr>
<th>Task</th>
<th>Native</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>0.95</td>
<td>1</td>
</tr>
<tr>
<td>T2</td>
<td>0.7</td>
<td>0.85</td>
</tr>
<tr>
<td>T3</td>
<td>1</td>
<td>0.95</td>
</tr>
<tr>
<td>T4</td>
<td>1</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Table 11: Tasks - Completeness

<table>
<thead>
<tr>
<th>Task</th>
<th>Native</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>0.1</td>
<td>0.15</td>
</tr>
<tr>
<td>T2</td>
<td>1.4</td>
<td>0.8</td>
</tr>
<tr>
<td>T3</td>
<td>2.25</td>
<td>2.3</td>
</tr>
<tr>
<td>T4</td>
<td>1.35</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Table 13: Tasks - Errors

<table>
<thead>
<tr>
<th>Total - Completeness</th>
<th>Native</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.91</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Table 12: Total - Completeness

<table>
<thead>
<tr>
<th>Total - Errors</th>
<th>Native</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>1.28</td>
<td>1.05</td>
</tr>
</tbody>
</table>

Table 14: Total - Errors
Figure 31: Total - Errors

<table>
<thead>
<tr>
<th>Total - Time</th>
<th>Native</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>49.56</td>
<td>47.7</td>
</tr>
<tr>
<td>Std.</td>
<td>47.292</td>
<td>42.127</td>
</tr>
</tbody>
</table>

Table 15: Total - Time

Efficiency

Efficiency was measured in terms of the average time in seconds it took for the user to do the task and in terms of the average number of clicks done during that time.

The average time it took to complete a task was lower on native on task 1 and 4 while it was lower for hybrid in task 2 and 3, which can be seen in Table 17 and in Figure 32.

The average of the total time for each framework were also compared and the results showed that the average time to complete the tasks where 49.56 seconds in native and 47.7 seconds in hybrid, which can be seen in Table 15 and in Figure 33.

The average number of clicks was lower on native in task 1, while it was lower on hybrid in task 2, 3 and 4, which can be seen in Table 18 and in Figure 34.

The average number of clicks between each framework were also compared and the results showed that the average number of clicks where 44.38 in native and 41.3 in hybrid, which can be seen in Table 16 and in Figure 35.
Figure 35: Total - Clicks

<table>
<thead>
<tr>
<th>Task</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean N</td>
<td>12.4</td>
<td>57.3</td>
<td>113.2</td>
<td>15.3</td>
</tr>
<tr>
<td>Mean H</td>
<td>14</td>
<td>52.2</td>
<td>108.6</td>
<td>16</td>
</tr>
<tr>
<td>Std. H</td>
<td>5,804</td>
<td>13,037</td>
<td>23,596</td>
<td>20,971</td>
</tr>
</tbody>
</table>

Table 18: Std. Tasks - Click

<table>
<thead>
<tr>
<th>Satisfaction - Mean per Question</th>
<th>Native</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>4.25</td>
<td>4.2</td>
</tr>
<tr>
<td>Q2</td>
<td>4.05</td>
<td>4.15</td>
</tr>
<tr>
<td>Q3</td>
<td>4.45</td>
<td>4.35</td>
</tr>
<tr>
<td>Q4</td>
<td>4</td>
<td>3.8</td>
</tr>
<tr>
<td>Q5</td>
<td>3.8</td>
<td>4.05</td>
</tr>
</tbody>
</table>

Table 19: Satisfaction - Mean per Question

Satisfaction

Satisfaction was measured by letting the users answer five different claims about their experience of the applications after the tasks were finished. The claims were rated from 1 to 5 and the results show that the users on average thought that the application was unnecessary slightly harder to use in the hybrid application compared to the native application (See Q1 in Table 19). The users thought that the native application on average was harder to use (See Q2 in Table 19) and less inconsistent than the hybrid application (See Q3 in Table 19). The users felt on average more confident while using the native application (See Q4 in Table 19), but also rated the native application as more cumbersome to use (See Q5 in Table 19). The average for the users answers can also be seen in Figure 36.

7.3.3 User Experience

The answers from the interviews were divided into eleven categories, which is all the categories used during the interview but with the subcategories of "Feel" also included. The conclusion of the interviews can be found below.
Overview

In the initial category overview most of the users did not see any differences between the applications. Those who had the strongest opinions about the difference where often in favor of the hybrid application as a start, this was due to the fact that the application focused on the input field that was chosen when entering information, something that was missing in the native application which caused the keyboard to cover the input field. Another difference mentioned was that the users had to push the booking button twice in the native version, since it on the first click wanted the keyboard to go away and on the second click responded to the click of the button.

Look

Most of the users did not see any visual differences between the applications but those who did were often in favor for the native application. One of the users answered that “They [the applications] are built on the same design, but the second one is a little bit more fresh. I’m not sure, but maybe it is bigger images or something. However, visually it looks more attractive than the first one”. The visual differences that were most frequently brought up was the difference between the looks of the horizontal scroll where the hybrid version leaves an empty space in the end of the scroll, something that was not appreciated by the users.

Another visual difference mentioned was the position of the days in the calendar, where the hybrid version has an American format on the calendar which was not appreciated by the users who wanted a calendar they are more accustomed to, which is a calendar where the week does not start with a Sunday. However, those two differences are caused by the choice of library used in each case and is thus something that is not caused by the different frameworks used.

Other visual differences seen was the shadow on the restaurant buttons which were described as harder on the hybrid version, but also appreciated more, and that the text was bigger on the hybrid version, which was not liked by the user who mentioned it. Those differences where small and due to the font-size chosen and the settings for shadows and thus is possible to alter and thereby is not something that indicates a direct difference in appearance between the frameworks.

Feel - Vertical Scroll

The opinions about the vertical scroll where diverse. Many users said that the native applications scroll where both faster and slower compared to the hybrid one, however most users preferred the native application since it was “… more comfortable to scroll with” and “the first one [native] was slower and due to that I was able to register which restaurants that I scroll by, but in the other one I have to adapt my scrolling more so I’m able to see the restaurants”. The hybrid application was also described as more choppy.

Feel - Horizontal Scroll

In the native version the horizontal scroll was described as better in the sense that it “acted as I expected it to do” and since it “followed better when I scrolled”. However, the hybrid version was also described as better than the native since other users thought that it “rolled better” and that it “acted more as a rubber band”. The hybrid version also
showed an empty area in the end of the favorite scroll which many users also pointed out as something negative as it did not look as they thought it would or were familiar with. This is due to the carousel library used and is thus something that could be avoided. However, this can affect the overall result since it was disliked by many users, even though it has nothing to do with the frameworks but with the chosen library.

Feel - Buttons

The users were in favor of the native application and the animation that is triggered when a button is pushed and the user is taken to another page. The animation is a stack animation that causes the next page to slide in from the right side of the screen on top of the old one. In the hybrid version the next page just pops up as it is loaded, which is described by one user in this way: “It [hybrid] feels like it flashes when you open it. The other [native] sweeps in more softly. That makes me like the first one [native] more, the other jumps too fast and that is kind of annoying for the eyes”.

The hybrid version is not just experienced as to fast when it pops up, but also as slower since the users reported that the loading time was higher on that application. “It takes longer time for it to think before it pops up. That is bad, especially in today’s society when everything is supposed to go fast. I even had to wait until I could push the [stop] button”. This also caused the users to distrust the hybrid application more “When you had pushed the button I thought - Is it going to crack now?”.

However, the native application also got negative feedback due to the fact that the button did not respond straight away when the keyboard was visible on the screen, something that many users did not see as an obstacle for pushing the button that would take them to the complete booking screen. Thereby, the hybrid version was experienced as faster and not as inert as the native version.

Feel - Input

When the user put information into the applications the favor was often on the hybrid application. This was due to the fact that the keyboard covered the input fields on the native application while the hybrid application refocused so that both the keyboard and the input field was visible.

Another positive aspect of the hybrid application was the possibility to click on a ”Done” button or on arrows that could take the user to the next input field straight away. This came up with the keyboard and made it a lot easier for the user to interact with the application. A negative comment about the input field was that it blinked when the user pressed it, the margins where smaller and it felt ”tougher” to enter the letters.

Familiarity

Most users felt a stronger familiarity with the hybrid application since it did not have a keyboard that covered the input fields and since it was possible to jump between with arrows between the input fields.

Trust

Most users trusted the native application more, which was due to a variety of reasons. One user said it was “Faster, feels more professional and fun to use” and another said that “The first one [native] feels less buggy and is a little bit more perfect in the details”.

The hybrid application was only trusted more by one user, who said that the trust was higher since “The two things I saw bothered me a little bit, and those where not in the first one [hybrid]”. Other users missed the gesture possibilities that only were present in the native version and also said that they were afraid that the application would crash in the hybrid version. Another negative aspect was that the pages was not loaded as fast in the hybrid version as in the native.
Performance

The users more often mentioned the hybrid application as the one with slower performance, both due to click delay and the transition between pages which was described by one user in the following way: “It is a little bit slower when it is not the smooth transition...It does not feel just as modern in some way”.

However, the native application was also described as slower by some users, but to them that was something positive since “Then you are able to see the dishes more” and “I’m that kind of person that just happens to push on things and then I wonder where I end up, so I think it is good that it is slow”.

Frustration

Most users did not experience any frustration while using the application, however what was brought up was the keyboard that covered the input fields in the native version and that the hybrid application was slower.

Enjoyment

Most users enjoyed the native application more, which was motivated by one user as “Designally, they are very similar but application number two [native] expresses itself in a better way. It is more tempting than application number one [hybrid] and the usability is also a good point. It is nicer to scroll, easier to click, everything is much smoother and that is very appreciated when you are searching for a restaurant etc.”. Other aspects mentioned were the slower scroll that made it easier to see the restaurants and that it was not as slow and jumpy as the hybrid version.

On the other hand, a few users preferred the hybrid application since it did focus on the input field and thus avoiding the keyboard from covering the fields.

8 Discussion

8.1 Usability

The hypothesis in this study have been that the native framework is better in terms of usability and user experience than the hybrid framework, however this hypothesis was divided into narrower and more easily measurable sections that is closely linked to the tasks.

The hypothesis for task 1 was that native is better than hybrid in terms of usability when the user solves hierarchical problems. In terms of effectiveness the users had a mixed result, with a higher completeness on hybrid and a lower number of errors on native. In terms of efficiency the users performed better on native, both in terms of time and in terms of clicks, which indicates that the performance part of the usability was better on native than on hybrid in task one. Thus the hypothesis is true in this case.

The hypothesis for task 2 was that the native application has a better usability when the user solves horizontal orientation problems and hierarchical problems. In all measurements done within effectiveness and efficiency the users performed better while using the hybrid application, which means that the hypothesis is false in this case.

The hypothesis for task 3 was that the native application has a better usability when the user solves vertical orientation problems and hierarchical problems. In terms of effectiveness the users performed better in the native application than in the hybrid one, however the efficiency was lower on the hybrid version which makes the hypothesis false.

The hypothesis for task 4 was that the native application has a better usability when the user solves vertical orientation problems. In terms of effectiveness, the average completeness rate was higher on native but the average number of errors were also higher in that framework, giving a mixed result. In terms of efficiency the results are also mixed, with a lower average time of completion on the native framework but also with a higher...
average number of clicks. This means that the hypothesis is false.

When studying the overall usability in terms of effectiveness, the total completeness was higher on hybrid and the total number of errors were also lower in that framework, thus making the effectiveness overall better on hybrid. Regarding the efficiency, the average time and the average number of clicks for all the tasks were also lower on hybrid than on native.

The difference in efficiency and effectiveness is small, and if the validity of the data is considered as well, it looks like there is no difference between the frameworks in terms of quantitative measurements. However, if there is any difference seen it is that the hybrid application is slightly better in terms of quantitative performance measures.

The measurements of effectiveness and efficiency points in the same direction as the satisfaction questionnaire in many ways. The users sees the hybrid application as easier to use, but on the other hand the users says that the hybrid version is harder to use than the native, which is contradictory with the results of the hybrid being easier to use. The results between those statements is however very small and might not indicate a difference that affects the satisfaction between the frameworks. The native application makes the user more confident and is more consistent. The result is hard to draw any conclusion from, since it is so diverse. Thus, the satisfaction between the frameworks are close to the same.

In both online articles [9] [8] and in previous research it is mentioned that the native application has a better performance than hybrid applications. For instance Saccomani says that the native applications are faster [9], however when the user interacts with the devices this studies results points at the opposite direction since the hybrid application had a better average time but also the overall performance of the users were worse on the native application. The applications can thus be technically faster, but when the user interacts with the application the performance seems to be equal or worse according to the results from these tests.

This means that the results from the objective measurements of the usability points towards a results where there are no differences between the frameworks, both in terms of effectiveness and efficiency but also in terms of satisfaction. If we look at the small differences that can be noted the results points towards higher performance, in terms of effectiveness and efficiency, in the hybrid application and a satisfaction that are equal between the frameworks. If a user takes longer time, makes more errors, clicks and does not reach the intended goal it is easy to assume that the satisfaction would be lower on the native application. However, since the results are contradictory, with a worse performance and an equivalent satisfaction on native, there might be other reasons for the users satisfaction rating.

To further investigate the differences between the frameworks, qualitative research were also conducted and the results from those are presented below.

8.2 User Experience

Even though it was not possible to see any greater differences in the quantitative measurements, it was easier to draw a conclusion from the qualitative measurements. The results show that most of the users favored the native application and there were several reasons for it; nicer finish, better animations, better flow and a higher trust are some of which are mentioned several times.

The native framework takes advantage of built in features which can create a more familiar look and feel [10], which might be one of the reasons to why users preferred the native application and talked about it with words such as ”better finish” and that talked about the hybrid application with words such as ”It does not feel as modern in some way”.

Regarding the feeling of using an application Saccomani mentioned that the native application takes advantage of built in features,
such as the swipe gesture, which would act more familiar to the user compared to a hybrid version [9]. The familiarity is of high importance both according to Saccomani, but also according to others [9] [8] [6] [40][5] [42] [43] [44], who points out that the users of a platform often feels a strong connection to that brand. The preference for something familiar is also seen in this study where the users mentions that the vertical scroll is much slower than the hybrid one, yet it made it possible to read the text and felt more good and familiar to them. The animation was also often referred to as more familiar to the users on the native version and also something that they liked more than the one in the native version.

How fast the applications responds to the users actions is also an important part of the user experience and should also be better in the native version according to Ingmars [39] since there is a built in delay on web and hybrid application that are not included in a native version. The delay was also mentioned by a few users who says that the native application "reacted faster" and that the hybrid version was slower, which was a negative thing "especially in today’s society when everything is supposed to go fast". The users also had a fear that the application itself would crash while using the hybrid application, indicating that there is some sort of mistrust to the quality of the application.

There were however a few cases where the native version was less liked, and most of it was due to a behavior in the application that was not familiar to the users. The two things mentioned were that the keyboard covered the input fields for the users when they were supposed to writing contact information in their bookings and the other thing that frustrated users was that the booking button did not seem to respond when they pushed it, which was due to the fact that the keyboard was clicked away the first time and the second time the button responded to its intended action. However, even though these were seen by many users most users still preferred the native application. The reason for this seems to be aligned with what the previous research points towards; a familiarity with the interaction types that corresponds to the user’s experiences and mental images about how an application should look and feel like [9] [8] [6] [40] [5] [42] [43] [44].

8.3 Contributing Factors

The hypothesis in this study was that the native framework is better in terms of usability and user experience than the hybrid framework, however the results shows that the quantitative measurements are equal between the frameworks while the user experience is in favor for the native framework.

A reason for this might be that the quantitative method was not able to capture all aspects of the usability, or that some of the aspects are much harder to capture in a more standardized usability test. For instance, many users mentioned that the familiarity of how the application looked and felt was important and also made them like the native application more. This is not something that can be measured in either clicks, time, errors or completeness and might not have been captured by the satisfaction questionnaire either.

The result from the quantitative study is not aligned with what current industry knowledge and research has found, which can be due to several reasons. First of all, this might be due to the ceiling effect that influenced the results on the completeness measurements. Due to that effect, it was not possible to see if there were a difference between the frameworks and all the data about the effectiveness could thus not be collected. Secondly, more users are needed to get a more statistically significant result in the usability tests and thus make it easier to talk about if there really is a difference between the frameworks. How the test is done can also influence the users performance. For instance, the tasks could be long and hard to read and some users had a hard time understanding when they finished the tasks, something that caused some users
to get a much higher finishing time than they could have had if they had understand the task from the beginning more correctly. Besides time, this could also affect other performance metrics such as clicks, errors and completion.

Besides more users it would also be of interest to test users in the missing age groups, in this case people between 30-40 years old and 60+ since that also might have given a better understanding of a larger population. In this study most users were in their 20’s, which means that many of them have used technology for a long time and a lot, something that might influences their perception about how an application should look like and behave. This might be different for people of other age groups, however this is nothing that is possible to conclude in this study but is however important to mention as an contributing factor to the studies result.

Another factor that influenced the usability and user experience of the hybrid and native frameworks is how they are programmed and which libraries that are used. In this study, the goal has been to look at the basic libraries and components that are included in the native or hybrid framework and thus look at what the core frameworks has to offer and thus compare them. However, some libraries are used and those has presented some differences between the applications. For instance the calendars used are two different libraries designed for each framework, but they are also visually and interaction wise designed differently, and was thus something that the users noticed as a difference. This influenced some user’s liking of the applications in favor for the native application just due to the fact that it had a European format instead of an American one. Another difference was the carousel which also were based on different libraries for the different frameworks, this made them behave and look differently, also in favor for the native framework. This also points towards the fact that the chosen libraries are better designed for the native application than the hybrid, but on the other hand there are a lot of different implementations of this and thus the way of implementing the feature might influence the user’s perception more than what the framework actually does.

Many other small differences in the frameworks are also possible to avoid with a little bit more programing and another type of implementation, however this might also mean that just by putting more time into the implementation a hybrid application can come really close to the look and feel of the native one and thus the differences might not be noticed at all. This would mean that there are no difference between the frameworks, but as mentioned earlier the goal is to look at the standard components in each framework and see what is already implemented and solved and thus follows the standard for each framework and compare those.

Due to all these contributing factors and since the validation of the results from the usability tests are not statistically significant it is not possible to conclude that the results of this study can show whether or not the hypothesis of this study is true or false, however it is possible to see tendencies in the results. Those tendencies points towards a usability that is very much the same on both application and a user experience that favors the native application, just as previous research suggests.

9 Conclusion

The hypothesis in this study was that the native version is better in terms of usability and user experience than the hybrid version. This was based upon previous research and current industry knowledge, however when tested the hypothesis was not proven to be true in all aspects.

The quantitative measurements of the usability showed no significant difference between the frameworks while the qualitative measurements of user experience pointed towards a strong preference for the native framework compared to the hybrid frame-
work. The reason for this diverse result is unknown since a result that showed a more equal liking of the frameworks should be expected when the usability metrics were so equal.

Thus, there is a subjective, but not an objective, difference between the hybrid and the native framework.

10 Future Work

To further explore the similarities and differences in usability and user experience between the hybrid and native frameworks there are a lot of studies that can be made. For instance it is possible to test the differences on several different programming languages. In this study React.js and React Native was used, however there are many other options out there to choose from which can influence the results as well. As mentioned earlier more test focusing on the usability part and quantitative information gathered is also of interest and could be explored more with more users testing the differences.

References


REFERENCES


REFERENCES
