Chapter 3

Methodology and Study Design

This chapter explores the employed methodology for testing both the performance hypothesis H_P and the usability hypothesis H_U (defined in Section 1.3). The goal is to explicitly show the reasoning for the chosen methods as well as provide specific method implementation details to aid transparancy about the obtained results discussed in Chapter 5.
Generally, the employed methodology to achieve the research objective (Section 1.3) is to replicate a relevant subset of functionality of an existing iOS app using Flutter. Thereby, the **original app** acts as a baseline with which the **Flutter clone** can be comparatively evaluated. Based in this comparison, the research question - whether the Flutter framework can match native performance and provide equivalent usability (Section 1.3) - is answered. Instead of creating artificial use cases, taking advantage of an existing app provides realistic instances for performance as well as user interface testing.
The first section in this chapter (Section 3.1) details the decision process for selecting the baseline testing app as well as its feature reduction for further comparison. The subsequent two sections (Sections 3.2 and 3.3) explore the specific methods and reasoning for the performance and usability comparison respectively.

3.1 Baseline App Testing Decision Process

The procedure for choosing the case study app is based on a filtering process of 4 steps (i.e. application of constraints):

1. The app is built and maintained by apploft.

2. The app includes common application **facets**.

3. The app uses modern iOS framework technologies.

4. The app conforms to the human interface guidelines (HIG) by Apple (Apple Inc. 2021a).

The reasoning behind selecting the above filtering constraints is detailed in the following paragraphs.

Creation and Maintainance by apploft

This constraint was imposed on the filtering process such that a contact person (apploft employee) is available for code specific questions.
Having a reference to the original source code further provides the ability of implementing the Flutter replica similarly to facilitate comparability with the baseline app. E.g. a particular algorithm could be implemented similarly in the Flutter application. Thereby, equivalent time and space complexities are produced and algorithm implementation can be retracted as a confounding variable.
Furthermore, access to the original source code provides the ability to reduce unnecessary features which are irrevalent regarding the hypotheses evaluation. This reduces the complexity of the Flutter replica.

Inclusion of Common Application Features

The goal of this thesis is to determine whether Flutter is comparable in terms of performance (H_P) and usability (H_U) for the archetypal mobile app (Section 1.4). Therefore, only facets commonly appearing in iOS apps are considered for finding the baseline testing app.
For the purposes of finding the baseline app, a **facet** is defined as either

(a) a generalizable UI component which is non-trivial, or

(b) an underlying technical attribute influencing the user experience.

Trivial UI components (a) such as buttons or text weren't considered **facets** as they are omnipresent throughout every app. As for (b), a technical attribute has to influence the user experience to be incorporated as the purpose of this thesis is testing Flutter's value as a UI framework (see Section 1.3). For example, networking can be viewed as a **facet** if fetched data is displayed via the UI, but is not a **facet** if the sole purpose of networking within an app is to extract analytics data.

Use of Modern iOS Frameworks

If this constraint were not applied on the filtering process, old iOS technology could be compared to a modernly built Flutter app. Therefore, constraining the baseline app to be built with modern iterations of iOS framework technology ensures a reasonable comparison against the replica app.

Conformance to Human Interface Guidelines

Conforming to Apple's HIG ensures the original app looks native to the iOS platform. Since Flutter comes from Google, it probably implements the **Material Design** (Google Inc. 2021) rather well. Rebuilding an app that conforms to the Apple's design guidelines is the more interesting case.
In addition, providing a recognizable UX for iOS users would keep participants in the usability study (detailed in Section 3.3) focused on noticing differences instead of being distracted by an ambiguous UI.

Based on the above constraints, a small study was conducted looking at 15 apps developed by apploft (constraint 1) from 9 different iOS App Store categories. The factes were extracted into Table [initial table] by going through each user interface (constraint 2). Furthermore, a facet had to appear at least twice before being added as a result.
Continuing the filtering process, as per constraint 2 uncommon facets - facets appearing in less than 50% of observed apps - are excluded. This reduces the list of facets to the following:

- **Networking** - Interaction with a remote API.
- **Login/Authentication** - User log in meachanism through a UI.
- **Tab navigation** - UI component to quickly switch between different sections of app (Apple Inc. 2021d)
- **Hierarchical navigation** - Screens are opened on top of previous screens using a Stack structure (Apple Inc. 2021b).
- **Keyboard interaction** - A UI for inputing text via a software keyboard.

- **Vertically scrolling collections** - A UI collection of items scrolling vertically.
- **Horizontally scrolling collections** - A UI collection of items scrolling horizontally.
- **Webview component integration** - An integrated UI component in the app displaying web content (Apple Inc. 2021e).

Out of the 15 initially tested apps, 5 include all of the above **facets** (conforming to constraint 1 and 2) (see Table [table after applying constraint 2]). Kickdown (see Section ...) is chosen among the remaining contestants for the baseline testing app. It was most recently released (Feb 2021) and is therefore built with modern iOS technologies (constraint 3) and complies to the most recent iteration of the **Human Interface Guidelines** (constraint 4).

The login and signup mechanism - although a common **facet** - is removed from the original app for baseline testing. This is due to the fact that textfield and button interaction as well as networking is already present in other parts of the app and would yield no further insight regarding the hypotheses evaluation.
The Flutter app is implemented as closely as possible to the original application to avoid an asymmetrical comparison as detailed in Section **??**.

3.1.1 Kickdown App Feature Presentation

Here I will present what the original features of the Kickdown app.

3.2 Performance Comparison

The methodology chosen to test the performance hypthesis HP (Section 1.3) is a quantitative measurement of computational resources during app runtime. Measurements are performed for specific load conditions (i.e use cases). In the process, the original app acts as an empirical baseline for testing the Flutter replica against.
Directly benchmarking system resources provides insight whether the Flutter framework consumes compute resources efficiently under typically imposed load settings. Furthermore, system benchmarking metrics are the underlying cause of more ephemeral measures for testing the system load itself, e.g. page load time. In addition, the chosen compute resources (explained in Subsection 3.2.1) are easily measured using software tooling (Subsection 3.2.3) which aids the traceability as well as reproducibility of this particular study methodology.

Table 3.1: Generated by Spread-LaTeX

App	category	Networking	Login/ Authentication	Maps	Tab Navigation	Stack Navigation	Keyboard Interaction	Vertically Scrolling Collection	Horizontally Scrolling Collection	Webview Components	Camera Interaction
bonprix	shopping	1	1		1	1	1	1	1	1	
couponplatz	shopping	1	1	1	1	1	1	1	1	1	
Fernsehlotterie	lifestyle	1			1	1	1	1	1	1	
Gerolsteiner TrinkCheck	health			1		1		1		1	
Hexal Pollenflug	weather	1			1	1	1	1		1	
HIPP Baby	health	1	1	1	1	1	1	1		1	1
Hipp Bio	food	1				1		1		1	1
Hipp Buddies app	family					1		1	1		
Hipp Windel	shopping	1			1	1		1	1	1	1
Kickdown	shopping	1	1		1	1	1	1	1	1	
Lotto Nds.	entertainment	1	1		1	1	1	1	1	1	
Kulturpunkte	travel	1		1		1		1			
Servus TV	entertainment	1	1		1	1	1	1	1	1	
Starcook	food	1	1			1	1	1	1	1	1
Zeit Online	news	1	1			1	1	1		1	
	9	**13**	**8**	**4**	**9**	**15**	**10**	**15**	**9**	13	4

3.2.1 Selected Performance Measurement Variables

The following paragraphs introduce the selected performance measurement variables. Concretely, a brief definition is given as well as the reasoning for including the particular metric in this study with regards to evalutating the performance hypothesis (Section 1.3).

CPU Utilization

CPU utilization is defined as the CPU time (Free Software Foundation Inc. 1988) of a task divided by its overall capacity expressed as a percentage. The CPU as well as its integrated GPU[1] are responsible for graphics rendering related computations. Generally, high CPU usage is an indicator of insufficient processing power to perform the executed task. Therefore, testing CPU utilization directly assesses whether Flutter's framework processing requirements for UI rendering can be fulfilled given the testing hardware (see Subsection 3.2.2).

Frames per Second

Frames per Second (FPS) describes the rate at which the system, and in particular the GPU, completes rendering individual frames. The FPS rate directly determines the smoothness of UI animations and transitions (Goolge Inc. 2020).

Memory Utilization

Memory utilization is the percentage of available memory capacity used for a specific task. A high level of memory usage negatively impacts performance of running tasks as well as interactive responsiveness (Ljubuncic 2015).

3.2.2 Measurment Process

To reduce measurement confounders, the device is restarted before each individual measurement to ensure that all irrelevant background processes are cancelled.
The measurement process for the individual metrics is further split into specific user actions which are executed and tested on both the iOS and Flutter app separately. These were chosen to test all relevant facets of the app (see Section **??**) and ensure a necessary load on the system:

- **app start:** The app is freshly installed on the test device, opened and idle until the visible postings are loaded.

1. The GPU is an integrated chip within the System-on-a-Chip (SoC) architecture (G. Martin 2001) for all iPhone processing units (WikiChip 2020).

- **scrolling:** On the postings overview screen, the posting cards are vertically scrolled fully to the bottom and subsequently back to the top.
- **detail view:** From the postings overview, the first posting is tapped to navigate to the detail view. Afterwards the back button is tapped to navigate back to the overview.
- **image gallery:** The image gallery of a posting is opened from the detail view of a posting and the first 10 images are viewed by swiping.

For each **user action**, the average of all values over time is recorded. This process is then repeated 3 times and averaged. The exact number of experiment repetitions was chosen as a tradeoff between marginal accuracy increase and additional experiment execution time.
Furthermore, 2 testing rounds are devised on separate devices. The iPhone 12 and iPhone 6s are chosen as the upper and lower bounds of hardware performance respectively. The lower bound is defined in this case as per Apples recommendation to set the deployment target to the current operating system version (iOS 14 at time of writing) minus one (iOS 13) which lists the iPhone 6s as the oldest supported device (Apple Inc. 2021c). iOS 13 is also the minimum deployment target for the Kickdown app.

3.2.3 Profiling Tools

Xcode Instruments (Apple Inc. 2019) - a part of the **Xcode** IDE tool set - are used for profiling the individual metrics. It provides multiple preconfigured profiling trace instruments. For the purposes of this thesis, the **Time Profiler** tool (see Figure ...) is used for CPU (see Paragraph 3.2.1), the **Core Animation** tool (see Figure ...) for FPS (see Paragraph 3.2.1), and **Allocations** tool (see Figure ...) for memory usage (see Paragraph 3.2.1) quantification over time.

3.2.4 Evaluation Process

To better understand the data gathered, it is subsequently examined using exploratory data analysis (EDA) (Tukey 1977). *To be continued a bit...*

3.3 User Experience Comparison

This Section explores the usability hypothesis evaluation methodology. Specifically, laying out the procedure to answer the question of whether or not the Flutter framework is capable of reproducing native iOS application user experiences (see Section 1.3).

Generally, as UX is built for other humans, any evaluation is prone to subjectivity and perception biases (Tversky and Kahneman 1974). Therefore, it is difficult to capture these impressions quantitatively in a reproducable manner.
However, the user experience of an app is directly dependant upon sufficiently underlying performance (e.g. for scrolling fluidity). Therefore the results of the performance comparison (Section 5.1) form the basis of the evaluation of the usability hypothesis H_U. Utilizing a mixed approach as a study methodology combining both the quantitative performance comparison and a qualitative method will draw upon the strengths of both approaches.

Specifically, *semi structured interviews with subject matter experts* (see Liebold and Trinczek 2009) are conducted to evaluate the baseline application with the Flutter replica by asking the participants for differences between the two apps (further explained in Subsection 3.3.2). This methodology has the advantage of covering predetermined topics relevant to the research question while also allowing spontaneous discussion possibly leading to novel insights.
Furthermore, expert interviews are an especially useful approach for scientific explorations with no or scant preexisting theory (cf. Bogner et al. 2009).

As soon as no new perspectives seem to emerge during interviews, no further interviews will be conducted. This is based on the fact that the method of expert interviews aims to provide a breadth of perspectives on a given topic unlike specific quantitative anaylses (see Liebold and Trinczek 2009).

3.3.1 Interview Preliminaries and Technicalities

The interviews are conducted with employees of apploft. They qualify as subject matter experts in the sense that they have been working in the mobile app industry for multiple years. They come from a variety of professional backgrounds including UX and UI design, project management as well as software engineering. Furthermore, some interviewees have actually worked on the original app itself. This diversity among the study participants is especially relevant in order to explore a breadth of perspectives. Due to the ongoing Covid-19 pandemic, the inter-

views are conducted through video calls, and the interviewees are asked to share their iPhone screen via Quicktime player (Apple Inc. 2014). The moderation and recording is facilitated by the author. For the comparison, the interviewees receive QR codes with which both apps may be downloaded. Behind each distributed code is a downloadable IPA (iOS App Store Package) binary executable file hosted by an HTTP server. These work exactly the same as any other apps downloaded from the iOS App Store. Both apps are blindly distributed to the participants as "Kickdown A" and "Kickdown B" in order to remove confirmation bias as a confounder (see Tversky and Kahneman 1974).

3.3.2 Interview Guideline

The interview guideline (see Appendix **section::interview_guideline**) is based on finding out perceptual differences between the iOS baseline and Flutter replica app.

Just like the performance comparison, use cases associated with particular UI facets form the basis for the evaluation:

- **App Start and Scroll Behavior** - ...
- **Detail Transition, Modal Transition, Textfield interaction** - ...
- **Horizontal Scrolling** - ...
- **Switch Interaction** - ...

The interviewees are asked to perform a particular use case for app A and app B. Then they are asked to detail differences between the two apps. Asking this open-ended question aims at receiving as much information possible about perceptual differences (Cf. Helferrich 2011, 182–185). Subsequently, the participant is asked to determine which of the two apps felt more natural (i.e. had a better UX). A determined trivalent response of: "A", "B" or "same" is expected. The goal of this question is to get overall impression of the usability.
Both questions are asked after each use case execution of the participant. To maintain a high participant engagement during interviews, use cases are described in a more captivating way, e.g. "Please find the blue Mercedes SUV in Kickdown A [Wait until participant has found it.]. Now, please look for the black BMW convertible in the other app.". After each use case, the ordering of app A and B is swapped. E.g. if the first case starts with A, the second starts with B. This choice is made as to avoid recency bias (Cf. Atkinson and Shiffrin 1968). Furthermore,

the ordering is also swapped after each interview. In this way, participant X starts with app A while participant Y starts with B. Finally, after the last use case, the participant is asked to answer the two questions with regard to the entire application.

3.3.3 Interview Evaluation

The videos from the interviews are transcribed into a textual format and further processed using **interview coding**. Thereby, each interview is categorized into semantic themes. These themes among all interviews are then merged into an overall theme structure - also known as code structure. This code structure forms the basis for the evaluation of the usability hypothesis H_U.

Bibliography

Apple Inc. 2014. *Quicktime Download,* February. https://support.apple.com/de_DE/downloads/quicktime.

———. 2019. *Instruments Help.* https://help.apple.com/instruments/mac/current/.

———. 2021a. *Human Interface Guidelines.* https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/.

———. 2021b. *Navigation.* https://developer.apple.com/design/human-interface-guidelines/ios/app-architecture/navigation/.

———. 2021c. *Supported iPhone models.* https://support.apple.com/guide/iphone/supported-iphone-models-iphe3fa5df43/13.0/ios/13.0.

———. 2021d. *Tab Navigation.* https://developer.apple.com/design/human-interface-guidelines/ios/bars/tab-bars/.

———. 2021e. *Web Views.* https://developer.apple.com/design/human-interface-guidelines/ios/views/web-views/.

Atkinson, Richard, and Richard Shiffrin. 1968. "Human Memory: A Proposed System and Its Control Processes." *Academic Press* 2.

Bogner, Alexander, Beate Littig, and Wolfgang Menz. 2009. "Introduction: Expert Interviews — An Introduction to a New Methodological Debate." Chap. Interviewing Experts, 1–13. Springer.

Free Software Foundation Inc. 1988. *GNU C Library Documentation: CPU time.* https://www.gnu.org/software/libc/manual/html_node/Processor-And-CPU-Time.html#Processor-And-CPU-Time.

G. Martin, H. Chang. 2001. "System-on-Chip design." *ASICON 2001.*

Google Inc. 2021. *Material Design.* https://material.io/design.

Goolge Inc. 2020. *Flutter performance profiling.* https://flutter.dev/docs/perf/rendering/ui-performance.

Helferrich, Cornelia. 2011. *Die Qualität qualitativer Daten.* VS Verlag.

Liebold, Renate, and Rainer Trinczek. 2009. "Handbuch Methoden der Organisationsforschung." Chap. Experteninterview. VS Verlag für Sozialwissenschaften.

Ljubuncic, Igor. 2015. "Problem-Solving in High Performance Computing." Chap. Fine-tuning the system performance.

Tukey, John Wilder. 1977. *Exploratory Data Analysis.* Addison-Wesley.

Tversky, Amos, and Daniel Kahneman. 1974. "Judgment under Uncertainty: Heuristics and Biases." *Science* 185, no. 4157 (September).

WikiChip. 2020. *Ax - Apple.* https://en.wikichip.org/wiki/apple/ax.

Appendix A

Appendix

A.1 Interview Guideline

A.1.1 Interview Setup

- brief interviewee about thesis and their role in the research
- ask if the interview session may be recorded and further processed for scientific inquiry
- ensure technicalities before start of interview with participant:
  - the participant's iPhone is sufficiently charged, has been restarted and is in "Do Not Disturb" mode
  - the participant has installed Quicktime player on their iPhone and are sharing it with their MacBook
  - the participant is sharing their MacBook screen in the video call
  - the participant has received the QR codes for kickdown A and B
- ask the participant if the recording may started

A.1.2 Background Questions

Ask the participant about their...

- job role
- years of experience in the mobile app industry
- previous experience with the Kickdown application

A.1.3 Main Interview

App Start and Scroll Behavior

Instructions

- Please open Kickdown (A/B) and find the [color] [brand] [attribute] car.
- Please open Kickdown (A/B) and find the [color] [brand] [attribute] car.

Questions

1. Are there any differences in terms of user experience in any way between the two apps?
2. If you had to pick one experience over the other, which would you choose (A or B)?

Detail Transition, Modal Transition, Textfield interaction

Instructions

- Please stay in Kickdown (A/B) and tap on a car of your choice. Bid on the car with an amount of your choice.
- Please repeat the process for the Kickdown (A/B). You may choose another car and enter a different amount.

Questions

1. Are there any differences in terms of user experience in any way between the two apps?
2. If you had to pick one experience over the other, which would you choose (A or B)?

Horizontal Scrolling

Instructions

- Please stay in Kickdown (A/B) and open the first posting. Find the picture with the [insert item].
- Please open Kickdown (A/B) and open the second posting. Find the picture with the [insert item].

Questions

1. Are there any differences in terms of user experience in any way between the two apps?
2. If you had to pick one experience over the other, which would you choose (A or B)?

Instructions

- Please stay in Kickdown (A/B) and use the tab navigation to navigate to the "More Screen". Please turn Tracking on.
- Please repeat the process for Kickdown (A/B)

Questions

1. Are there any differences in terms of user experience in any way between the two apps?
2. If you had to pick one experience over the other, which would you choose (A or B)?

Overall

Instructions

- You may test out any functionality of the app that you would like to have.
- *The following questions regard their impression of the entire app*

Questions

1. Are there any differences in terms of user experience in any way between the two apps?
2. If you had to pick one experience over the other, which would you choose (A or B)?