Interface Design for a Mobile IDE for Mobile Application

Md Ismail Alam Mokul
University of Waterloo
Waterloo, Canada
miamokul@uwaterloo.ca

Md Omor Faruk
University of Waterloo
Waterloo, Canada
mofaruk@uwaterloo

Derrick M. Wang
University of Waterloo
Waterloo, Canada
dwmaru@uwaterloo.ca

ABSTRACT

Mobile application development is usually done across multiple devices, but the processing power of modern day mobile devices is more than enough to handle the development process, not to mention the convenience it can bring by reducing the number of devices involved. The fact that no widely used development environment exists on mobile suggests there are major issues that prevent developers from using mobile devices to program. In this research, we reviewed literature and worked with developers and students to examine the potential concerns and expectations they may have towards the concept of single-device mobile development, and attempt to design an interface that addresses these concerns and expectations. This study shows that the developers do not refuse the concept of mobile application development in mobile IDE for any distinct issue other than the fact that they have never tried this method of development.

CCS Concepts
•Human-centered computing → User interface design; Scenario-based design; Usability testing; Interface design prototyping.

Author Keywords
Interface Design; Usability Test; Programming Interface; Mobile Development; Prototyping.

INTRODUCTION

The demand and utility of Mobile applications (apps) are ever-increasing. Mobile apps are usually designed and programmed on a desktop computer, brought to mobile devices for preview and evaluation, then back to the desktop computer for the next iterations. Therefore, the development of mobile applications is dependent on having multiple devices where features such as real-time visualizations are hard to achieve and developers have to switch between devices to preview changes to the code. Of the two major Operating Systems (OS) [1], while Android's development kit allows developers to use either an emulator or real devices (requires special set-ups) connected to the computer to view the program code in action[2], Apple's iOS not only forces developers to use its own MacOS computers, but also offers no direct device views like Android does [3]. What these two examples have in common is that a computer is always required in the development process of a mobile device.

CHI 2020, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6708-0/20/04...$15.00.

The availability of multiple devices is not always the case. There are scenarios such as sitting on a bus or an airplane where developers can do some work, but unable to do so because it is hard to use multiple devices due to the physical space limitation. Such times, while fragmented, can potentially be utilized to achieve more progress on the development.

With larger mobile screens and multifaceted interaction techniques, it is now a possibility that mobile application development can be done within a single mobile device. As such, we can greatly reduce the dependency on multiple devices and offer a more convenient way to develop mobile apps. In this study, we attempt to create a mobile interface that, assuming all required functionalities are working, allows developers to create mobile apps efficiently and comfortably.

BACKGROUND AND PREVIOUS WORKS

As the aim of this study is to design a user interface that not only is "good", but also a mobile version of a desktop application, we needed to first understand what a user interface is, what difference exists between desktop and mobile applications, and what denotes good practices in designing interfaces. In this section, we will briefly introduce some previous works we referred to while designing the interface and the methods of evaluation used in our study.

User Interface

The user interface is a set of commands or menus that is offered to their users to communicate with a program[4]. A friendly user interface for a mobile application is critical for retaining the user's loyalty to the application[4]. It has been proven many times that a powerful program might have very little value when the user interface is poorly designed[4], [5].

User interface design requires consideration of both the hardware and the software of the system. The hardware for a mobile device decides its looks and feels, such as dimensions, screen size, weight, shape, touch actions, input methods, and resolution. On the other hand, the "software" mostly focuses on menus, navigations, and graphics displayed on the screen [5], [6].

There are many previous works on interface design for mobile applications that reflect the critical limitations of mobile devices as they have relatively limited memory capacities, screen dimensions, keyboard sizes, and methods of interaction[6]–[8]. These limitations demand a more intuitive and simple interface design[7]. For a mobile interface to be user-friendly, it is crucial for designers and developers to keep these limitations in mind [9]. Nilsson, in his study [10], categorized the said limitations into three groups: utilizing the screen space, interaction mechanisms, and design. The most important concepts for mobile interface design are the context of the application and the context in which a user uses the application[11]. Ayob et al. [4] pointed out several considerations for mobile user interface designs:

- Identify what users want from the mobile device.
- Understand the characteristics of users and the environment.
- Develop a system that meets the users' needs.
- Test the system to see if it promotes user satisfaction.

Good Practices in Mobile Interface Design

There are several standards, guidelines and best practices for mobile interface design. Ayob et al.[4] reviewed four guidelines, namely Shneiderman's Golden Rules of Interface Design, Seven Usability Guideline for Mobile Device, Human-Centered Design (ISO Standard 13407) and W3C Mobile Web Best Practices. While the study was conducted in 2009, the guidelines reviewed in their study are essential starting points for the mobile interface design approach. They, in their study, provided a three-layered design guideline for mobile application design (See Table 1, reproduced from [4]).

Gong et al. compared and evaluated how guidelines for desktop applications overlap ones for mobile applications. Their study suggested that four desktop design principles can be directly applied to mobile devices, namely designing shortcuts for frequent actions, designing informative feedback, designing dialogs to yield closure, and supporting internal locus of control [12]. Given the limited amount of information on the screen, users also need feedback about the arrangement of information through the help menu and pop-ups [8]. It is also essential that the interface design adopts standard mobile functionalities such as text input methods,

Phases	Exising Guideline	Activities
Analysis (Context of Use)	Specify user and organizational requirements	1. Identify and document user tasks 2. Identify and document organizational environment 3. Define the use of the system
Design (Context of Medium)	Produce design solutions	1. Design shortcuts for frequent actions 2. Design informative feedback 3. Maintain consistency 4. Reverse actions 5. Prevent errors, design error handling mechanisms 6. Reduce short-term memory load 7. Design for multiple, dynamic contexts 8. Design for small devices (responsive) 9. Design for speed and recovery 10. Design for "top-down" interaction 11. Allow personalization 12. Prevent repeating navigations across multiple pages 13. Distinguish selected items clearly
Testing (Context of Evaluation)	Evaluate the design against user requirements	1. Use quick and dirty approaches 2. Perform usability testing 3. Perform field studies 4. Perform predictive evaluations

Table 1. Three-layer design guideline for mobile applications.

menu and command organizations to avoid users' disorientation [8].

Interestingly, some studies have shown that user expectations for a mobile application are well reflected by user reviews on that application. Khalid et al.[13] concluded that most users complain about the mobile application are related to their functionality issues such as crashing, lagging (heavy resource demands), compatibility errors, and so on. Two specific complaints (feature request and interface design) were related to the interface design of the application. Feature request implies that the app needed some additional features and some ways to customize it. Interface design relates to the design, control, and visuals of the application.

Horizontal display

To overcome the limitation presented by the small mobile screen, the horizontal layout is often suggested. While

Figure 1. The horizontal layout of the iPhone leaves only 40% of the display area for information display.

the horizontal display on a phone greatly reduces the information on the screen, it does provide a much larger typing area, thus could potentially lead to fewer mistakes (Figure 1).

Visual Scripting Systems

There exist visual scripting systems such as the Unreal Engine's Blueprints feature[14], where instead of writing codes manually, developers choose from blocks of pre-defined functions, that create connections of functions to denote how the program works (refer to Figure 2). This method greatly reduces the number of user inputs and the chances of making mistakes. However, visual scripting systems are not popular among programmers, because it limits the programmers flexibility to explore.

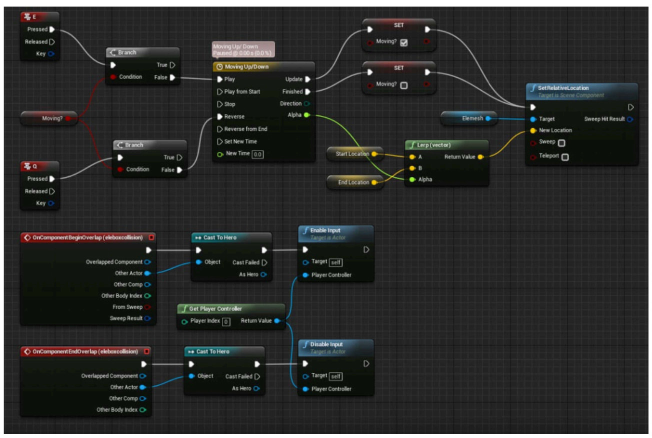

Figure 2. An example of Unreal Engine's blueprint scripting system [taken image from Unreal Engine by Epic Games].

Programming Languages for Mobile Screens

Tillmann et al. [15] developed a programming syntax and interface named TouchDevelop (Figure 3), where instead of writing all the codes, a major fraction of the coding is done by sets of pre-defined actions, such as conditional statements and structures, that occur very frequently during the coding process. Removing those from the necessary manual typing can also greatly reduce the amount of user inputs and thus chances of making mistakes.

Reflecting on the above literature, this study attempts to answer the primary question of how do we design a user-friendly mobile application interface for mobile application development IDE to reduce the dependency on desktop devices. To answer the primary question, we tried to answer three secondary questions: 1) why and under what circumstances does a user want to use a mobile IDE for mobile application development, 2) what are the expectation of users from mobile IDE for mobile application development, 3) what are the features that make a mobile IDE easy for the users? We focused on mobile platforms in this study, thus the type of IDE we would be working with is mobile development, rather than desktop development. It is only logical that desktop developments should be carried out on desktop devices for only they can provide the right environment for testing. However, whether a desktop device is necessary for mobile development remains unknown, and in this study, we aim to explore this unknown. Considering that all researchers possess only Apple iPhones, any findings from the study may or may not apply to devices running in other Operating Systems (such as Android)

Figure 3. TouchDevelop's interface [15]**.**

We adapted a scenario-based design method [16] to envision the possible scenarios where programmers might need to use their mobile phones to program, and design our interface accordingly. Three possible scenarios that we validated with participants later to determine whether they were realistic are as following:

- On a moving vehicle, e.g. bus and airplane for an extended period of time (hours), having limited space and freedom.
- At a location where no desktop computer/laptop is present (e.g. listening to a talk) when a sudden inspiration flashes, easy to program but hard to record in other formats.
- Quick trials and demonstrations when desktop devices are not present, or would take too long to prepare.

METHODOLOGY AND STUDY DESCRIPTION

To address the research questions and objectives of this study, we adopted a task-centered design process, which included interviews and usability testing with participants in every stage (See Figure 4). The study followed four major steps with overlapping and continuous evaluation from the users: i) identification of users and specific requirements, ii) analysis of the requirements, iii) Design and development of a low-fidelity prototype and usability testing, iv) development of a medium-fidelity prototype and usability testing with participants.

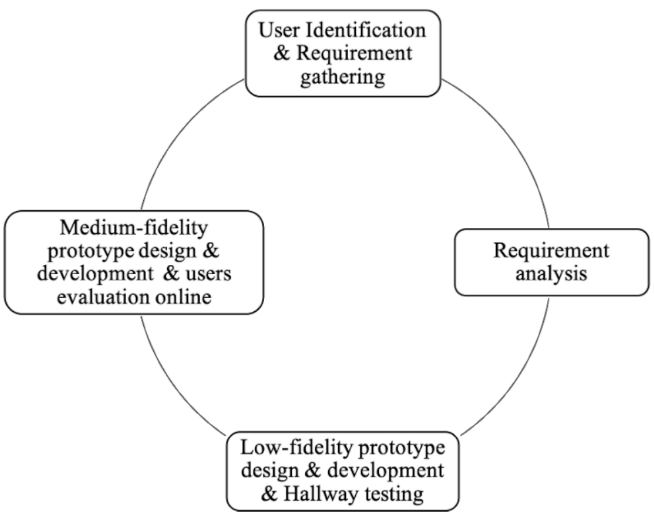

Figure 4. Task-centered design process for the study.

Participants

Considering the interface we would design is a for mobile devices, and the fact that we would be unable to distribute the test interface remotely without a Mac computer and a subscription-based developer account from Apple[17], we initially decided that for the best usability testing results, we would only be recruiting participants who could physically join us, therefore, we listed several of types of participants we would reach out to.

- Students with programming experience, in the University of Waterloo or other institutions in the region.
- Developers working in tech companies in the K-W region, e.g. Google and Shopify.
- Independent mobile app developers in the region.

However, due to study limitations, we could avail only 7 participants who agreed to participate in interviews and usability tests involved in this study. The participants' age ranged from 19 to 24 years old (M=21) and among them 1 was female. Participants were divided into three categories based on their knowledge and experience in programming (see Table 2). None of the participants were professionally involved in mobile application development. To increase the validity of the evaluation, we invited personal friends who were working in major tech companies, namely Amazon and Google, to offer their feedback on the prototypes.

Participants	Programming Exp.	Count
ECE Students	Less than 1 year	2
CE and ECE Students	Between 1 and 3 years	3
Working Professionals	Over 3 years	2

Table 2. Participants details.

Study Description

Participants were first interviewed with open-ended questions to learn about the circumstances under which a mobile IDE for development of mobile application could be useful and what expectations they had for such an application. After figuring out the contexts that might be appropriate for using a mobile IDE, we asked the participants about their specific requirements, given some contexts/ scenarios. Scenario-based questions made the participants to mentally construct the real situation and provide us with their requirements. The feedback was then categorized into a list of features that our interface needed to include in our first low-fidelity prototype.

Requirements were categorized into two groups: i) usefulness of the application and ii) features and expectations of such an application from a user point of view. Requirements were analyzed to understand the underlying assumptions and how they could be accommodated in the mobile interface. Analysis of the requirements led to the drafting of two low-fidelity prototypes and sharing with the same participants to evaluate the interface design. Participants were shown the low-fidelity prototype and asked whether they would consider using a mobile IDE to develop mobile applications. Initially, all participants rejected the idea of using mobile phones to develop applications. In further interviews, participants provided several concerns that prevent them from attempting to use such applications. Two low fidelity interfaces were designed iteratively, addressing the concerns of the participants.

The results of the evaluations were utilized to design the first medium-fidelity prototype using an online software tool 'Figma'[18]. The interface was shared with the participants for their feedback. The final medium-fidelity prototype was designed based on the concerns received from the participants. However, due to technological limitations during the COVID 19 situation, we could not conduct another evaluation of the final medium-fidelity prototype. Thus, usability testing could not be conducted traditionally due to the imposition of physical distancing in the study location.

INTERFACE DESIGNS AND EVALUATION

Following the iterative process, we prepared two low fidelity prototypes before attempting to design a medium prototype. After designing the first medium-fidelity prototype, we shared the outcome with the participants. Addressing those issues, we prepared our final version of the medium-fidelity prototype. Subsequent paragraphs describe the designs and evaluations.

Low-fidelity Design 1

Before embarking on a low-fidelity design, we conducted a scenario-based discussion with participants to learn about their perception of the complete and stand-alone mobile IDE. Important responses from the participants are given in Table 3.

Scenario	Context	Requirement mentioned
1	user/developer is on a vacation	Current mobile keyboard is not feasible. Expect different shortcuts such as special characters
2	User/developer is travelling (long journey, 7-8 hours)	Magnifying screen, loop shortcuts (may be in form of a shortcut button since loops are used frequently)
3	User/developer has no access of laptops (common)	IDE should be compatible with different devices (of different configuration)
Common Responses		1. Cloud system for real time integration between mobile IDE and desktop IDE 2. Different types of themes, testing/debugging feature, ease to window switching (easy swipe left/right) 3. Getting syntax suggestion, auto alignment feature for written code 4. All other major features a desktop IDE offers such as find and replace, database connectivity

Table 3. Scenario-based requirements.

The requirements received from the participants are listed below:

- Quick preview.
- Highlight sections.
- Quick access to functions.
- Numerous themes.
- Code alignment/format.
- Compiler.
- Debugger.
- Multi-file development.
- Collaboration.
- Synchronization between desktop and mobile IDEs.
- Syntax suggestion.
- Find and replace.

The objections shared by the participants on their reservation of using a mobile IDE independent of a desktop are the following:

- Relatively small screens leading to difficulty in reading codes
- Soft keyboard is uncomfortable to type on.
  – Create shortcuts for frequently used symbols.
- Difficulty to move the typing cursor.
- Possible discomfort coding in moving vehicles.

We sketched our first low-fidelity prototype (Figure 5) iteratively to capture the users' requirements in the design. The sketch was modified by taking the cumulative feedback provided by the participants.

Low Fidelity Design 2

After compilation of the feedback on our first low-fidelity prototype, we designed a second prototype (Figure 6). We presented our second low-fidelity prototype to the same participants to evaluate. In addition to the feedback from the participants, we also compared the prototype in two ways: 1) coding websites opened in the default Safari browser on the iPhone, and 2) coding apps such as Koder and Code Editor by Panic on the iOS store.

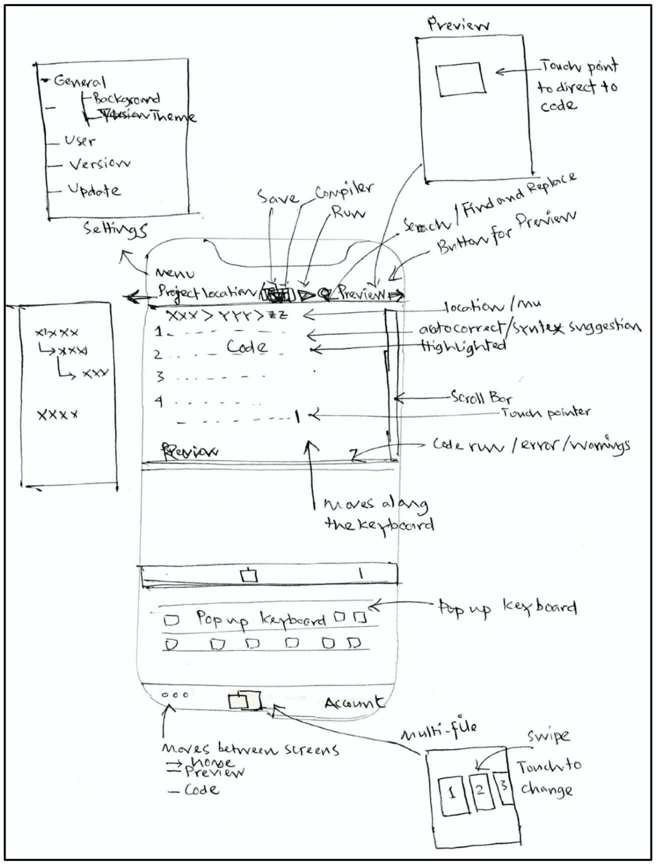

Figure 5. Low-fidelity design 1.

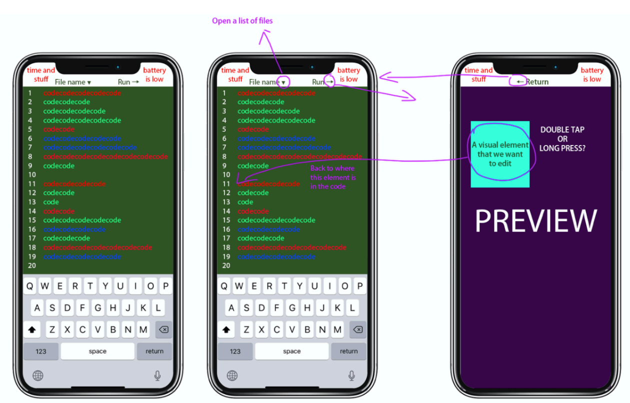

Figure 6. Low-fidelity design 2.

Medium Fidelity Design 1

For the first mid-fidelity prototype, we used a tool named Figma to create a prototype with interactions (See Figure 7). Many requested features were only reflected as menu items on the interface, whether it was technically possible remained unknown and was not addressed in this research. This prototype only served as a visual example of one design of the interface and interaction, as the coding example shown on the prototype is written in HTML, a web programming language, instead of a mobile development language such as Swift.

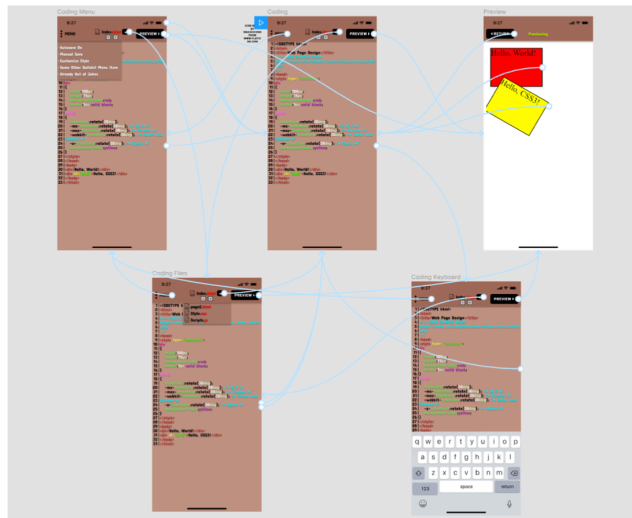

Figure 7. Medium fidelity design 1.

Figure 8. Medium fidelity design 2 [18]**.**

Medium Fidelity Design 2

The final medium prototype is given in Figure 8. The final version took into consideration all the requirements and feedbacks received from the participants. However, no evaluation of the final design could be carried out due to technical issues during COVID 19 situation.

DISCUSSIONS

Complete stand-alone mobile IDE for mobile application development is a new concept. Thus, initially, all participants rejected the idea of coding on a mobile phone, but such a negative reaction was not a result of bad experience, but a lack of experience with such a concept. Once exposed to a prototype, visually recognizing the concept, participants provided positive feedback, stating they would help test the interface design and would consider using such an application if desired functionality could be achieved.

Another interesting finding was that all participants had stated that they were more likely to test sections of codes than to develop an entire application solely on mobile. They will also use it in scenarios such as when they have no desktop devices available, and required to maintain a development deadline.

Most participants reported that typing codes on a mobile phone was frustrating because it was slower and easier to make mistakes. The problem with typing on mobile phones is that all 26 letters will have to fit on the screen, causing individual keys to be relatively small in size. We have been unable to solve this issue, but horizon layouts may be adapted to reduce typing difficulty and chance to make mistakes.

All researchers have had years of experience with programming with various languages and IDEs, but none has used or even heard of mobile IDEs. This is an intriguing finding, because there exists coding applications on mobile platforms such as Koder and Code Editor[19] by Panic on iOS, and a few coding websites that can be opened on browsers on a phone such as tutorialspoint and w3schools, but none of which were popular or widely mentioned, at least not in universities. That led us to work with Software Engineering students and graduates to find out whether it was possible to code on a phone, provided a well-designed interface.

As the medium-fidelity prototype was on mobile, it was imperative that the testing be conducted on a mobile phone instead of an emulator on a desktop computer, which made remote distribution difficult. Apple does not offer free application tests via the App Store or third-party installations, nor does Figma provide a solution for remote participants to access the prototype on a mobile phone. However, we came to learn about a testing tool named, LookBack, but could not use that for time constraint. We believe it would be beneficial if we conducted usability tests as originally planned.

LIMITATIONS AND FUTURE WORK

One of the major limitations of the study was the fewer number of participants due to researchers' less access to professional mobile application developers and the imposition of social distancing. With the sample greatly confined to a single type of people, we cannot tell whether the study can be replicated. However, we do believe that this idea merits further study, and soon it will gain popularity over the traditional multi-device development environment. Finally, usability testing, conducted in a structured fashion, could provide the tools for quantifying the user-friendliness of the interface.

CONCLUSION

It is possible to have mobile development done solely on a mobile device through powerful IDE tools that have the same or similar functionalities as their desktop counterparts, and a well-designed interface to maximize the amount of code displayed at a time, and provide convenient operations. Our user study shows that people do not reject this concept for any particular reason other than the fact they have never attempted this method of development. It is only a matter of time before such a concept gains popularity, and eventually obsolete the traditional multi-device development environment.

ACKNOWLEDGMENTS

We are grateful to Professor Mark Hancock and TA Milad for their continuous support and guidance. We would also like to thank the reviewers for their meaningful feedback on our report. Finally, we thank all the participants for their time and feedback.

REFERENCES:

[1] "• Mobile OS market share 2019 | Statista." [Online]. Available: https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/. [Accessed: 27-Mar-2020].

[2] "Run apps on a hardware device | Android Developers." [Online]. Available: https://developer.android.com/studio/run/device. [Accessed: 10-Apr-2020].

[3] "Xcode - Apple Developer." [Online]. Available: https://developer.apple.com/xcode/. [Accessed: 26-Mar-2020].

[4] N. Z. B. Ayob, A. R. C. Hussin, and H. M. Dahlan, "Three layers design guideline for mobile application," *Proc. - 2009 Int. Conf. Inf. Manag. Eng. ICIME 2009*, pp. 427–431, 2009.

[5] M. Mehrban and M. Asif, "Challenges and Strategies in Mobile Phones Interface for elder people," 2010.

[6] J. Trimble *et al.*, "Design Method for Developing a Mobile," *Univ. Edinburgh, Edinburgh*, vol. 6, no. December, pp. 362–366, 2014.

[7] S. Glissmann, S. Smolnik, R. Schierholz, L. Kolbe, and W. Brenner, "Proposition of an m-business procedure model for the development of mobile user interfaces," in *International Conference on Mobile Business (ICMB'05)*, 2005, pp. 308–314.

[8] J. Mirkovic, H. Bryhni, and C. M. Ruland, "Designing User Friendly Mobile Application to Assist Cancer Patients in Illness Management Jelena," *Third Int. Conf. eHealth, Telemedicine, Soc. Med.*, vol. 84, no. 11, p. e36, 2011.

[9] C. Ryan and A. Gonsalves, "The Effect of Context and Application Type on Mobile Usability: An Empirical Study," in *Proceedings of the Twenty-Eighth Australasian Conference on Computer Science - Volume 38*, 2005, pp. 115–124.

[10] E. G. Nilsson, "Design guidelines for mobile applications," *Sintef Ict*, no. June, p. 73, 2008.

[11] E. Kangas and T. Kinnunen, "Applying user-centered design to mobile application development," *Commun. ACM*, vol. 48, no. 7, pp. 55–59, 2005.

[12] J. Gong and P. Tarasewich, "Guidelines for handheld mobile device interface design," 2004.

[13] H. Khalid, E. Shihab, M. Nagappan, and A. E. Hassan, "What do mobile app users complain about?," *IEEE Softw.*, vol. 32, no. 3, pp. 70–77, 2015.

[14] "Unreal Engine 4 Documentation | Unreal Engine Documentation." [Online]. Available: https://docs.unrealengine.com/en-US/en-US/Engine/ Blueprints/GettingStarted/index.html. [Accessed: 27-Mar-2020].

[15] N. Tillmann *et al.*, "The future of teaching programming is on mobile devices," in *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education*, 2012, pp. 156–161.

[16] J. M. Carroll, *Making use: scenario-based design of human-computer interactions*. MIT press, 2000.

[17] "TestFlight - Apple Developer." [Online]. Available: https://developer.apple.com/testflight/. [Accessed: 27-Mar-2020].

[18] "▶️ Prototyping in Figma - 630." [Online]. Available: https://www.figma.com/proto/Mv2wLq4Z35p61JD8SE4S5T/630?node-id=0%3A2&amp%3Bscaling=scale-down&scaling=min-zoom. [Accessed: 08-Apr-2020].

[19] "Code Editor." [Online]. Available: https://panic.com/code-editor/. [Accessed: 29-Mar-2020].