JÖNKÖPING UNIVERSITY
School of Engineering

Designing Better Mobile Apps: An Experimental Evaluation of Apple's and Google's Design Guidelines

How analysing the Human Interface Guidelines for iOS and Material Design for Android better our understanding of the usability challenges app users face and what we can do to overcome key issues.

Main Subject area:	Informatics
Specialisation in:	User Experience Design & Information Architectures
Author(s):	Tom Gülenman
JÖNKÖPING	2022, June

Certificate of Completion

This final thesis has been carried out at the School of Engineering at Jönköping University within Informatics. The author is responsible for the presented opinions, conclusions, and results.

Examiner:	Bruce Ferwerda
Supervisor:	Jasmin Jakupovic
Scope:	30hp
Date:	15.06.2022

Attestation of Authorship

I hereby declare that this submission is my own work. To the best of my knowledge and belief, it contains no material published or written by another person – except where explicitly listed in the References and properly cited. Nor does it contain any material of mine that, to a substantial extent, has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Tom Gülenman

Abstract

When developing mobile apps, multiple factors must be considered when choosing between native or cross-platform technologies. The latter offers deployment of one codebase to multiple operating systems, such as Android and iOS. However, we argue that common design techniques lack an understanding of specific needs that separate iOS and Android users. This work presents an experimental approach using UI prototypes and existing native iOS and Android mobile applications to identify issues in usability of the two systems. We conduct a large amount of usability tests involving 34 participants and find that our prototypes and apps that follow Apple's Human Interface Guidelines are easier to use in terms of one primary usability metric, notably task time, with a statistically significant difference between iOS and Android testers in two out of four tests. On this basis we investigate what key UI elements and design patterns cause disruptions in otherwise smooth User Experiences. Alongside documenting those, we also list key elements that influence the usability on a more general level. We provide suggestions to app designers and developers on avoiding designs that are considered producing usability issues by at least one of the two groups of iOS and Android users and on design patterns to improve the User Experience.
Keywords— Mobile Applications, Cross-platform, Native, iOS, Android, User Experience, Usability

Contents

List of Figures

List of Tables

1 Introduction

1.1 Problem Statement

1.1.1 Background

One of the central questions to answer in the app development industry is whether to employ native or cross-platform technologies. Native apps refer to applications that employ different codebases for different operating systems (OS), e.g., Swift for iOS applications and Kotlin for Android applications. Cross-platform technologies, on the other hand, are used to publish apps to multiple platforms with the same codebase. This essentially translates to only needing to implement an app once and adding a few configurations on top, while native apps have to be programmed once for each platform. Popular cross-platform technologies are the Javascript framework React Native and the Dart framework Flutter. Native technologies have been shown to be more performant generally but also require more resources in the development stage. Additionally, cross-platform technologies have been on the rise for a few years now and continue to improve rapidly.

1.1.2 Problem Description

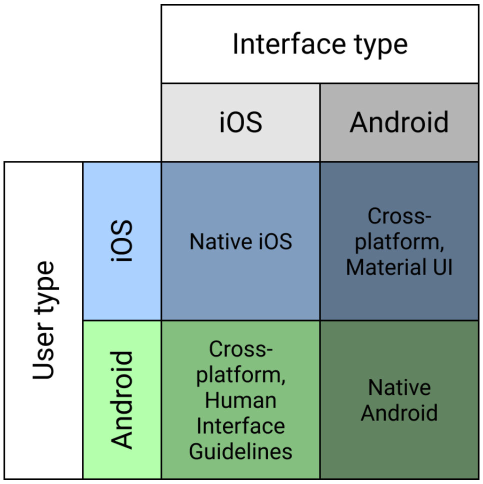
Mobile apps typically follow different design guidelines depending on the platform they are deployed to. Google published the Material Design guidelines (MDG) (Google, 2022) for helping designers and app developers make their Android apps feel more integrated into the operating system, while Apple's counterpart are the Human Interface Guidelines (HIG) (Inc., 2022). Let us examine the example of the popular Javascript framework React Native that allows running apps on iOS and Android with one codebase. React Native comes with native components that will change their appearance based on the OS. However, this list of components can not account for the entirety of the respective design guidelines. For this reason and because their employment allows faster implementations, UI toolkits are popular within the React Native community. However, those toolkits are mostly based on the MDG: e.g. the most popular toolkit, React Native Elements, with over 21800 Github stars as of January 2022. This might represent usability issues for iOS users who are used to Apple's design system. On the other hand, many use cases exist for apps to be developed to resemble native iOS apps, in which case they might look unfamiliar to Android users.

Cross-platform apps can be looked upon from two different perspectives depending on the user type: iOS or Android. The UIs of cross-platform apps make compromises in their design to be available on multiple platforms with the least amount of OS-specific code and appeal to both user types simultaneously. Apps may present UIs that are neither similar to Apple's nor to Google's design guidelines, but, in case they do, the UIs might be unfamiliar to one of the two user types, resulting in usability issues.

1.2 Purpose and Research Questions

This section describes the purpose of the study and how three research questions that will be answered in this work emerge from it.

We previously stated that cross-platform apps, due to their "one codebase for multiple platforms"-nature, have to compromise their designs in terms of OS-specific functions and styling. Focusing on iOS and Android, this study's purpose is in a first step to investigate how these compromises affect iOS and Android users in terms of usability issues and their perceptions thereof. In a second step, we investigate these usability issues and make out key UI elements at the root of the problem to give recommendations on which ways of employing UI elements and design patterns to avoid and how to better existing solutions in cross-platform apps. Lastly, we identify insights on users' needs and other key elements that positively or negatively affect the UX on a more general level.



		Interface type	
		iOS	Android
User type	iOS	Native iOS	Cross-platform, Material UI
	Android	Cross-platform, Human Interface Guidelines	Native Android

Figure 1.1: Matrix showing the user and interface types as well as resulting app categories.

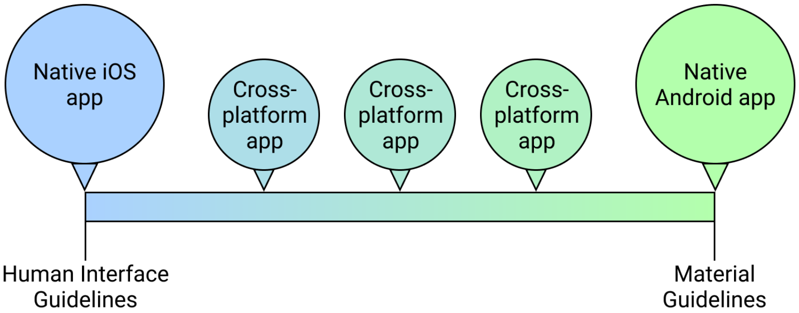


Figure 1.2: Spectrum showing the degree to which apps follow the HIG and the MDG.

Apple's and Google's design guidelines are not opposites and do not represent the only dimension determining a certain technology's employment or the users' perception of an app's UX. In addition, many apps operate on the basis of their company's branding that might not be fully compatible with either guideline or do not follow any guidelines besides their own at all. Despite the mentioned limitations, the two design guidelines are the dimension of interest that determines the user's familiarity with mobile app UIs. So far, we have looked at them as a binary choice. The guidelines, or UI types, and user types can, up to this point, be represented by a 2x2 matrix, as shown in figure 1.1. However, mixing the two guidelines can make for apps on a more continuous spectrum when it comes to which guidelines they follow, as depicted in figure 1.2. This theoretical spectrum is intended to visualise the flexibility of cross-platform apps on which set of guidelines they are built on. To simplify matters, it represents the best case scenarios of native apps following the respective guidelines to 100%. Once more, this is an abstraction of reality. Following a set of guidelines to 100% and another one to 0% is virtually impossible as they do not fully oppose each other in all matters. It is equally worth mentioning that cross-platform apps could focus exclusively on one set of guidelines. Ultimately, this does not justify the need for an additional UI type to be considered in our work: We operate on the assumption that the most unfamiliar UIs a user of a specific OS can operate on are those that are built upon the respective adversary guidelines, i.e. an iOS user will be most unfamiliar with an app that presents an interface that is built upon Google's guidelines. The same is true for an Android user and Apple's guidelines. The point of maximal unfamiliarity is of the highest priority to this research as it represents the worst case in terms of usability. For this reason, we argue that the investigation of native iOS and Android apps and prototypes designed to follow Apple's or Google's design guidelines represent the UIs that meet our comparative research needs. Having established that two different types of UIs are subject of this research, the first goal is defined by the first research question:

Research Question 1	What differences in app usability exist when users are presented with two different types of UIs: ones that follow the design guidelines for iOS and ones that follow the design guidelines for Android?

Further clues to bettering the usability of cross-platform apps can be given by prioritising the factors that make the investigated interfaces seem unfamiliar. This will help designers and developers make more accurate decisions on what parts of the design guidelines should be followed when developing cross-platform apps. It is worth mentioning that this might be valuable for the use case of native apps as well, where designs might be heavily reliant on company branding or largely unified for both OS for other reasons, such as time and development cost. In these cases, a strict approach to uniquely following one set of design guidelines might not always be realistic. On the other hand, it translates into profiting from insights into the key factors that produce usability issues most frequently in unfamiliar UIs. The second research question is:

Research Question 2	What are key factors of unfamiliar UIs that produce usability issues most frequently?

Lastly, we expect to gain a better understanding of mobile app design that is not necessarily attributable to a specific set of guidelines, but concerns more general design and usage patterns. As a consequence, we formulate the third research question:

Research Question 3	What general insights about mobile app design are to be gained from testing the design guidelines?

Answering these research questions gives us new insights on the decision-making process of the technology to base a mobile app on and what key UI and design guideline factors to focus on from a UI type point of view.

1.3 Scope

In a first step, this study entails the identification of a research gap, notably how the UX of interfaces built upon Apple's design guidelines compares to the UX of interfaces built upon Google's, but also how the UX of cross-platform mobile apps compares to the UX of native mobile apps on a UI level. For this purpose, we argue for the usage analysis of different mobile app UIs, focusing on the usability discrepancies between interfaces based on Apple's HIG and those based on Google's MDG. We argue for the usage of an experiment in pursuit of answering the formulated research questions.

The starting point for conducting the experiment builds an analysis of ordinary tasks in the mobile app environment and the re-creation of those, once following Apple's design guidelines and once Google's as Figma based UI prototypes, and by utilising existing native apps from both systems. Furthermore, user testings are conducted where iOS and Android users complete a series of tasks. At the same time, we measure multiple performance metrics to evaluate the usability of the prototypes as well as native apps by Apple and Google. As the last step before analysing the collected data, post-test interviews are conducted to gain additional insights into the users' perceptions of the UX of the prototypes used in the experiment. The results are presented and discussed to answer the research questions with a focus on the comparison of the usability of iOS and Android-based UIs and the identification of key factors that obstruct a good UX.

Lastly, the collection and analysis of data are critically evaluated to conclude the methods and results' success and relevance, and further research opportunities are mentioned.

1.4 Outline

The sections of this report are organised as follows: the preparatory phase prior to carrying out the experiment is described by going into detail about the chosen methods and why they are valuable to this study in 2.1, how the prototypes and apps have been prepared in 2.2, and why the selected data collection techniques are relevant procedures to answer the research questions in 2.3. The experiment and post-test interviews are then described before examining the employed data analysis techniques in 2.4 and reviewing their validity as well as reliability in 2.5. In 3.1, an overview of state-of-the-art literature is given, and theories of relevance, as well as employed technologies and tools, are described in 3.2. The results of the experiment and post-test interviews are presented and analysed in 4 before discussing them in 5. Final conclusions are drawn in 6.1 before exploring future research suggestions in 6.2.

2 Method and Implementation

Having motivated the need to evaluate usability discrepancies between iOS and Android UIs, we conduct a usability study in the form of an experiment where users carry out a series of tasks on different Figma-based UI prototypes and native iOS and Android apps. This section provides an overview of our method choices and the design guidelines before going into detail about the artefact creation process and data collection as well as analysis concepts. Finally, we argue for this work's validity and reliability.

2.1 Method Choice

In the experiment part of our research methods, we conduct a series of tests on our UI prototypes and apps with iOS and Android users as participants. We need to understand what usability problems arise when users are confronted with unfamiliar UIs. This need is best satisfied by the experiment research method: the UI elements are our variables, and we set up hypotheses prior to the experiment, such as "There is no difference in terms of task time between Android and iOS testers". The execution of the experiment will either confirm or falsify our hypotheses and might confirm links between the mentioned variables, following the applicability description of (Saunders et al., 2016). Conducting experiments, and more specifically user-testings, is the optimal choice because it is crucial to collect data, that is at the same time plenty and thorough, about the interactions with the UIs. Participants are not supposed to think about their choices for a long time before we can observe their interaction or lack thereof, which is why a survey would not be a good fit, to name an example. Using surveys would also signify the existence of an additional layer of abstraction since participants would not be able to use any UI but would only be able to think about their usage and then answer the related questions about it. For the same reason, only conducting interviews would not be satisfactory. However, post-test interviews are considered helpful in gaining additional insights into the thought processes of participants during the tests.

Since we heavily focus on UI elements to test the usability of apps, a realistic approach to creating the necessary prototypes for the experiment is a combination of Figma based high-fidelity designs and native apps. Comparing this approach to developing native apps, the experiment stage is largely facilitated. Another angle to consider is the exclusive usage of existing apps, including those not developed by Apple or Google, instead of creating our own prototypes. This would allow for the most natural UX but also bear several disadvantages regarding the proposed research questions. Popular apps are often heavily influenced by the distributing company's branding. Our goal is to reveal the effects of Google's and Apple's design guidelines on the usability of interfaces. We suspect this to be easier if those effects are not mixed with the effects of a company's branding and design guidelines. This way, we exclude additional potential distractions in the UIs, thus facilitating the filtering out which factors result in changes to the usability of UIs.

2.2 Artefact Creation

2.2.1 Preparations

Real-life conditions are essential to this study's relevance, which is why the experiment tasks are extracted from a selection of the most popular apps throughout different app categories. A list of those apps alongside their category in the App Store is shown in table 2.1. Relying on the functionalities of these mentioned apps, the in this study created prototypes are exclusively vertical UIs that require multilevel approaches to solving tasks.

Choosing the perhaps most common use cases of these apps is not an easy task. Precise app usage statistics are an often well-kept secret of the publishing companies. Therefore, we chose the following approach to assess everyday use cases for the in table 2.1 mentioned apps: We received support from the designers and engineers of Papaja AB, a UX Design and App Development Company based in Jönköping,

App name	Category	Chart Placement App Store
WhatsApp Messenger	Social Networking	1
Instagram	Photo & Video	1
Google Maps	Navigation	1
Spotify	Music	1

Table 2.1: Selection of popular apps to base prototypes on. The categories are taken from the App Store and the chart placement is based on the popularity of the app within the category on the App Store.

Sweden. Papaja provided us with initial user stories they deem typical for the mentioned apps. Being experts in the field, their educated guess is more likely to estimate what kind of tasks users regularly work on and what tasks are interesting for us to further research in the scope of this study.

The creation of the prototypes does not follow a by-element approach. We are not choosing specific parts or elements of the design guidelines on which we want our research to focus. However, instead, we analyse everyday tasks the users deal with in popular apps and recreate those with the tools the different guidelines provide. This approach is based on the real-life needs of the users instead of the structures of the design guidelines. Furthermore, this allows for better comparisons between both versions of each prototype. We are not worried about choosing design guidelines elements that are similar enough between Google's and Apple's versions to make for fair comparisons. Instead, we know that the comparisons have real-life applications, as the developed user stories are not OS-dependent. On the other hand, assuring that the two versions are different enough to discover usability discrepancies can not be guaranteed, but finding that they are non-existent or relatively minimal would be a valuable insight nonetheless.

Further on, this section provides an overview of the user testing scenarios and brief documentations of key decisions in the creation of prototypes and examples of finalised screens.

2.2.2 User testing scenarios

Four user stories serve as starting points to base the prototypes on. In cases where native apps are employed instead of prototypes, the user stories are relevant in helping us formulate tasks for the users to complete during the experiment. In terms of prototypes, it is then specified which functions are needed to allow the completion of the described tasks. Especially helpful are native iOS and Android apps by Apple and Google that offer similar functions to what is needed, e.g. Apple's Messages and Google's Messages app as a template for redesigning WhatsApp.

The finalised user testing scenarios that emerge from the initial user stories and an iterative designing and reflection process are the following:

1. WhatsApp: You want to start a group chat to plan a friend's, Laura's, surprise party this weekend. You realise that it concerns the same friend group that was present at Greg's birthday party last year. You remember being part of a quite old group chat that had a name probably including the words "birthday" and/or "surprise". You want to check who was in that group, then create a new group including the same people, with the difference of including Greg this time instead of Laura.
2. Google Maps: You want to find the quickest way to the restaurant where you're meeting a friend in a couple of minutes. You don't remember the exact name of the restaurant, but it has something to do with Sushi and you have already searched for it recently. You want to check if public transport or walking is faster. You also want to see if you come by the station "Jönköping Rådhusparken" so that your friend might join you on the way. Finally you want to switch to satellite view to get a better impression of where exactly the restaurant is.
3. Spotify: You got reminded of a song. You don't remember the name of the song or band but you remember it being in your first playlist ever created. You recall that the album cover features a lot of red color and that the album must have been released in 2003. You want to find that song in your playlist and make sure it's actually from 2003.

4. Instagram: You just downloaded Instagram after a time of inactivity. You want to check a number of things out to get up to date. Find out about:
    - The amount of followers you have
    - Who of your followed contacts have created posts in the last 12 hours
    - What the comments say on the first post in your feed
    - Recent activity on your profile in terms of likes and follow requests
    - What your current profile description is
    - Whether you still have that picture of your old dog on your profile
    - Whether your friend Amy (nickname: amy_muller) is still often posting stories
    - (As your last task) how you can create a new post

2.2.3 Prototypes and apps

This section describes the final prototypes and apps used during the experiments. Examples of screens are given, and the entirety of prototype screens can be inspected in the appendices 8.1 and 8.2.

Our WhatsApp prototypes blend the original app and Apple's and Google's Messages apps. Google Maps is directly tested for the Android version, and Apple Maps is taken as its iOS counterpart. Spotify has two native alternatives that are being tested directly: Apple Music and YouTube Music. Lastly, Instagram is fully redesigned to follow the HIG in one and MDG in the other case.

WhatsApp

The WhatsApp prototypes are primarily based on Apple's and Google's Messages apps. The final interfaces must offer the same possibilities as WhatsApp itself, but in a different design, including that functions might be differently grouped and accessible in other ways and on other screens. An example of this is the screens that allow for inspection and manipulation of group chat details. Ways of accessing these screens and the presentation of information and settings are done differently depending on the OS. Interactions are similarly dependent on the OS, e.g. revealing the full names of group participants requires an extra tap on iOS. As often, there are also multiple ways to reach the goal of completing the given scenarios. For example, in this case, the testers are looking for a specific group chat and might use the scroll function or the searchbar. Using the searchbar can lead to faster results, but its level of accessibility, especially to participants that are used to another OS, varies between the two systems. Finalised screens of this prototype are shown in figure 2.1.

Google Maps

Google Maps is considered a native Android app that also has been ported to iOS. Apple's Maps, however, is the native iOS alternative and includes similar main functions making both apps eligible to be included in our experiment. Changes to the original UIs are not necessary in this case. A lot of the relevant differences between the two apps are based on styling as well as how information is accessed. Switching between transportation methods in the apps is a good example, as Google Maps lists them in a scroll view which might seem like hiding additional information at first compared to Apple Maps' approach of offering buttons at the top of the corresponding card view to switch between transportation methods. This example can be inspected in the Apple and Google Maps screenshots in figure 2.2.

Spotify

Spotify is a music streaming platform with the direct competitors of Apple Music and YouTube Music. Navigation and playlist management offer, in both cases, mainly the same functions but present different approaches to reach the goals set for the participants of the experiment. Once more, key differences between the apps are ways of accessing data. The example in figure 2.3 shows a menu that can be accessed from a song detail-view to find out more about the artist.

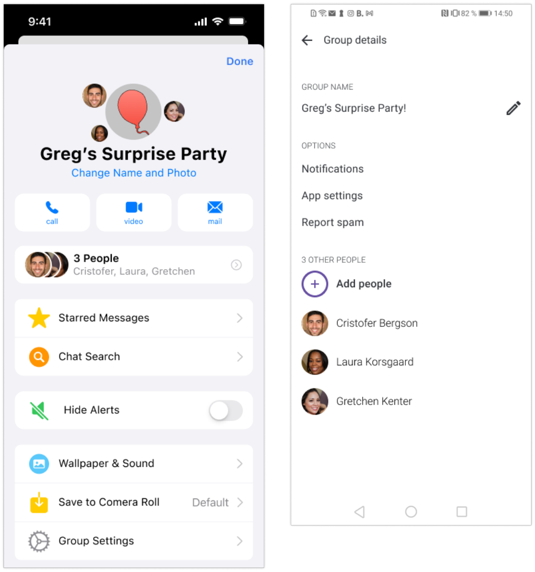


Figure 2.1: Screens of the WhatsApp iOS (left) and Android (right) prototypes.

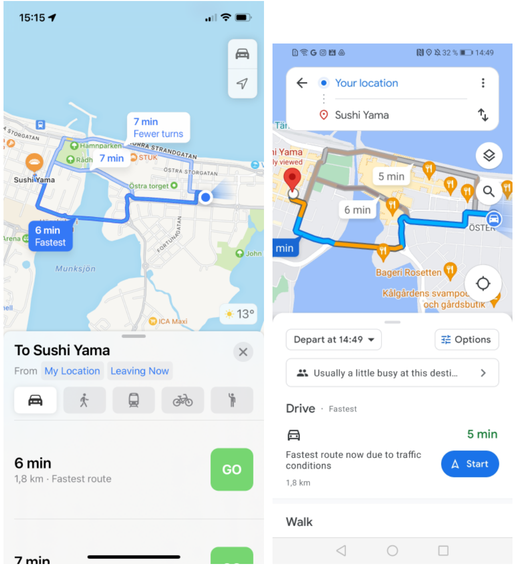


Figure 2.2: Screenshots of the Apple Maps (left) and Google Maps (right) apps.

Instagram

The Instagram app is almost identical on iOS and Android. Single native apps by Apple or Google that support us in nativising Instagram are non-existent. However, a range of native apps is being taken into consideration to create the prototypes. It is essential to thoroughly analyse the app's functions and use native component resources. It is easy to end up with UIs that resemble the original Instagram by a large margin and do not present enough diversity between the two interface variations. Figure 2.4 depicts how two UIs of the final prototypes differ when accessing a post detail-view to reveal comments.

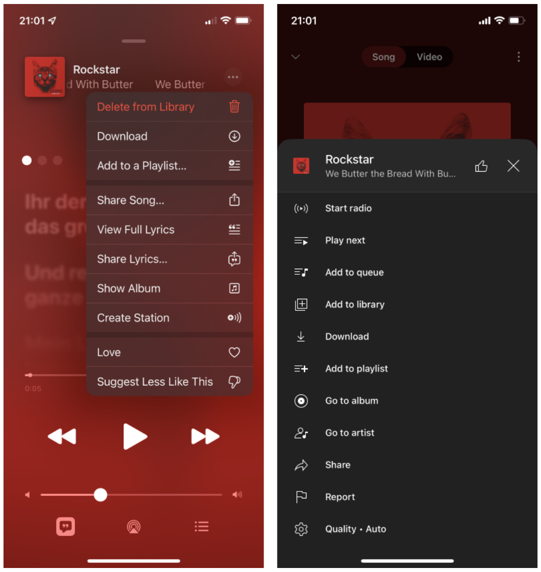
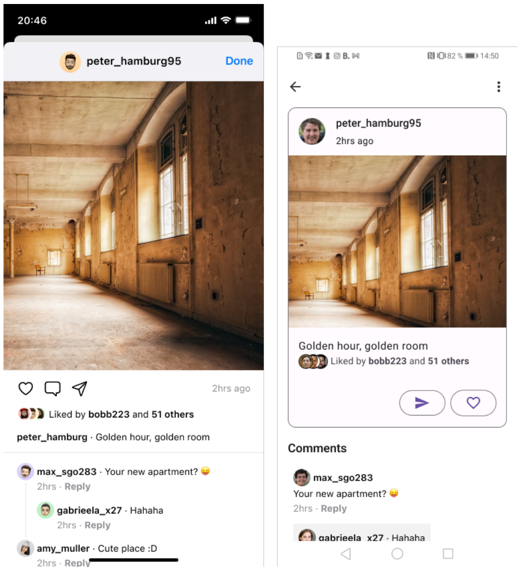


Figure 2.3: Screenshots of the Apple Music (left) and YouTube Music (right) apps.



Liked by bobb223 and 51 others

peter_hamburg · Golden hour, golden room

max_sgo283 · Your new apartment? 😜
2hrs · Reply

gabrieela_x27 · Hahaha
2hrs · Reply

amy_muller · Cute place :D
2hrs · Reply

Golden hour, golden room
Liked by bobb223 and 51 others

max_sgo283
Your new apartment? 😜
2hrs · Reply

gabrieela_x27 · Hahaha

Figure 2.4: Screens of the Instagram prototypes with iOS on the left and Android on the right.

2.3 Data Collection

This section gives an overview of how we collect data during the experiment and interview sections.

2.3.1 Participant Information

We recruited 34 participants, 16 female and 18 male, between 19 and 28 years old. Half of the participants are iOS users and have been for at least the past 3 years, while the other half have been Android users for at least the past 3 years. All of the participants are students and are familiar with the concepts of the researched apps, even if they do not use them regularly.

2.3.2 Experiment

Conducting the experiment, participants are given a series of tasks, and we measure usability metrics, including the time and interactions needed to complete the tasks. This allows us to assess the usability of the presented prototypes, which will ultimately build the foundation for answering the research questions. Test subjects interact with the UIs by using a given iOS and Android smartphone running the Figma Mirror app making the prototypes susceptible to inspection and interaction. The screen, as well as audio, are recorded during the entire test session.

2.3.3 Post-Test Interviews

Lastly, we conduct post-test interviews with the participants to reiterate the strengths and weaknesses of both types of UIs from their perspectives. The interviews are recorded. The interview questions can be inspected in the appendix 8.3.

2.4 Data Analysis

Having collected the experiment results and post-test interview answers, we analyse the data to answer the in 1.2 mentioned research questions.

Our analysis focuses on the performance of the two different user types during the experiment and their observations that are communicated in the post-test interviews. Our objective is to determine what type of UIs is more usable and by which factors this is determined. Various quantitative statistical measures are employed to compare the performances of user types. For the experiment and interviews, we are tracking metrics based on the usability attributes that (Saleh et al., 2017) present as part of their "Mobile Application Usability Evaluation Metrics" model. Key attributes to collect information about are efficiency, effectiveness, and simplicity. As to what kind of performance measurements are important in answering the research questions, we focus on a selection of usability metrics that (Saleh et al., 2017) list as responsible for evaluating the aforementioned usability attributes. This selection is shown in table 2.2. The criteria for selecting metrics were that they should be measurable through the course of the experiment or the interviews in the frame of this study. Since answering our research questions relies on determining if two groups of testers possess different characteristics in using apps, we compare the means of the in table 2.2 mentioned metrics. We test our results for normality with Shapiro-Wilk test and proceed with either Student's t-test or Mann-Whitney U tests depending on the normality outcome. Results of each pair of prototypes are tested to determine statistically significant differences between iOS and Android testers.

Usability attribute	Usability metric
Efficiency	Task time
Effectiveness	Number of navigational steps
Effectiveness	Number of errors
Simplicity	Number of touches to finish task

Table 2.2: Usability attributes and metrics taken into consideration when evaluating the usability of mobile apps.

Many measures to determine the participants' performances are also qualitative: determining if users managed to solve problems optimally with multiple possible solutions and understanding why their choices turned out as they did is crucial to revealing which UI elements have the greatest impact on the usability.

2.5 Validity and Reliability

This section gives an overview of the measures that have been taken to maximise the internal, external, and construct validity, as well as the reliability of the in the scope of this research obtained insights.

Choosing a between-subject design for the experiment structure, as later described in 3.3, contributes to establishing causal links between the operations of interfaces during our user testings as well as the identification of usability impediments on both UI types by eliminating carry-over effects, thus contributing to higher internal validity than a within-subject design. Basing the prototypes on some of the most popular mobile apps and testing some of the most popular native apps is a choice that creates experiments at least close, and possibly identical, to real-life app usage scenarios of the testers. This enhances the external validity of this study compared to building prototypes that resemble lesser-known app usage processes. The post-test interviews are essential to our efforts to provide sufficient construct validity. Speaking to the participants gives us insights on things that are harder to measure, such as shorter periods of confusion or understanding the reasons behind decisions the test subjects make during the experiment. All of these types of validity can be improved in future work by adding to this work and possibly enhancing the scale of experiments: especially having a higher number of UIs, as quite a high number of testers has been achieved within this research, further adding to the validity and reliability of this study.

A qualitative analysis of the experiment recordings and the post-test interviews make for higher reliability of the results that could, similarly to the validity, be further enhanced by a larger scale experiment.

3 Theoretical Framework

This chapter describes the state-of-the-art literature about a multitude of comparisons of Android and iOS, as well as native and cross-platform technologies, while also further specifying the research gap that this report is tackling. An overview of employed technologies for the preparation and execution of experiments is given before further detail on the experiment structure.

3.1 Literature Overview

This section gives an overview of existing literature and identifies the research gaps this work attempts to fill. Researchers have recognised the importance of investigating the many differences between Android and iOS over the years, and the debate about iOS versus Android extends to numerous fields of research. Those range from purely technical aspects over concerns for privacy and security to usability issues.

(Garg & Baliyan, 2021) explore differences between iOS and Android in terms of a wide array of security aspects. Results of their work show that iOS is the generally safer OS. A study on privacy is conducted by (Kollnig et al., 2022), resulting in the findings of alarming privacy concerns on both systems. When looking into recent research that relates to issues in UX, we notice works that are focused on specific apps, such as WhatsApp, as in the case of (Mkpojiogu et al., 2020) who explore differences in UX between recent iPhone and Android devices. The authors found that the Android app was easier to learn for first time users. (Caro-Alvaro et al., 2018) investigate the related topic of usability issues in instant messaging apps on a broader scale and list a series of key issues, as well as present recommendations aimed at developers of messaging apps. However, state-of-the-art literature lacks comparative studies between iOS and Android that concern usability issues on a UI level and a broader scale involving different types of apps.

Additionally, since we argued that the topic of native versus cross-platform technologies is a related subject, it is only plausible to provide an overview of the literature comparing cross-platform and native technologies, where researchers have undertaken a multitude of approaches. (Biørn-Hansen et al., 2019) conduct an online survey to analyse opinions about cross-platform technologies from developers' perspectives and describe the awareness of performance and UX penalties when employing cross-platform technologies. (Ma et al., 2018) research the differences in performance native app and web-app approaches provide, mainly on a network and energy consumption level. Two different types of apps have been developed by (Nawrocki et al., 2021) who measure various performance metrics for identical apps in native and cross-platform contexts. (Masner et al., 2015) focus on comparing the two different approaches from an economic point of view while (Pinto & Coutinho, 2018) use the economic advantages, amongst other things, to motivate the usage of hybrid technologies as a solid alternative to native ones. (Shah et al., 2019) even go as far as saying that "the use of cross-platform tools usually outweighs the disadvantages that come with it" while recognising the compromised UX. However, for this related subject, state-of-the-art research lacks approaches that focus on UI components in their comparisons, which is the gap our work is attempting to fill.

3.2 Overview of the Employed Technologies and Tools

The basis for the in this study conducted experiment are, besides native apps, our UI prototypes. These are screens displayed on mobile devices to simulate the usage of mobile apps without involving program code. These apps are non-functional or almost non-functional with only a minimal set of interactions possible, e.g. tapping a button might take the user to another screen, simulating the usage of a mobile app; however, all screens are pre-built and wired, without actual processing of any kind of data. Creating these testable UI prototypes is done with Figma (Figma, 2022), an interface design and prototyping tool. Testing those prototypes on mobile is enabled by the Figma mobile apps for iOS and Android that allow real-time interactions with the in Figma created UIs. The testing equipment consists of an iPhone 12 and a HUAWEI P smart 2019 that allow for screen and audio recording during the testing and the post-test interviews.

3.3 Experiment Structure

Android users test four iOS UIs consisting of two prototypes and two existing native apps by Apple, while iOS users test four Android UIs consisting of two prototypes and two existing native apps by Google.

Participants are given a sheet of paper containing the instructions in written form.

4 Results

After introducing the necessary definitions, this section presents notable findings of the experiments, interviews, and statistical analyses. Furthermore, these findings are analysed by means of interpretation and unifying insights from the experiments and post-test interviews.

With 4 tests done by each of the 34 testers and 4 key metrics tracked for 18 different tasks, we count 2448 baseline data points to analyse. Our focus is to make statements about more general structures of performances, UI elements, and app design patterns that matter in the context of this study and specifically to answer the research questions. We mainly use box plots to visualise the results and analysis, giving an overview of the minimum value, the first quartile, the median, the third quartile, the maximum value, and outliers with the addition of the standard deviation and t-test result.

4.1 Results Discussion Setup

This section introduces the definitions of usability metrics, gives an overview of the participants we recruited, the handling of testers struggling to complete tasks, and explains how we refer to testers to weight results of the experiments and interviews.

4.1.1 Usability Metrics Definitions

The in 2.2 specified usability metrics are used to assess the performance of testers during the experiments and, ultimately, the usability of the examined prototypes and apps. The following definitions are based on the work of (Saleh et al., 2017) and added upon to fit the context of our study.

- **Task time**: The duration to complete the task at hand in seconds.
- **Number of touches**: The number of screen touches participants make when attempting to complete the task at hand.
- **Number of navigational steps**: The number of accessed interfaces. This includes switching between pages but also major navigation patterns within a single page.
- **Number of errors**: The number of interactions that lead the tester further away from the current goal or that would lead the tester further away if they were functional in the case of prototypes. Repetitions of interactions are not counted, such as revisiting screens during the same task or that have been visited in a preparatory task. Subsequent errors are not counted.

4.1.2 Major Difficulties in Completing Tasks

During the experiments, testers sometimes get stuck on a problem. We propose to define "being stuck" as struggling with the completion of a given task until giving up on attempting to solve it without help. Calculating the time of a participant being stuck essentially translates to identifying two events: the point in time when a user asks for help and the point in time when a user has made progress towards the current goal for the last time. It is essential to include the collection of new information in this construct of "making progress". More often than not, the first point in time where a tester is not making progress anymore is when an error is made. However, calculating the time users are stuck for requires careful evaluation of the preceding steps a tester has made, their gestures during the testings, and their verbal comments.

One problem with users needing assistance in completing tasks is that some might be more reluctant than others to ask for help. To even out the playing field, we calculate the average time of being stuck and set it as an upper limit to the task time metric. In addition, every touch, navigational step, and error that occurs after the upper task time limit is reached is ignored. However, we are also applying a lower limit for the metrics of touches and navigational steps equal to the minimal number of steps required to reach the goal from the point where participants start to get stuck.

Out of 136 tests, users got stuck 32 times for an average duration of 54 seconds. Each time a user is stuck for more than 54 seconds on a task, the difference between the time they are stuck and 54 seconds is deducted from their task time.

4.1.3 Data Cleaning

Other than adapting the various metrics in scenarios where users have major difficulties in completing tasks as described in 4.1.2 no cleaning of data has been done. No additional procedures were needed for dealing with outliers or faulty data.

4.2 Quantitative Experiment Outcomes

This section presents the quantitative experiment outcomes. For simplification purposes, prototypes and apps are referred to as prototypes.

Of the 34 participants, 0 participants were excluded during the process. They completed 4 prototypes each, making for 136 prototype testings in total of which 68 (50%) were iOS testings and 68 (50%) were Android testings.

To assess the usability of iOS and Android UIs, we analyse 4 metrics for both tester groups of iOS and Android testers: task time, number of touches, navigational steps, and errors. For each participant these usability metrics are calculated for all 4 tests. The individual results are then aggregated into a group average for each metric, depending on whether the participant is in the iOS or Android tester group. The results are summarised over all tasks of a prototype for each metric. Appendix 8.4 presents box plots of the the results in terms of descriptive statistics.

To assess whether statistically significant differences between the two tester groups with regard to the usability metrics are given, we conduct Student's t-tests or Mann-Whitney U tests, depending on the outcome of Shapiro-Wilk tests for normality, comparing the results of both groups for each metric and prototype.

4.2.1 Task Time

On each prototype individually, iOS testers were faster on average than the Android testers as shown in table 4.1.

Prototype	iOS testers		Android testers	
	M	*SD*	*M*	*SD*
WhatsApp	114.1	41.8	163.5	49.0
Google Maps	77.5	36.3	109.5	38.5
Spotify	123.8	77.6	172.6	76.8
Instagram	128.2	26.6	152.8	49.2

Table 4.1: Results of the task time metric in seconds over the individual prototypes for iOS and Android testers.

Shapiro-Wilk tests on each dataset indicate that the results are normally distributed for both tester groups only in the case of WhatsApp, as can be seen in table 4.2. We therefore rely on non-parametric tests for the other three prototype pairs.

The difference of 49.4 seconds in task time on WhatsApp prototypes is statistically significant, $t(32) = 3.17, p = .003363$.

Mann-Whitney U tests on the remaining pairs of prototypes indicate that the respective differences are statistically significant for the Google Maps prototypes, as can be seen in table 4.3.

Over all prototypes combined, the iOS tester group was faster (s) on average ($M = 110.9$, $SD = 52.4$) than the Android tester group ($M = 149.6$, $SD = 59.2$).

Prototype	iOS testers			Android testers		
	df	statistic	*p*	*df*	statistic	*p*
WhatsApp	17	0.94	.2980	17	0.95	.4045
Google Maps	17	0.92	.1346	17	0.88	.0355
Spotify	17	0.88	.0369	17	0.95	.4662
Instagram	17	0.96	0.5738	17	0.87	.0216

Table 4.2: Results of the Shapiro-Wilk tests for normality on the results for task time showing degrees of freedom (*df*), Shapiro-Wilk test statistic (statistic) and p value (p).

Prototype	$N_{android}$	N_{ios}	$U(N_{android}, N_{ios})$	z	p
Google Maps	17	17	79	2.24	.0251
Spotify	17	17	90	1.86	.06288
Instagram	17	17	111.5	1.12	.26272

Table 4.3: Results of the Mann-Whitney U tests on the task time metric over the individual prototypes.

4.2.2 Touches

On the WhatsApp and Instagram prototypes, iOS testers used less touches on average than the Android testers, but more during the testings of the Google Maps and Spotify prototypes as shown in table 4.4.

Prototype	iOS testers		Android testers	
	M	SD	M	SD
WhatsApp	27.9	9.0	31.1	9.2
Google Maps	29.0	12.6	27.9	14.6
Spotify	48.5	29.9	47.6	27.8
Instagram	43.8	12.0	46.2	12.2

Table 4.4: Results of the touches metric in seconds over the individual prototypes for iOS and Android testers.

Shapiro-Wilk tests on each dataset indicate that the results are normally distributed for both tester groups only in the case of Instagram, as can be seen in table 4.5. We therefore rely on non-parametric tests for the other three prototype pairs.

Prototype	iOS testers			Android testers		
	df	statistic	*p*	*df*	statistic	*p*
WhatsApp	17	0.88	.0377	17	0.97	.7538
Google Maps	17	0.90	.0679	17	0.85	.0122
Spotify	17	0.87	.0218	17	0.88	.0334
Instagram	17	0.97	0.7962	17	0.95	.3917

Table 4.5: Results of the Shapiro-Wilk tests for normality on the results for touches showing degrees of freedom (*df*), Shapiro-Wilk test statistic (statistic) and p value (p).

There is no statistically significant difference for Instagram prototypes, $t(32) = 0.58, p = .566323$.

Mann-Whitney U tests on the other pairs of prototypes indicate that the remaining differences between tester groups are not statistically significant, as can be seen in table 4.6.

Over all prototypes combined, the iOS tester group needed less touches on average ($M = 37.3$, $SD = 19.7$) than the Android testers ($M = 38.2$, $SD = 19.2$).

Prototype	$N_{android}$	N_{ios}	$U(N_{android}, N_{ios})$	z	p
WhatsApp	17	17	112	1.10	.27134
Google Maps	17	17	128.5	-0.53	.59612
Spotify	17	17	141.5	0.09	.92828

Table 4.6: Results of the Mann-Whitney U tests on the touches metric over the individual prototypes.

4.2.3 Navigational Steps

On each prototype individually, iOS testers used less navigational steps than Android testers as shown in table 4.7. Note however, that for the WhatsApp and Spotify prototypes, the minimum amount of required navigational steps to complete the prototype differs, which would make for an unreliable comparison. We perform tests on the remaining Google Maps and Instagram prototypes.

Prototype	iOS testers		Android testers		Minimum required steps	
	M	SD	M	SD	iOS	Android
WhatsApp	11.5	4.7	15.2	5.5	5	6
Google Maps	6.2	2.0	7.2	3.4	4	4
Spotify	9.8	4.8	10.9	5.7	5	4
Instagram	13.5	2.2	14.9	4.2	6	6

Table 4.7: Results of the navigational steps metric in seconds over the individual prototypes for iOS and Android testers. Also showing the minimum number of navigational steps needed to require each prototype for each OS.

Shapiro-Wilk tests on each dataset indicate that the results are normally distributed for both tester groups only in the case of Instagram, as can be seen in table 4.8. We therefore rely on a non-parametric test for the Google Maps prototypes.

Prototype	iOS testers			Android testers		
	df	statistic	p	*df*	statistic	p
Google Maps	17	0.90	.0741	17	0.86	.0157
Instagram	17	0.89	0.0557	17	0.97	.885

Table 4.8: Results of the Shapiro-Wilk tests for normality on the results for navigational steps showing degrees of freedom (*df*), Shapiro-Wilk test statistic (statistic) and p value (p).

There is no statistically significant difference for Instagram prototypes, $t(32) = 1.24, p = .224905$.

A Mann-Whitney U test on Google Maps indicates that the difference between tester groups is not statistically significant $U(N_{android} = 17, N_{ios} = 17) = 127.5, z = 0.57, p = .56868$.

Over all prototypes combined, the iOS tester group needed less navigational steps on average ($M =$ 10.3, $SD = 4.4$) than the Android testers ($M = 12.1$, $SD = 5.7$).

4.2.4 Errors

On each prototype individually, iOS testers committed less errors than Android testers as shown in table 4.9.

Shapiro-Wilk tests on each dataset indicate that the results are not normally distributed for both tester groups in any case, as can be seen in table 4.10. We therefore rely on non-parametric tests for all prototype pairs.

Mann-Whitney U tests on each pair of prototypes indicate that these differences are not statistically significant, as can be seen in table 4.11.

Prototype	iOS testers		Android testers	
	M	SD	M	SD
WhatsApp	2.8	1.4	3.4	1.1
Google Maps	1.4	1.1	2.5	2.1
Spotify	1.6	1.4	2.4	2.0
Instagram	2.9	2.2	4.1	3.0

Table 4.9: Results of the error metric in seconds over the individual prototypes for iOS and Android testers.

Prototype	iOS testers			Android testers		
	df	statistic	p	*df*	statistic	p
WhatsApp	17	0.95	.4452	17	0.88	.0333
Google Maps	17	0.88	.0322	17	0.91	.0937
Spotify	17	0.82	.0044	17	0.91	.0853
Instagram	17	0.92	0.1730	17	0.88	.0281

Table 4.10: Results of the Shapiro-Wilk tests for normality on the results for errors showing degrees of freedom (*df*), Shapiro-Wilk test statistic (statistic) and p value (p).

Prototype	$N_{android}$	N_{ios}	$U(N_{android}, N_{ios})$	z	p
WhatsApp	17	17	109	1.21	.22628
Google Maps	17	17	103	1.41	.15854
Spotify	17	17	117	0.93	.35238
Instagram	17	17	111	1.14	.25428

Table 4.11: Results of the Mann-Whitney U tests on the error metric over the individual prototypes.

Over all prototypes, the iOS tester group committed less errors on average ($M = 2.2$, $SD = 1.7$) than the Android testers ($M = 3.1$, $SD = 1.7$).

4.3 Qualitative Experiment and Interview Outcomes

This section provides an overview of the qualitative learnings from the experiment, such as comments by the participants and obvious fixations on certain UI elements, and of the answers participants gave during the semi-structured post-test interviews. During the interview, participants were given the chance to look through the prototypes. The interview questions can be inspected in the appendix 8.3. The outcomes are categorised and presented in a OS and prototype specific manner.

4.3.1 Analysing the Qualitative Data

As previously mentioned, qualitative data has multiple origins. The information we collect during the experiments are verbal comments of the testers, gestures, and ways of interacting with the UIs. During the interviews we collect the participants' answers. These different types of data are then unified into outcomes by identifying how the according user performed on a given task or sub-task, or how they perceived the usability of the UI, and what UI elements are involved and the reason for the observation, if applicable. An example for this would be a user wanting to create a new group on WhatsApp and commenting that they don't think the floating action button is the right one to create a new message. The associated outcome would then be: "Floating Action Button (FAB): Had trouble associating the "new message" function with it". We extract possible outcomes of all mentioned data, group them by prototypes in a first step, and rank them by the amount of testers the outcome applies to. We are then able to tell what the biggest issues or more generally insights were in using the UIs and which UI elements were involved. Before being further put into context in 4.4, these outcomes are presented.

4.3.2 Referring to Testers

We executed a considerable amount of user tests with 17 iOS users and 17 Android users. However, these numbers are not as high as to argue based on differences of a couple of testers only. Combined with an attempt to make the documentation of experiment outcomes and analysis more intuitive and easily readable, we refer to testers in 5 ways to describe the number of testers involved. Table 4.12 shows the number of testers that the different expressions refer to.

Expression	**Number of testers**
None	0
Few	1-5
Several	6-11
Many	12-16
All	17

Table 4.12: Referring to testers.

4.3.3 Android Prototypes

17 iOS users have been asked to tell us about their experiences with our WhatsApp and Instagram prototypes, as well as the native apps of Google Maps and YouTube Music. The following tables present the qualitative data we gathered during the testings of the prototypes and the according interview sessions. Those are table 4.13 for the WhatsApp prototype, table 4.14 for the native Google Maps Android app, table 4.15 for the native YouTube Music Android app, and table 4.16 for the Instagram prototype. Answers and insights are categorised and ordered by the number of participants that the learnings originated from.

WhatsApp

Outcome	**Amount of testers**
Create group button: Had trouble finding it/Thought the new message screen was already enough for making a group	Many
Floating Action Button (FAB): Had trouble associating the "new message" function with it	Several
Overall design: Feels messy / unclean / bad / not very appealing / cheap	Several
Group chat: Expected to be able to tap the group name	Several
FAB: Used to button on the top right for creating groups	Several
Searchbar: Searchbar on top takes up too much space	Several
FAB: Looks like "single chat starter"	Few
FAB: Like it, including fixed positioning	Few
Searchbar: It is "ugly"	Few
Tabs: Menu at the top seems useful	Few
Bottom Navigation: Labels seem unnecessary	Few
Navigation: Would like to go back to the main screen faster (from group screen)	Few
Group chat: Likes different colours for chat bubbles from different people	Few
Overall design: Don't like the font	Few
Main page: Names should be standing out more	Few
Main page/Chat rows: Lines should separate different chats	Few
Overall design: Don't like how the colours play together	Few
Overall design: Would want to be able to choose different colours	Few

Table 4.13: Qualitative learnings from the WhatsApp Android prototype experiment and post-test interview answers categorised and ordered by the number of participants that the learnings originated from.

Google Maps

Outcome	**Amount of testers**
Filter buttons (Chips): They are convenient	Many
Navigation types: Having to scroll down to change is bad	Many
Overall design: Looks similar to Google Maps on iOS	Several
Overall design: Simple to use	Several
Start page: Too cluttered	Several
Cards: Card dimension are bad (too big) / screen is halved when tapping a restaurant / goes up all the way when swiping / can barely see map when navigating somewhere	Several
Overall design: Like the additional functions compared to Apple Maps	Few
FAB (Start navigation): Button is too big	Few
Overall design: Harder to use than Apple Maps	Few
Overall design: Would like a combination of Apple Maps' design and Google Maps' functions	Few
Overall usage: Started using Google Maps a long time ago, which is why they are not switching to Apple Maps	Few
Overall design: the app is showing too much information	Few
Navigation types: Would be better to always show all transportation types instead of hiding unavailable ones	Few
Search suggestions: It's nice to see if a restaurant is closed in the search already	Few

Table 4.14: Qualitative learnings from the Google Maps Android native app experiment and post-test interview answers categorised and ordered by the number of participants that the learnings originated from.

YouTube Music

Outcome	**Amount of testers**
Playlist sorting: It's very hidden	Several
Overall design: Font is bad / It's too big	Few
Overall design: UI feels cheap / Outdated / Not appealing / Not clean enough	Few
Filter buttons (Chips): They are convenient	Few
Overall design: Feels simple / Easy to navigate	Few
Play bar: Could be prettier	Few
Recommendations/App structure: Get the feeling that they show me what they want me to see, not what I want to see	Few
Overall design: Quite clear to use	Few
Overall design: Feels similar to Spotify	Few
Song menu: Feels overfull	Few
Random song button: Just gets bold when toggled - too hard to know if it's activated	Few
Playlist sorting: Its function is confusing (recently added = last playlist added or last time a song was added?)	Few
Album information: Would like more information	Few
Overall design: Sliding cards up from the bottom is awkward	Few
Next songs: Not able to toggle "up next" like on Apple Music	Few

Table 4.15: Qualitative learnings from the YouTube Music Android native app experiment and post-test interview answers categorised and ordered by the number of participants that the learnings originated from.

Instagram

Outcome	**Amount of testers**
Tabs: Like favourites and close friends tabs / Like tab for reels / Would love to have those in the actual Instagram / Better than the chevron icon on Instagram	Many
Feed: Don't like the margin / Pictures should be bigger	Several
Cards in feed: Like those / They are compact / It's easier to process everything	Several
Overall design: UI elements feel too big (fonts, icons, cards)	Few
FAB: It's in the way	Few
Overall design: UI feels cheap	Few
Buttons on cards: They should be redesigned / Are bad	Few
Buttons on cards: They are well grouped / Like them as opposed to a messy Instagram	Few
Buttons on cards: Share should be on the right / Share more than they like - should be more easily accessible	Few
FAB: Confused about it / Didn't know what it was for / Couldn't find it	Few
App bars top: Taking too much space / Don't like it	Few
Settings: Should be in profile	Few
Profile post filters (Chips): They are ugly	Few
Overall design: Colours are bad	Few
Comment button: Don't like it / Feels unnatural to interact with	Few
Bottom Navigation: Don't like it / Icons are ugly / Labels are unnecessary	Few
Bottom Navigation: Activity tab is nicer here than top right in Instagram	Few
Bottom Navigation: Like messages tab here	Few
Activity screen looks good	Few
Feed: Like that posts are in chronological order as opposed to actual Instagram	Few
Cards in feed: Don't like the border	Few
Overall design: The whole app is very easy to navigate	Few
Overall design: Seems quite similar to Instagram	Few
Feed: Dislike the term "Feed"	Few

Table 4.16: Qualitative learnings from the Instagram Android prototype experiment and post-test interview answers categorised and ordered by the number of participants that the learnings originated from.

4.3.4 iOS Prototypes

In equivalent manner to the iOS users that tested Android prototypes, 17 Android users have been asked to tell us about their experiences with our iOS prototypes and apps. Those include Apple Maps instead of Google Maps and Apple Music instead of YouTube Music. Our findings are presented in the following tables: table 4.17 for the WhatsApp prototype, table 4.18 for the native Apple Maps iOS app, table 4.19 for the native Apple Music Android app, and table 4.20 for the Instagram prototype.

WhatsApp

Outcome	**Amount of testers**
Searchbar: Trouble finding the initially hidden searchbar	Many
New message: It just feels like a single message, not for creating a new group	Several
Overall design: Liked it / Seems clean	Several
Overall design: The app is easy to navigate	Few
Group chats: Looks clean / neat	Few
Overall design: UI seems similar to WhatsApp	Few
Creating a group: Want to create a group without having to send a message	Few
Settings: It's nice how accessible they are	Few
Bottom navigation: Would prefer these as tabs on top	Few
Overall design: Everything was OK	Few
Overall design: Colour theme is too bright	Few
Creating a group: Would like suggestions of people to add to group	Few

Table 4.17: Qualitative learnings from the WhatsApp iOS prototype experiment and post-test interview answers categorised and ordered by the number of participants that the learnings originated from.

Apple Maps

Outcome	**Amount of testers**
Terrain type button: Confused by change in icons depending on transportation method	Many
Choosing a route: It's confusing to have the car icon for starting a route that includes all transportation methods	Many
Overall design: Easy to use / Easy to navigate the app / Love the app / Feels smooth	Several
Overall design: UI looks great / Clean / Tidy	Several
Overall design: Google Maps feels more advanced	Several
Overall design: Similar to Google Maps	Several
Overall design: Prefer Google Maps because I'm used to it	Few
Overall design: Less confusing than Google Maps	Few
Transportation methods: Easier to change those than on Google Maps	Few
Main screen: Having the searchbar further down is nice	Few
Card views: Like to have the information from card views and see the map at the same time	Few
Overall design: Like the colours	Few
Overall design: The colours are too pale	Few
Filters: Is missing the filters (chips) under searchbar like in Google Maps	Few
Choosing a route: Like to have map and routing included at the same time and not having to go back and forth	Few
Searchbar: Should be bigger	Few
Search: Like that it's just three suggestions at the start and not "all in your face"	Few

Table 4.18: Qualitative learnings from the Apple Maps iOS native app experiment and post-test interview answers categorised and ordered by the number of participants that the learnings originated from.

Apple Music

Outcome	**Amount of testers**
Overall design: UI looks clean	Several
Overall design: Seems similar to Spotify	Several
Song options: Dislike the too many options	Several
Playlist sorting: Trouble recognising the sorting button	Several
Cards: Like how big the pictures are	Few
Playlists: Would like more information - how many songs are in there, when it was created	Few
Overall design: App feels very simple	Few
Overall design / Bottom Navigation: Compared to Spotify this is easier to use / Especially the navigation bar with 5 options	Few
Overall design: UI is too bright	Few
Song options: Like the many options	Few
Filtering: Missing the filtering chips like on Spotify	Few

Table 4.19: Qualitative learnings from the Apple Music iOS native app experiment and post-test interview answers categorised and ordered by the number of participants that the learnings originated from.

Instagram

Outcome	**Amount of testers**
Searchbar: Trouble finding the initially hidden searchbar	Many
Discover tab: Would prefer it as a tab on the bottom	Several
Profile: Accessing it is nice	Several
Profile description: Should be directly underneath profile name	Several
Overall design: Like how clean it looks	Several
Overall design: The app is simple to navigate	Several
Profile: The functions offered there are good	Several
Posts in feed: Margins on the side are not nice	Few
Overall design: Don't like memojis	Few
Profile: Didn't like how it was made - don't want things on top of my own posts	Few
Posts in feed: Like to have them in chronological ordered as opposed to the real Instagram	Few
Posts in feed: Would like to see the first comment as well	Few
Posts in feed: Like to only have people I'm following here	Few
Navigation: Would like quicker access to activity	Few
Posts in feed: Like the card designs	Few
Creating a post: It's easy	Few
Messages: Easily accessible	Few
Overall design: The app is easy to understand	Few
Overall design: Using this app was a pain	Few

Table 4.20: Qualitative learnings from the Instagram iOS prototype experiment and post-test interview answers categorised and ordered by the number of participants that the learnings originated from.

4.4 General Analysis and Research Questions

In this section, we analyse the findings of the experiments and interviews. We establish links between the previously presented quantitative and qualitative results as well as statistical analyses to answer the in 1.2 defined research questions and fulfil the purpose of this study.

4.4.1 Differences in App Usability between iOS and Android

The purpose of this study is to provide insights on usability discrepancies between UIs that are built on Apple's design guidelines and UIs that are built on Google's. As for our primary goal, we therefore set out to answer the first research question of "What differences in app usability exist when users are presented with two different types of UIs: ones that follow the design guidelines for iOS and ones that follow the design guidelines for Android?".

What Are the Most Impacted Metrics?

The most notable result is that tasks are performed substantially faster on iOS prototypes than Android prototypes, which is true for each prototype. With almost 40 seconds less in average prototype completion speed, iOS testers clearly had an easier time performing the given task under the aspect of time. In the cases of WhatsApp and Google Maps prototypes a statistically significant difference between the two tester groups concerning the task time has been confirmed suggesting that an analysis of factors impacting the usability is especially valuable for these UIs. No other differences between tester groups could be labeled statistically significant for any metrics.

Our metrics bear a certain degree of interdependence: Committing more errors results in longer task times, as do more navigational steps. A high number of navigational steps often result in higher errors because navigating to a screen can be considered erroneous. Therefore, it is no surprise that, given the longer task times, Android users committed more errors than iOS users, with an average of 3.1 to 2.2.

We observed different types of users concerning their touch behaviour. Some users had a very low frequency of touching the screen and seemed to think a lot before each action. Others were touching the screen a lot more frequently, including taps that essentially did not serve any purpose, e.g. navigating around the map in the Maps testing scenarios. Different user types were present on both sides, and the average results are close to each other, with Android testers touching the screen on average 38.2 times to finish all prototype tasks, while iOS testers needed 37.3. No pattern of reproducing a similar minor difference is apparent, with Android testers needing more touches on WhatsApp and Instagram but less on Spotify and Maps.

How Did the Testers Perceive the Prototypes?

Android users that turned into iOS testers for our purposes were generally much more open to the appearance of the prototypes. For each iOS prototype, several testers described the UIs as clean or tidy and mentioned that they liked it that way. Several described different prototypes as easy to navigate, easy to use, easy to understand, having a smooth feel to it, or even that they loved the app in the case of Apple Maps. A few testers referred to Apple Maps as less confusing than Google Maps. A few got different impressions and mentioned that the overall design of prototypes was OK, while another few found that the Instagram prototype was a pain to use. The participants showed different views concerning the colour themes, with a few liking the overall colours of specific prototypes, while others thought of the same as too pale or too bright. It was also interesting to see several or few, depending on the prototype, express how the prototypes were very similar to their Android counterparts.

iOS users who played the role of Android testers had a more challenging time especially adapting to the visuals of the prototypes they were given. Several testers described the UIs as messy, unclean, bad, not very appealing, or cheap. A few of them criticised cluttered screens and information overload as a result, as well as not liking the overall colours and how they play together throughout different prototypes. A few participants also criticised the fonts and UI elements in general as not pleasant or too big. For the Spotify and Instagram prototypes, a few testers mentioned that the app was easy to use or easy to navigate. However, several said the same about Google Maps, which might originate from many Android testers using Google Maps in their daily lives. Several also proclaimed that the app was similar to the iOS version of Google Maps. At the same time, only a few compared YouTube Music to Spotify or the Instagram prototype to the actual Instagram.

Concluding Research Question 1

34 testers have demonstrated in 136 testings of prototypes that the most impacted metric from our selection of usability metrics is the task time. Answering our first research question, we observed mobile app users perform day-to-day tasks substantially faster on prototypes and apps that UIs constitute following the design guidelines for iOS than on ones that follow the design guidelines for Android. Statistically significant differences between the two tester groups have been found in 2 out of 4 cases. The experiments and post-test interviews also suggest that users generally find UIs following Apple's guidelines more visually pleasing and more often easier to use and navigate. At the same time, UIs built upon Google's guidelines are critiqued more often by mobile app users who are not used to them, most commonly characterising them as messy and cheap looking. Even though both tester groups make out similarities to their counterparts, Android UIs are less often described as easy to use, but in the case of Google Maps, recognised as offering more functions.

4.4.2 Identifying Key Factors That Produce Usability Issues

Having identified differences in app usability between apps following Apple's and those following Google's design guidelines, research question 2 asks "What are key factors of unfamiliar UIs that produce usability issues most frequently?". Our observations during the experiments and explanations provided by the participants during the interviews permit us to answer this research question: We describe how OS-specific elements are responsible for usability issues before listing them in a summary.

Key Elements on Android

Android testers' most significant issue was understanding and interacting with floating action buttons (FAB). Those troubles ranged from having issues associating the correct function with it and being used to its function being offered in another way, for example, through a button in top navigation bars, to it being considered too big, in the way of primary content, and simply confusing.

Another common issue was that when testers switched between transportation methods on Google Maps they had to scroll down through the list of available options. Apple Maps solved this issue by offering buttons to filter the transportation methods at the top of the menu, quite similar to what is often done with the tabs component on Android. Google's approach led to confusion and was criticised to a great extent. On the other hand, none of the iOS testers criticised Apple's approach; on the contrary, a few testers even recognised that it is easier to change transportation methods on Apple Maps. Our study's results suggest that a list of information in a scrollview should be easy to filter if there is a need for retrieving specific information that is distinguishable from other entries in more than one dimension.

When testers interacted with group chats during the WhatsApp experiment, they often tried to tap the group name to reveal more information. This is possible on the native iOS Messages app, but this prototype required them to use the dot menu in the top navigation menu. Users were able to figure out the correct way quickly, but the frequency of this event was high.

Searchbars were another component at the root of frequent discomfort for Android testers. Searchbars were perceived to be more permanent than in typical iOS apps. Searchbars on Android are often visible from the start, as they were in the WhatsApp prototype, and even though some top navigation bars are impacted by scrollviews and disappear upon scrolling down, they often reappear as soon as the scroll is taken to the other direction. iOS users are used to the searchbar not being visible from the start and thought of it as an initial waste of space. Testers also complained about its appearance, labelling it ugly.

When it comes to draggable views, testers mainly faced issues in Google Maps. Several participants described the several menus, such as restaurant and navigation detail views, as inconvenient in dimensions and responsiveness. After choosing a route, the according menu at the bottom of the screen takes up half of the screen, which was incredibly irritating to testers. Interacting with the menus often made them suddenly take up the whole screen or disappear or hide too much information. On the other hand, iOS testers appreciated seeing quite a lot of the map in the same situation. A few users mentioned that dragging these "cards" feels awkward on YouTube Music where they can reveal more information about the currently played song and a list of queued songs.

Testers were also presented with different card views throughout the experiment, such as albums and playlists on YouTube Music and posts in the Instagram feed. In the latter case, a few testers thought that the layout of the cards was bad and that the buttons specifically should be redesigned. The ordering of the buttons and the card dimensions and margins were bothering participants. Although a few testers also liked the approach and preferred it over Instagram's usual feed because they had an easier time keeping an overview of where posts start and end, they were in the minority.

The styling of listviews was also criticised by a few testers, who said that the chats rows on the main page of the WhatsApp prototype were a little challenging to differentiate from each other. They mentioned lines as means of separation as they were used to from many iOS list views, such as those used in the native iOS Messages app and names having to stand out more.

A more complex problem is represented by different types of navigations between the OS. Navigating back to the main screen from a group chat screen in the WhatsApp prototype takes an additional tap on Android. This is due to the behaviour of the back arrow button in the searchbar, which is based on one of the native Android Messages apps. A few users complained that there was an unnecessary step involved in going back to the main screen. Another problem related to different navigation systems occurred when testers tried to toggle the next up feature in YouTube Music, revealing which songs would be played next. It is common on iOS that pages remember more states of navigation so that when users return from another page, they can continue right where they left off. This is different on Android, where so-called forward navigations are reset after accessing another screen, referred to as lateral navigation. This is the same principle here where users have to repeatedly press the up next button instead of pressing it once on Apple Music, for example.

Key Elements on iOS

The most notable element throughout the iOS prototypes that test participants had trouble dealing with was the searchbar component in WhatsApp. Testers often struggled to reveal the initially hidden searchbar.

The terrain type button caused almost equally much confusion in Apple Maps. Depending on the currently selected transportation method, it changes its icon while maintaining the same function. In addition, most icons in question bear no resemblance to the icon used for the same button in Google Maps. On the other hand, when starting the navigation on Apple Maps, users press a button that always contains the icon of a car. This was similarly confusing to testers looking for other ways to start the navigation with other transportation methods.

Creating new groups in our WhatsApp prototype is done like in the native iOS Messages app. A new message button is pressed and people are added to the participants field before sending a message. This procedure was strange to several testers that felt like this function was reserved for new messages to single participants. Closely related were comments a few testers had about this: creating a group should be possible without sending a message right away was the feedback we got.

Still concerning WhatsApp, a few iOS testers mentioned that they would prefer the bottom navigation bar as tabs instead. They referred to the tabs component commonly used on Android, including the Android version of the actual WhatsApp.

A few testers missed the filtering chips that Google Maps offers on top, below their searchbar when using Apple Maps. Interestingly enough, this was one of the components iOS users appreciated the most about Google Maps.

A minor critique point was that a few testers did not like Apple's memojis as profile pictures on the Instagram prototype.

Concluding Research Question 2

In this section, the key factors that produced usability issues during the testing session most frequently have been described separately for Android and iOS.

To reiterate, summarise, and lastly answer research question 2: for Android, they are floating action buttons, how information is handled in scrollviews, page headers, searchbars, draggable views and menus, as well as their dimensions, the layouts of card views, the styling of listviews, and system-specific navigation patterns. For iOS, they are searchbars, buttons with changing icons, buttons with misleading icons, the to a large extent fusion of the processes for creating new personal and group messages, bottom navigation bars, missing filters, and usage of memojis.

4.4.3 General Insights on Users' Needs in Mobile Apps

So far, we have provided an overview of differences in app usability and which key factors are responsible for those on both OS. However, more insights are to be gained when evaluating the experiment results and interviews more generally about what users appreciate and need in mobile apps that are not specific to one of the two OS. We set out to answer the third and last research question: "What general insights about mobile app design are to be gained from testing the design guidelines?".

Merging Functions

When creating groups, the prototypes chose different approaches. Many Android testers were observed not finding the Create group button, while several iOS testers did not understand they could add more than one participant to the contact field. These results suggest that when creating usage flows that contain multiple similar functions, it is important to highlight when one function is incorporated into another, such as creating groups as a sub-function of the new message function on both systems.

Tabs Are Appreciated

Tabs components, which are commonly used on Android, made a good impression on iOS users, who were Android testers, during the experiment. From replacing traditional bottom navigation bars to adding another dimension of filtering or navigating between content, users had an easy time understanding its function and in few to many cases, depending on the prototype, even commented on liking it. This suggests that tabs are likely to be appreciated by many users of both OS.

Bars and Space

A few testers complained on different occasions about bottom navigation and top app bars and searchbars taking up too much space on Android UIs. Additionally, a few stated that the labels were unnecessary. Meanwhile, iOS testers suggested replacing bottom navigation with tabs altogether. This suggests that app designers should focus on keeping menus and navigational elements minimal to not distract from the primary information the screen is supposed to deliver or offer means of interaction to the user.

FABs and their Common Usage

Floating Action Buttons are widely used in cross-platform apps. However, our testings have shown that iOS users struggle with FABs quite heavily and disapprove of floating content that is responsible for covering other primary content. Designers need to be careful when assuming overall comfort in handling FABs.

Colours and Personal Preference

The in the experiments employed colour themes have been critiqued on both sides, but testers reacted positively to some factors, including the usage of colours to show differences in who creators of messages are in group chats. A few testers also mentioned that they would want to be able to choose different colours in different contexts. This suggests a desire for more personalisation in terms of colour themes within apps.

Suggestions, Recommendations, and Information Overload

Suggestions come in many forms: they can be typing suggestions or recommendations based on recent activities, to just name two examples. During the testings of Google Maps, participants complained about a cluttered interface and information overload. Testers of Apple Maps, on the other hand, liked a few aspects of the app that were more simplistic than their Google counterparts. The initial showing of only the three most recent searches is an example. Tapping a More button reveals additional recent searches. Google Maps presents a more extensive list of recent activity. Music streaming platforms work a lot with recommendations. A few testers complained about YouTube Music and that they feel like the app's structure is constructed to show them what YouTube Music wants them to see and not what they would want to see themselves. These few testers suggested that the app employed persuasive methods that bothered them. Another few users had the idea that WhatsApp should suggest who of their contacts to add to a group that is being created. These are three quite different examples of the testers' opinions about in-app suggestions and recommendations. Still, they show that participants are used to suggestions and recommendations in many forms and are open to those expanding, as long as they are handled responsively. Users should not feel like they are being persuaded, and don't overwhelm them in terms of quantity either.

On a related note, a few Android testers commented that they appreciate being able to see the opening times of restaurants in the search suggestions already. This suggests that designers might want to figure out the crucial information that users navigate to utilising searches and display a piece of this crucial information in the search suggestions.

In terms of information overload, designers need to be mindful that users of different OS seem to have different thresholds of what makes an interface cluttered.

Chips

The testings have shown that filter buttons in terms of chips components are appreciated when available and missed when not. A challenge of including these is to not end up with a cluttered interface that was criticised in the case of Google Maps.

Respect Existing Patterns

When many testers tried to tap the group names in the WhatsApp experiment, it became apparent that users assume gaining access to more information by interacting with the names of their contacts. This might be due to the degree of familiarity the testers have with WhatsApp and suggests that designers should value how popular apps function. It might be valuable to reproduce certain concepts such as clickable group names and perhaps the consistent possibility of interaction with users' names in general. Another example of this is how iOS testers told us that they prefer the discover function in Instagram as a tab in the bottom navigation bar, how the profile description should be right beneath the user name, and how the settings should be accessible through the profile.

Interesting was also to see that a few users liked the messages tab in the bottom navigation bar more than its original position on the main page of Instagram. This might require further testing but at the same time suggests that users care for consistency between apps: A messages tab in the bottom navigation bar is, for example, used by WhatsApp on iOS.

Sorting

When it comes to sorting lists and information, users struggled with figuring out that the playlists in both Apple and YouTube Music were sorted in a way that did not match their current goal and, in some cases, finding the sorting function in general. This suggests that apps need to make it more obvious how data is ordered and how to change the ordering to provide users with the means of efficiently processing information.

In addition, designers need to be mindful of the terms they use: "recently added" refers to when playlists have been created on Apple Music but refers to when songs have been added to playlists on YouTube Music. Choosing unambiguous labels is crucial to providing a smooth UX.

Bottom Navigation

The Apple Music experiment has shown that users prefer more tabs in the bottom navigation bars - 5 compared to 3 - offering them more efficient means of accessing the information they are looking for. Testers had mentioned that it gives them the impression that information within the app is better sorted this way.

Pictures and Grouping Content

Comments the testers made during the Instagram experiments suggest that the app's main focus, user-created pictures, should take up as much space as possible, at least in terms of width. Testers on both sides did generally not appreciate a card design that restrained the dimension of the pictures, added borders, or drop-shadows.

On the other hand, testers complimented the grouping of information on the cards. They mentioned that it is easier to process the offered information. This suggests that, even though pictures should not be restrained in width, there is a need for separating groups of information such as Instagram posts from each other.

Accessing Your Own Profile

iOS testers generally appreciated Apple's design for accessing their own profile. This suggests that, especially when lots of tabs are needed for content in the bottom navigation bar, the profile could be accessible through a button on the top, as iOS users are used to it from apps like the Apple App Store, and as Android users seem not to have any issues with.

Let Users Control Their Feed

The post-test interviews have shown that users want better control over what is shown in their feed. When developing feeds and thinking about what constitutes a pleasant UX, consider giving users control over what is shown and how it is sorted. Testers mentioned that they like the chronological order more than the order of posts currently offered by Instagram and that they would like only to see people they are following.

Concluding Research Question 3

The general insights we have gained about mobile app design from testing the design guidelines have been explained in this section. We answered the third research question by discussing key learnings related to multi-purpose functions, tabs, different types of bars, spacings, FABs, colours, suggestions and recommendations, information overload, chips and filters, existing design patterns, sorting, navigation menus, pictures, grouping content, user profiles, and feeds.

5 Discussion

5.1 Results Discussion

This study aimed to find out about OS-specific and general factors that impact the usability of mobile apps and investigate how these factors impact the users. We designed, conducted, and analysed the results of a usability testing experiment in order to understand what obstacles to a good UX exist with a focus on Android and iOS specific designing practices. We also identified key elements in the different types of UIs that produce usability issues. Our analysis has shown that those elements can be OS-specific and that specific general design approaches lead to changes in usability, for the better or worse. These insights build the foundation for the in our analysis described recommendations to mobile app designers and developers, focusing on cross-platform apps since this is where iOS and Android users face the most similar interfaces.

We tracked and analysed 4 usability metrics to quantitatively assess the performance of mobile app and prototype testers. As documented in our answer to research question 1 in 4.4.1, task time represented the most significant difference in terms of usability between our iOS and Android prototypes. Together with our post-test interviews, this allowed us to identify key elements that produce usability issues most often confidently. On this basis we were able to offer mobile app designers and developers an overview of which UI elements and underlying design patterns to pay special attention to with respect to the diverse needs of Android and iOS users. Documenting these key elements in 4.4.2 constitutes our answer to research question 2. However, it was not possible to classify all insights we gained through the experiment and post-tests interviews into OS-specific issues. We therefore set out to answer a third research question in 4.4.3 to share what we had learned in terms of what more general key elements were that impacted the UX of mobile apps. Our results include the documentation of usability issues but also of elements that improved the usability of our prototypes and tested apps from a user perspective, further providing designers and developers with recommendations on ways to impact the UX of mobile apps.

5.1.1 Platform-Specific Results

Going into the specific findings, the difference in time needed to complete the tasks between the two user groups was a surprise. We did not expect that much of a difference if any. This leads us to believe that certain key UI elements commonly found in Android apps substantially negatively impact the UX of cross-platform apps when utilised by iOS users. Those elements specifically are FABs and different kinds of views. Those views make their usability impact through their dimensions and layouts, as well as behaviour concerning resizing and filtering information. Another striking discovery was that the styling of the UIs in terms of colours, dimensions, fonts, font sizes, shapes, labels, and borders, as well as shadows, negatively stuck out to Android testers which was not the case the other way around at nearly the same intensity. However, it is hard to assess to what extent the more aesthetical side of designs accounts for in terms of usability issues, but it is valuable considering nonetheless.

iOS users were unsatisfied with the visual aspect of Android UIs, leaving us wondering if a potential bias towards other brands in iPhone users contributed to those findings. While it would be next to impossible to exclude such a potential bias in this study, we believe its effects to be limited: We observed Android testers' struggles and rarely found them to be without cause, even for their many comments on the styling of components. As an example, few testers mentioned they would like lines to separate the different chats on the WhatsApp main screen. They are used to such separators from the original WhatsApp but also the iOS Messages app. However they also mentioned that this would help them distinguish the contact buttons, making it easier to believe that their comment was sincere instead of base solely on prejudice towards Android.

Regarding navigation patterns, the feedback testers gave was, once more, one-sided. They criticised the Android prototypes and apps, but further research is needed to confirm our results: iOS users struggle with Android's mechanisms of resetting the states in forward navigation after a lateral navigation is performed.

As far as iOS experiments are concerned, we learned that iOS-styled searchbars are a major issue for Android users. When not revealed from the start, and even though few iOS users complained that a visible searchbar is a waste of space, apps involving Android users should always show the searchbar or icon from the start. Additional filter functionality was missed on multiple occasions, and we conclude that chips are an Android UI component that adds much value for Android users but that iOS users are also able to use intuitively without noticeable obstruction to a smooth UX.

With regard to related literature our results partly differentiate from other studies: In 2011 (Larysz et al., 2011) performed a study on UIs and usability issues of mobile apps and found that, in comparison with Windows Phone and Android, iOS provided the best UIs, based on consistency and maintenance throughout the system. Much later, (Kaya et al., 2019) found that, while investigating WhatsApp, Facebook, YouTube, and Mail apps on iOS and Android, there was "no significant difference between operating systems in terms of the usability of mobile applications". Similarly, Android and iOS are both identically characterised as very user friendly for lay users by (Adekotujo et al., 2020) in their comparative study of operating systems. In an industry that evolves at a very high speed it is important to continuously question such outcomes. Literature in comparative usability studies is scarce, especially on a UI level with focus on the design guidelines. While related work attributes a usability advantage to iOS in earlier stages, recent years seem to have balanced out user perceptions a little more. We however question an equal status in usability questions when it comes to iOS and Android and argue for the necessity of additional studies in the field to confirm what we have found: a superior usability of UIs built upon Apple's HIG with regards to task time.

5.1.2 General Results

We were surprised by how much we learned about UI elements impacting the usability of apps on a general level. We conclude that much value is to be gained from inspecting functions on native apps before implementing similar ones in cross-platform environments. Very interesting to see was that all types of users appreciate tabs to an extent where their usage seems reasonable in more than just Android apps. However, the most surprising finding was to what extent users wish to have better control over their apps. This control should come in the form of making users feel in charge of their app, translating to what type of content they are shown, letting them adjust colour themes, or, if possible, even adjusting the amount of information presented in interfaces. Users demonstrated the need for more intuitive sorting of information and more options to change content sizes of images or draggable menus and cards.

5.2 Method Discussion and Limitations

Our usability study is composed of an experiment and an interview section. The experiment section required preparations in terms of artefact creation of UI prototypes, amongst others. Recruiting 34 testers and conducting all experiments and interviews constitutes a major part of this study, allowing us to inspect many quantitative and qualitative data points. Having much data proved extremely useful in assessing the users' performances and analysing their perspectives and opinions about the usability of our prototypes. We are confident in the results of our study, having fulfilled the mentioned purpose and set out to provide incentives for further research, as the limiting factor to this work was time. We would have liked to construct and test more prototypes and apps offering a greater diversity in interfaces and UI elements to assess test participants' performances and potentially identify more usability enhancing or diminishing key UI elements and design patterns. We are equally confident that we, given our methods and scale of usability testing, achieve the requirements of validity and reliability.

Constructing and employing prototypes and apps requiring a different minimum of navigational steps to be completed proved disadvantageous. This essentially turned potential performance comparisons of half of the prototypes unreliable in terms of navigational steps. This should be handled with more care in future studies.

Employing lots of non-parametric tests in the statistical analysis as opposed to parametric tests is another limitation worth mentioning, as they are less efficient.

Assessing users' performances on only 4 native applications and 4 UI prototypes can be considered a limitation. Similarly, focusing on several apps that are the most popular in their respective app categories is a decision that might be worth rethinking the next time a similar study is conducted. We have seen that for many functions, that were rebuilt in our prototypes, the users were already heavily influenced from existing patterns of the original apps. A wider variety of apps, also in popularity, to involve in the experiment section might yield more diverse insights.

The participants in our experiments and interviews were all students aged 19-28, making it difficult to generalise our insights to larger remaining age groups that were not included.

Employing a survey to answer niche questions that came up during this research could have been valuable as well: Trying to understand if and how much iOS users possess prejudices towards Android phones is one example of what a survey might be used for.

6 Conclusions and Further Research

6.1 Conclusions

When designing apps, especially ones intended to be used by Android and iOS users, it is crucial to consider the functioning of native components of the two systems. A central challenge is understanding how they work and finding a compromise in merging different approaches into one that users of both systems easily understand. In this study, we found that UIs following Google's design guidelines generally require more time to complete various tasks on compared to UIs following Apple's design guidelines. We provide a documentation of key elements responsible for usability issues on both systems and in general and highlight factors that prove beneficial to the usability of mobile apps.

6.1.1 Practical Implications

Our documentation serves as a recommendation to app designers and developers on which elements to pay special attention to in order to create an inclusive and smooth UX for both Android and iOS users throughout their mobile apps.

6.1.2 Scientific Implications

Our study explores the research gap of comparative UI component research focusing on UX as an experimental study between iOS and Android. Further research may be conducted to build upon this study and analyse additional approaches to UI design and their impact on the UX of the users.

6.2 Further Research

We provided a variety of apps with 4 native apps and 4 prototypes that we conducted our experiments on. A larger variety of apps, including apps of different categories and popularity, might yield different insights on usability issues.

As our pool of participants only included students aged from 19 to 28, future work may include assessing performances of other age groups and users with different occupations.

Future work may include surveys designed to find out about usability issues. We would expect the results to be more superficial than choosing an experiment approach, but at the same time broader, with more participants being able to partake within a shorter amount of time.

Another technique that could be employed in future similar experiments is eye tracking. This would permit researchers to identify points of first fixation as well as the measure of dwell time in regions of interest. (Carter & Luke, 2020) describe the investigation of these metrics as the foundation for answering research questions like "What part of the face attracts attention first" and "What part of the face receives the most attention". It is up to future researchers to assess if the collection of eye tracking data is valuable in cases of interface user testings, where questions like "What part of the UI attracts attention first" and "What part of the UI receives the most attention" could be asked.

Lastly it would be interesting to know if and how results of a similar study would differ if focusing on tablets instead of smartphones and more generally how screen sizes affect the UX of apps with regard to similar usability metrics.

7 Bibliography

Adekotujo, A., Odumabo, A., Adedokun, A., & Aiyeniko, O. (2020). A comparative study of operating systems: Case of windows, unix, linux, mac, android and ios. *International Journal of Computer Applications*, *176*(39), 16–23.

Biørn-Hansen, A., Grønli, T.-M., Ghinea, G., & Alouneh, S. (2019). An empirical study of cross-platform mobile development in industry. *Wireless Communications and Mobile Computing*, *2019*.

Caro-Alvaro, S., Garcia-Lopez, E., Garcia-Cabot, A., De-Marcos, L., & Martinez-Herraiz, J.-J. (2018). Identifying usability issues in instant messaging apps on ios and android platforms. *Mobile Information Systems*, *2018*.

Carter, B. T., & Luke, S. G. (2020). Best practices in eye tracking research. *International Journal of Psychophysiology*, *155*, 49–62. https://doi.org/https://doi.org/10.1016/j.ijpsycho.2020.05.010

Figma, I. (2022). *Figma: The collaborative interface design tool.* https://www.figma.com/ (accessed: 30.03.2022)

Garg, S., & Baliyan, N. (2021). Comparative analysis of android and ios from security viewpoint. *Computer Science Review*, *40*, 100372. https://doi.org/https://doi.org/10.1016/j.cosrev.2021.100372

Google. (2022). *Material design 3.* https://m3.material.io/ (accessed: 30.03.2022)

Inc., A. (2022). *Human interface guidelines - design - apple developer.* https://developer.apple.com/design/human-interface-guidelines/ (accessed: 30.03.2022)

Kaya, A., Ozturk, R., & Altin Gumussoy, C. (2019). Usability measurement of mobile applications with system usability scale (sus). In F. Calisir, E. Cevikcan, & H. Camgoz Akdag (Eds.), *Industrial engineering in the big data era* (pp. 389–400). Springer International Publishing.

Kollnig, K., Shuba, A., Binns, R., Van Kleek, M., & Shadbolt, N. (2022). Are iphones really better for privacy? a comparative study of iOS and android apps. *Proc. Priv. Enhancing Technol.*, *2022*(2), 6–24.

Larysz, J., Němec, M., & Fasuga, R. (2011). User interfaces and usability issues form mobile applications. *International Conference on Digital Information Processing and Communications*, 29–43.

Ma, Y., Liu, X., Liu, Y., Liu, Y., & Huang, G. (2018). A tale of two fashions: An empirical study on the performance of native apps and web apps on android. *IEEE Transactions on Mobile Computing*, *17*(5), 990–1003. https://doi.org/10.1109/TMC.2017.2756633

Masner, J., Šimek, P., Jarolımek, J., & Hrbek, I. (2015). Mobile applications for agricultural online portals–cross-platform or native development. *Agris on-line Papers in Economics and Informatics*, *7*(665-2016-45057), 47–54.

Mkpojiogu, E. O., Kamal, F. M., Akusu, G. E., & Hussain, A. (2020). A comparative evaluation of the ux of whatsapp messenger on iphone x and samsung s9 plus mobile platforms. *International Journal*, *8*(10).

Nawrocki, P., Wrona, K., Marczak, M., & Sniezynski, B. (2021). A comparison of native and cross-platform frameworks for mobile applications. *Computer*, *54*(3), 18–27. https://doi.org/10.1109/MC.2020.2983893

Pinto, C. M., & Coutinho, C. (2018). From native to cross-platform hybrid development. *2018 International Conference on Intelligent Systems (IS)*, 669–676. https://doi.org/10.1109/IS.2018.8710545

Saleh, A., Ismail, R., & Fabil, N. (2017). Evaluating usability for mobile application: A mauem approach. *Proceedings of the 2017 International Conference on Software and E-Business*, 71–77. https://doi.org/10.1145/3178212.3178232

Saunders, M., Lewis, P., & Thornhill, A. (2016). Research methods for business students vol. 7th.

Shah, K., Sinha, H., & Mishra, P. (2019). Analysis of cross-platform mobile app development tools. *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, 1–7. https://doi.org/10.1109/I2CT45611.2019.9033872

8 Appendices

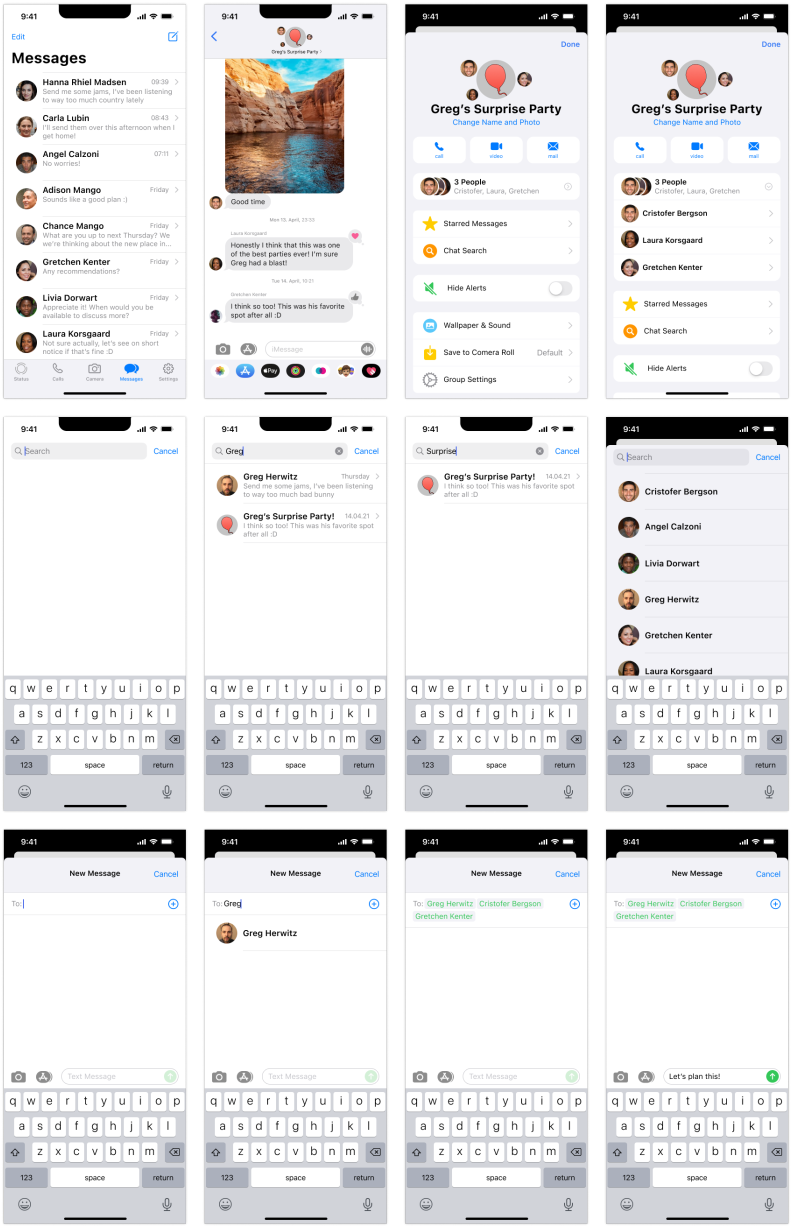
Appendix A: WhatsApp Prototypes
Collection of screens constituting the WhatsApp prototypes used during the experiment.

Appendix B: Instagram Prototypes
Collection of screens constituting the Instagram prototypes used during the experiment.

Appendix C: Post-Test Interview Questions
List of questions to be asked about each prototype and app during the post-test interviews.
Appendix D: Quantitative Test Results Box Plots
Box plots giving more information on the quantitative results of the experiments.

8.1 Appendix A: WhatsApp Prototypes

Collection of screens constituting the WhatsApp prototypes used during the experiment. iOS and Android variants.



Messages

Hanna Rhiel Madsen 09:39
Send me some jams, I've been listening to way too much country lately

Carla Lubin 08:43
I'll send them over this afternoon when I get home!

Angel Calzoni 07:11
No worries!

Adison Mango Friday
Sounds like a good plan :)

Chance Mango Friday
What are you up to next Thursday? We we're thinking about the new place in...

Gretchen Kenter Friday
Any recommendations?

Livia Dorwart Friday
Appreciate it! When would you be available to discuss more?

Laura Korsgaard Friday
Not sure actually, let's see on short notice if that's fine :D

Greg's Surprise Party

Good time

Mon 13. April, 23:33

Laura Korsgaard
Honestly I think that this was one of the best parties ever! I'm sure Greg had a blast!

Tue 14. April, 10:21

Gretchen Kenter
I think so too! This was his favorite spot after all :D

Greg's Surprise Party
Change Name and Photo

3 People
Cristofer, Laura, Gretchen

Cristofer Bergson
Laura Korsgaard
Gretchen Kenter

Starred Messages
Chat Search
Hide Alerts
Wallpaper & Sound
Save to Comera Roll Default
Group Settings

Greg

Greg Herwitz Thursday
Send me some jams, I've been listening to way too much bad bunny

Greg's Surprise Party! 14.04.21
I think so too! This was his favorite spot after all :D

Surprise

Greg's Surprise Party! 14.04.21
I think so too! This was his favorite spot after all :D

Cristofer Bergson
Angel Calzoni
Livia Dorwart
Greg Herwitz
Gretchen Kenter
Laura Korsgaard

New Message

To:

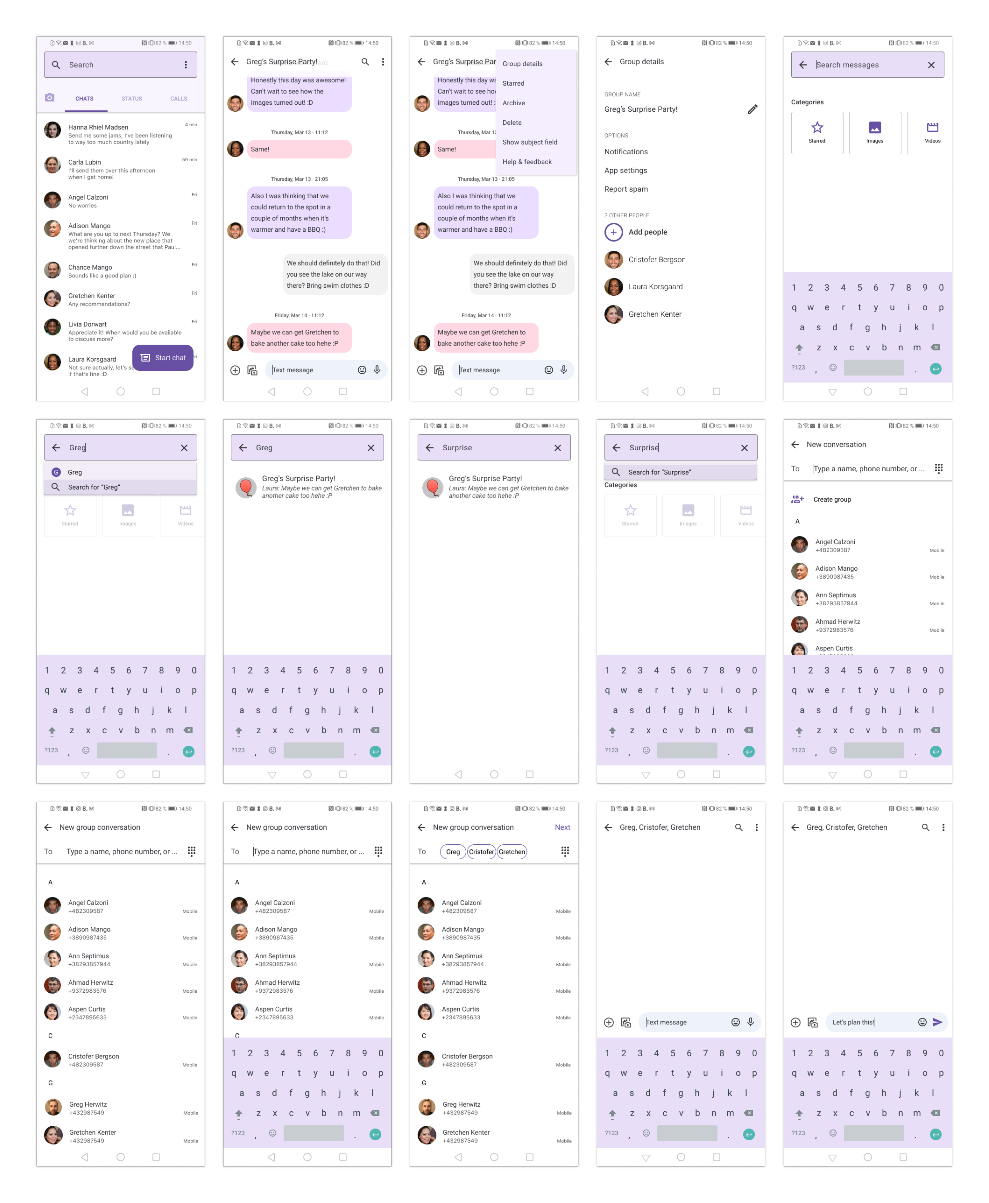
To: Greg

Greg Herwitz

To: Greg Herwitz Cristofer Bergson Gretchen Kenter

Let's plan this!

Figure 8.1: iOS WhatsApp prototype screens used during the experiment.


Hanna Rhiel Madsen 4 min
Send me some jams, I've been listening to way too much country lately

Carla Lubin 59 min
I'll send them over this afternoon when I get home!

Angel Calzoni Fri
No worries

Adison Mango Fri
What are you up to next Thursday? We we're thinking about the new place that opened further down the street that Paul...

Chance Mango Fri
Sounds like a good plan :)

Gretchen Kenter Fri
Any recommendations?

Livia Dorwart Fri
Appreciate it! When would you be available to discuss more?

Laura Korsgaard
Not sure actually, let's se if that's fine :D

Greg's Surprise Party!

Honestly this day was awesome! Can't wait to see how the images turned out! :D

Thursday, Mar 13 · 11:12

Same!

Thursday, Mar 13 · 21:05

Also I was thinking that we could return to the spot in a couple of months when it's warmer and have a BBQ :)

We should definitely do that! Did you see the lake on our way there? Bring swim clothes :D

Friday, Mar 14 · 11:12

Maybe we can get Gretchen to bake another cake too hehe :P

Group details

GROUP NAME
Greg's Surprise Party!

OPTIONS
Notifications
App settings
Report spam

3 OTHER PEOPLE
Add people
Cristofer Bergson
Laura Korsgaard
Gretchen Kenter

Greg's Surprise Party!
Laura: Maybe we can get Gretchen to bake another cake too hehe :P

Angel Calzoni	+482309587	Mobile
Adison Mango	+3890987435	Mobile
Ann Septimus	+38293857944	Mobile
Ahmad Herwitz	+9372983576	Mobile
Aspen Curtis	+2347895633	Mobile
Cristofer Bergson	+482309587	Mobile
Greg Herwitz	+432987549	Mobile
Gretchen Kenter	+432987549	Mobile

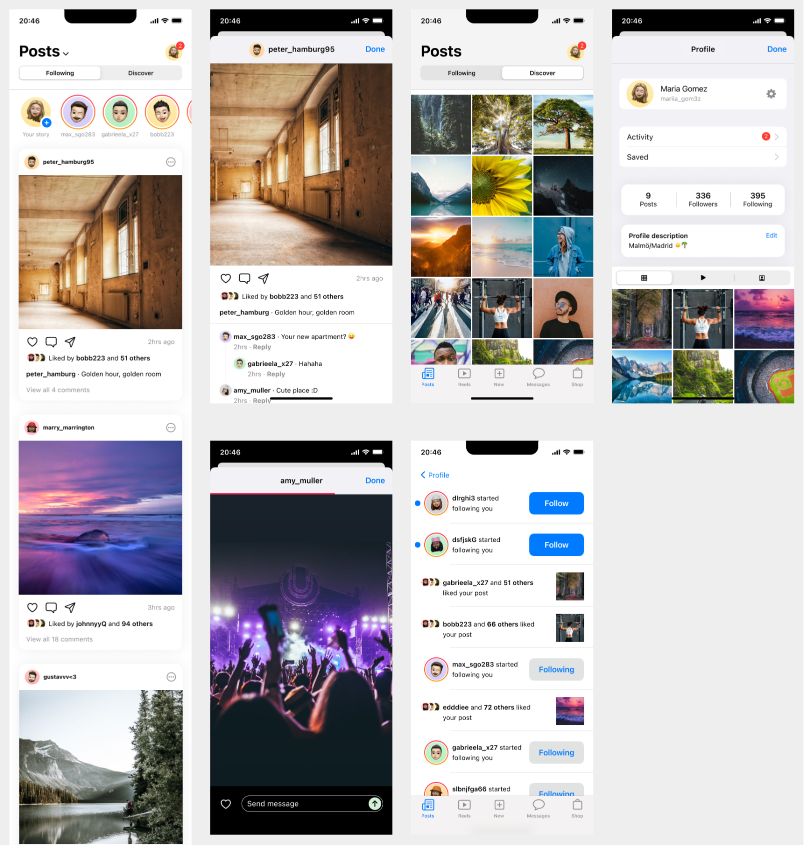
Greg, Cristofer, Gretchen

Let's plan this!

Figure 8.2: Android WhatsApp prototype screens used during the experiment.

8.2 Appendix B: Instagram Prototypes

Collection of screens constituting the Instagram prototypes used during the experiment. iOS and Android variants.



Liked by bobb223 and 51 others

peter_hamburg · Golden hour, golden room

View all 4 comments

Liked by johnnyyQ and 94 others

View all 18 comments

max_sgo283 · Your new apartment? 😛
2hrs · Reply

gabrieela_x27 · Hahaha
2hrs · Reply

amy_muller · Cute place :D
2hrs · Reply

Maria Gomez
mariia_gom3z

Activity 2

Saved

9	336	395
Posts	Followers	Following

Profile description Edit
Malmö/Madrid ☀️🌴

dlrghi3 started following you

dsfjskG started following you

gabrieela_x27 and 51 others liked your post

bobb223 and 66 others liked your post

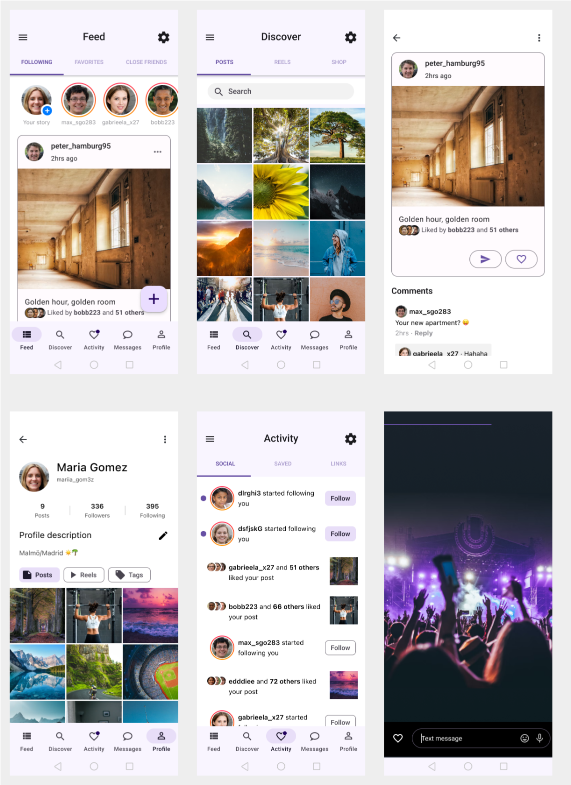
max_sgo283 started following you

edddiee and 72 others liked your post

gabrieela_x27 started following you

slbnjfga66 started

Figure 8.3: iOS Instagram prototype screens used during the experiment.



Golden hour, golden room
Liked by bobb223 and 51 others

max_sgo283
Your new apartment? 😮
2hrs · Reply

gabrieela_x27 · Hahaha

Malmö/Madrid ☀️🌴

dlrghi3 started following you

dsfjskG started following you

gabrieela_x27 and 51 others liked your post

bobb223 and 66 others liked your post

max_sgo283 started following you

edddiee and 72 others liked your post

gabrieela_x27 started

Figure 8.4: Android Instagram prototype screens used during the experiment.

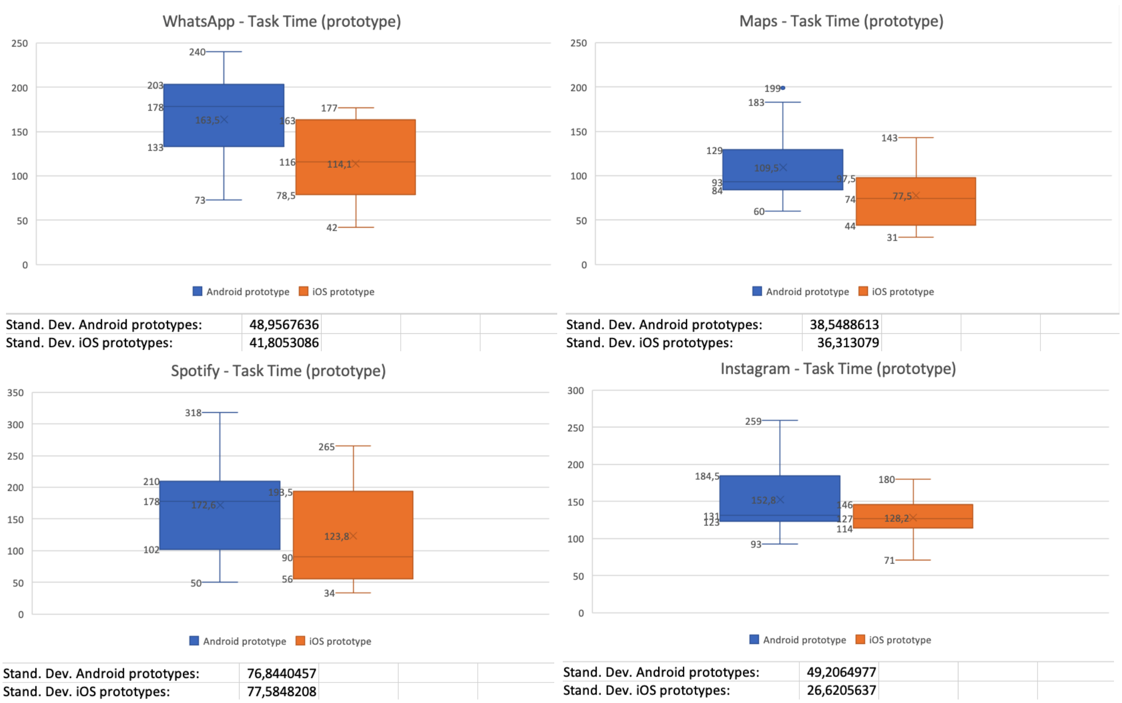
8.3 Appendix C: Post-Test Interview Questions

List of questions to be asked about each prototype and app during the post-test interviews.

- First of all, are there any thoughts that come to mind about this prototype/app?
- What did you like the most about this app?
- What did you like the least?
- Did anything feel particularly easy, straight-forward, or pleasant to do?
- Did anything feel particularly difficult, confusing, or frustrating to do?
- Situational questions.

8.4 Appendix D: Quantitative Test Results Box Plots

Box plots giving more information on the quantitative results of the experiments.



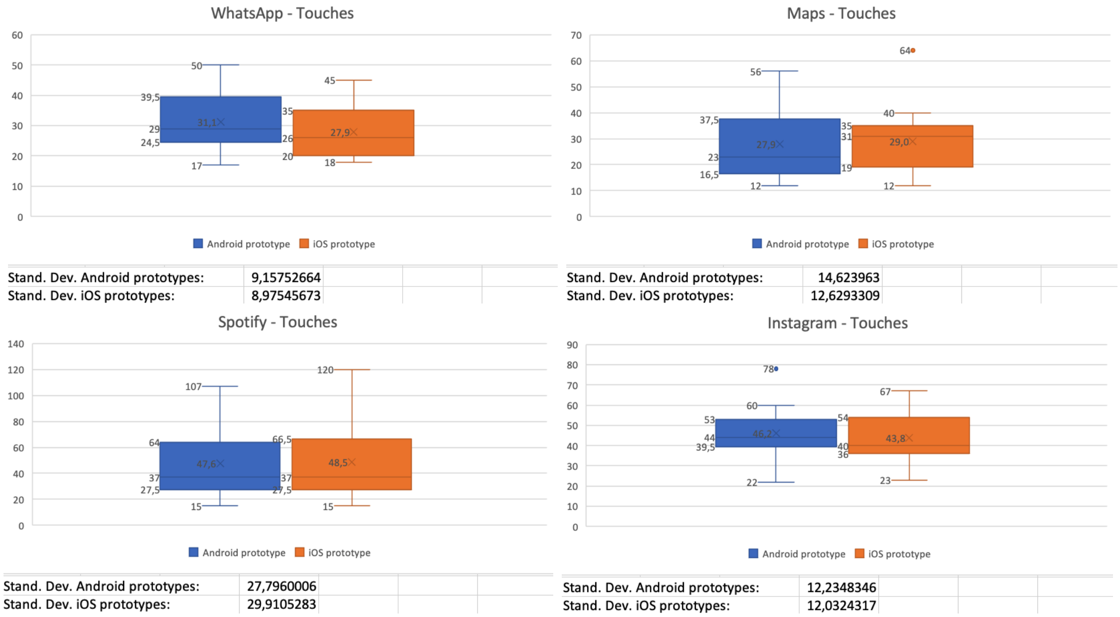
Stand. Dev. Android prototypes:	48,9567636
Stand. Dev. iOS prototypes:	41,8053086

Stand. Dev. Android prototypes:	38,5488613
Stand. Dev. iOS prototypes:	36,313079

Stand. Dev. Android prototypes:	76,8440457
Stand. Dev. iOS prototypes:	77,5848208

Stand. Dev. Android prototypes:	49,2064977
Stand. Dev. iOS prototypes:	26,6205637

Figure 8.5: The average task time on Android and iOS prototypes by prototype. This is the average time users need to complete all tasks of the according prototype.



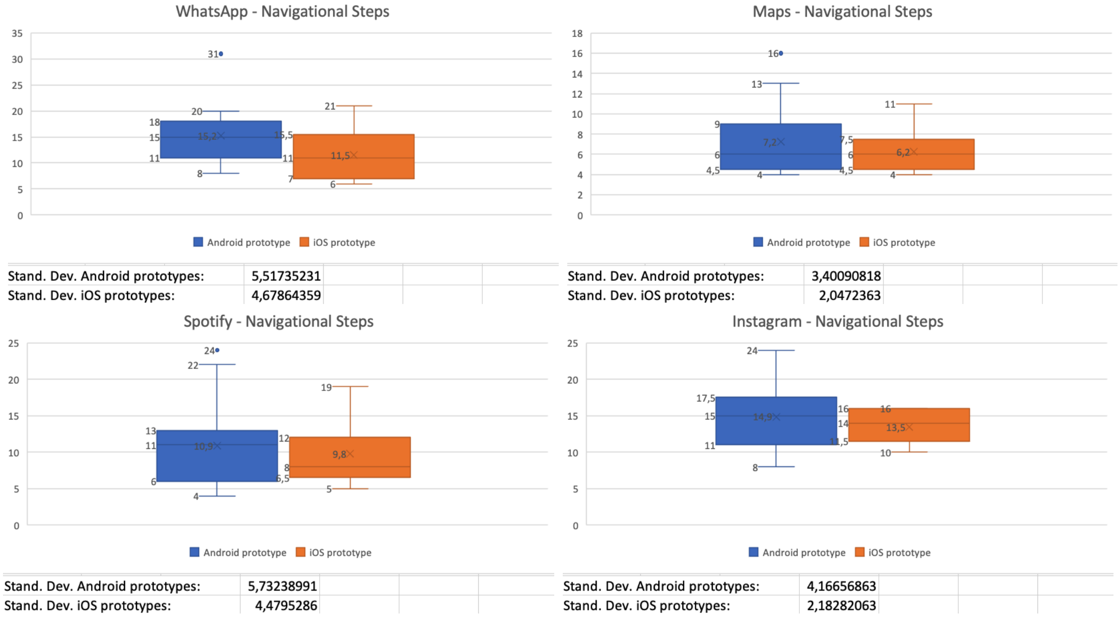
Stand. Dev. Android prototypes:	9,15752664
Stand. Dev. iOS prototypes:	8,97545673

Stand. Dev. Android prototypes:	14,623963
Stand. Dev. iOS prototypes:	12,6293309

Stand. Dev. Android prototypes:	27,7960006
Stand. Dev. iOS prototypes:	29,9105283

Stand. Dev. Android prototypes:	12,2348346
Stand. Dev. iOS prototypes:	12,0324317

Figure 8.6: The average number of touches on Android and iOS prototypes by prototype. This is the average number of touches users need to complete all tasks of the according prototype.



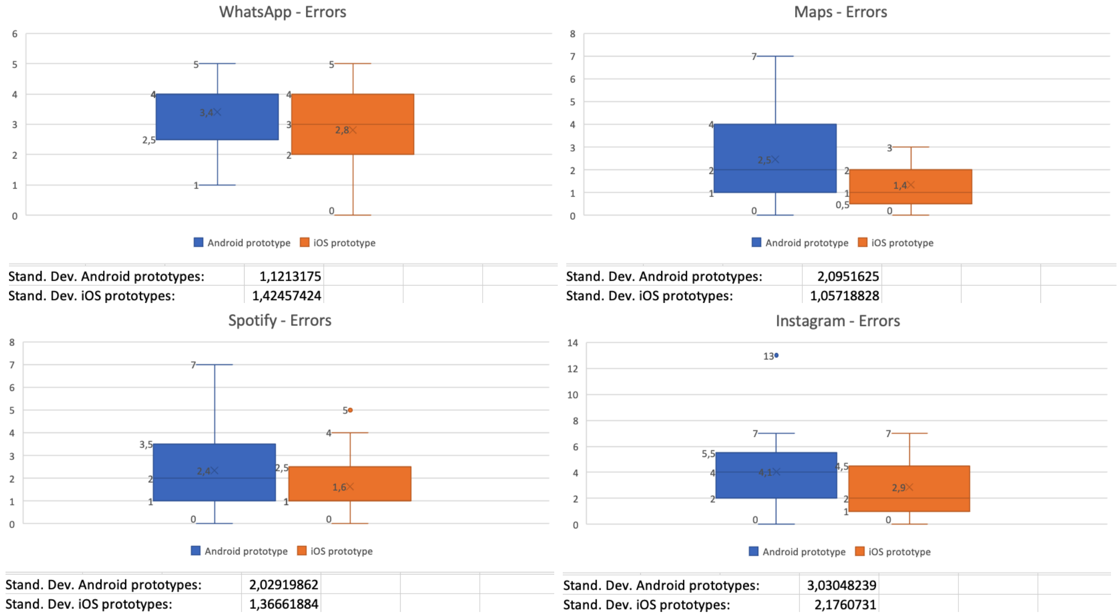
Stand. Dev. Android prototypes:	5,51735231
Stand. Dev. iOS prototypes:	4,67864359

Stand. Dev. Android prototypes:	3,40090818
Stand. Dev. iOS prototypes:	2,0472363

Stand. Dev. Android prototypes:	5,73238991
Stand. Dev. iOS prototypes:	4,4795286

Stand. Dev. Android prototypes:	4,16656863
Stand. Dev. iOS prototypes:	2,18282063

Figure 8.7: The average number of navigational steps on Android and iOS prototypes by prototype. This is the average number of navigational steps users need to complete all tasks of the according prototype.



Stand. Dev. Android prototypes:	1,1213175
Stand. Dev. iOS prototypes:	1,42457424

Stand. Dev. Android prototypes:	2,0951625
Stand. Dev. iOS prototypes:	1,05718828

Stand. Dev. Android prototypes:	2,02919862
Stand. Dev. iOS prototypes:	1,36661884

Stand. Dev. Android prototypes:	3,03048239
Stand. Dev. iOS prototypes:	2,1760731

Figure 8.8: The average number of errors on Android and iOS prototypes by prototype. This is the average number of errors testers commit while completing all tasks of the according prototype.