

**Czech University of Life Sciences Prague**

**Faculty of Economics and Management**

**Department of Information Technologies**



**Diploma Thesis**

**Development of a mobile application for university  
students**

**Kambarov Doszhan**

**© 2021 CULS Prague**

# CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

## DIPLOMA THESIS ASSIGNMENT

Bc. Doszhan Kambarov

Systems Engineering and Informatics  
Informatics

Thesis title

**Development of a mobile application for university students**

---

### Objectives of thesis

The main objective of the thesis is to analyse, design and deploy a mobile application for students of the Czech University of Life Sciences Prague.

Partial goals of the thesis are such as following:

- to make a literature overview of current mobile apps for university students and mobile development frameworks;
- to conduct user survey and develop the application prototype;
- to implement and test the mobile application and deploy it to a selected store.

### Methodology

The thesis will draw on an extensive literature review and secondary information desk research. In the practical part, a survey will be conducted to collect user requirements for the proposed application. Further, the methods of analysis, synthesis and deduction will be used to compare App Store and Play Market and mobile development frameworks. The application prototype will be developed by using software engineering methods such as User Centred Design, prototyping and usability testing.

**The proposed extent of the thesis**

60 – 80 pages

**Keywords**

Mobile application, development, deployment, mobile application market, Android, iOS, Play Market, Apple Store, operating system.

---

**Recommended information sources**

Apple Developer Documentation. In: Apple Developer [online]. One Apple Park Way, Cupertino, CA 95014, USA: Apple, 2009 [cit. 2018-09-02]. Dostupné z: <https://developer.apple.com/documentation/>

Google Play Developer API [online]. 1600 Amphitheatre Parkway, Mountain View, California 94043, United States: Google, 2018 [cit. 2018-09-09]. Dostupné z: [https://developers.google.com/android-publisher/getting\\_started](https://developers.google.com/android-publisher/getting_started)

HERMES, Dan. Learn Android Studio 3: efficient Android app development. 3rd edition. New York, NY: Springer Science Business Media, 2018. Wrox programmer to programmer. ISBN 978-148-4231-555.

HERMES, Dan. Xamarin Mobile Application Development: Cross-Platform C# and Xamarin.Forms Fundamentals. 3rd edition. California: Apress, [2015]. Wrox programmer to programmer. ISBN 978-148-4202-142.

MCWHERTER, Jeff a Scott GOWELL. Professional Mobile Application Development. 3rd edition. India: Wiley India Pvt., 2012. ISBN 9788126538249.

MUKHERJEA, Sougata. Mobile application development, usability, and security. Hershey PA: Information Science Reference, [2017]. ISBN 978-152-2509-455.

---

**Expected date of thesis defence**

2020/21 SS – FEM

**The Diploma Thesis Supervisor**

Ing. Miloš Ulman, Ph.D.

**Supervising department**

Department of Information Technologies

Electronic approval: 2. 10. 2018

**Ing. Jiří Vaněk, Ph.D.**

Head of department

Electronic approval: 19. 10. 2018

**Ing. Martin Pelikán, Ph.D.**

Dean

Prague on 31. 03. 2021

## **Declaration**

I declare that I have worked on my diploma thesis titled "Development of a mobile application for university students" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break copyrights of any their person.

In Prague on 31.03.2021

---

### **Acknowledgement**

I would like to thank my supervisor Ing. Miloš Ulman, Ph.D. for his advice and support during my work of this thesis.

# **Development of a mobile application for university students**

## **Abstract**

The main purpose of this thesis is to develop a mobile application for higher education institutions to improve their online services. The Object of the research is to make a literature overview of current mobile apps for university students and technologies used to develop this application.

The practical part of this work is to make a survey for the users of the future application to get a conception for development of user-friendly and full mobile application for students. According to the survey results, conception makes a user interface and chooses technology and tools for development.

Keywords: Mobile application, development, deployment, mobile application market, Android, iOS, Play Market, Apple Store, operating system.

# Vývoj mobilní aplikace pro studenty vysokých škol

## Abstrakt

Hlavním účelem této diplomové práce je vyvinout mobilní aplikaci pro vysokou školu za účelem zlepšení její online služeb. Cílem výzkumu bylo vytvořit aplikaci, k vývoji které byl zpracován literární přehled relevantních mobilních aplikací a technologií pro studenty vysokých škol.

V praktické části této práce je provedeno průzkum uživatelů budoucí aplikace s cílem získat koncepci uživatelsky přívětivé a plné mobilní aplikace pro studenty. Podle výsledků průzkumu byly vytvořené uživatelské rozhraní, vybraná technologie a nástroje pro vývoj mobilní aplikací.

**Klíčová slova:** Mobilní aplikace, vývoj, nasazení, trh mobilních aplikací, Android, iOS, Play Market, Apple Store, operační systém.

# Table of content

<b>1 Introduction</b> .....	<b>11</b>
<b>2 Objectives and Methodology</b> .....	<b>12</b>
2.1 Objectives.....	12
2.2 Methodology .....	12
<b>3 Literature Review</b> .....	<b>13</b>
3.1 Mobile application markets.....	13
3.1.1 Google Play.....	14
3.1.1.1 Google Play statistical numbers .....	15
3.1.1.2 Requirements for publishing application on Google Play.....	15
3.1.2 Apple Store .....	19
3.1.2.1 Apple Store statistical numbers .....	19
3.1.2.2 Requirements for publishing application on Apple Store .....	19
3.2 Mobile application types .....	21
3.2.1 Native mobile application.....	22
3.2.2 Hybrid mobile application .....	22
3.2.3 Web application .....	23
3.3 User Centered Design .....	25
3.3.1 What is UCD and how do recognize it? .....	25
3.3.2 Main features of UCD.....	26
3.4 Development tools review.....	26
3.4.1 Trello.....	27
3.4.2 Versioning.....	28
3.4.3 Project versioning .....	29
3.4.4 Expo Tools.....	30
3.5 Similar Existing Solutions .....	31
3.5.1 University of Otago .....	31
3.5.2 European University Cyprus.....	32
3.6 Literature review summary .....	34
<b>4 Practical Part</b> .....	<b>35</b>
4.1 Introduction .....	35
4.2 Survey .....	36
4.2.1 Summary.....	36
4.3 Defining development technologies.....	39
4.4 Requirements.....	39



4.4.1	Functional requirements .....	40
4.4.2	Non-Functional requirements .....	40
4.5	Use Case Scenarios .....	41
4.6	Activity diagrams .....	44
4.7	Wired models .....	45
4.8	Project description.....	52
4.8.1	Navigations .....	53
4.8.2	Screens .....	56
4.8.3	Components .....	57
4.8.4	Maps integration .....	59
4.9	Testing.....	60
4.9.1	Developers Testing .....	60
4.9.1.1	Use Case Scenario Testing .....	60
4.9.2	Use Case Scenario Testing Results.....	61
4.9.3	Use Case Scenario Testing Summary .....	63
4.10	Deployment .....	64
<b>5</b>	<b>Result and Discussion .....</b>	<b>66</b>
5.1	Installation.....	66
5.2	Accessing contents .....	66
5.3	Using campus map .....	66
5.4	Discussion .....	67
5.5	Recommendations .....	67
<b>6</b>	<b>Conclusion.....</b>	<b>69</b>
<b>7</b>	<b>References .....</b>	<b>70</b>
<b>8</b>	<b>Appendix.....</b>	<b>71</b>
8.1	Questions.....	71
8.2	Screenshots.....	73

## List of pictures

Figure 20 - Trello project management board .....	28
Figure 1 - University of Otago app screenshots .....	32
Figure 2 - European University Cyprus app screenshots.....	33
Figure 12 - Main content activity diagram .....	44
Figure 13 - Campus map activity diagram.....	45
Figure 14 - Main screen wired model .....	46
Figure 15 - Category screen wired model.....	47
Figure 16 - Sub-content screen wired model .....	48
Figure 17 - Campus buildings map wired model.....	49
Figure 18 - Campus parking map wired model .....	50
Figure 19 - Campus map object information modal wired model .....	51
Figure 21 - Project source-code structure .....	53
Figure 22 – Screens Structure .....	56
Figure 23 - CZU Student Guide application QR code.....	66
Figure 24 – Home Screenshot.....	73
Figure 25 – Practical information Screenshot.....	73
Figure 26 – Home Screenshot.....	74
Figure 27 – Practical information Screenshot.....	74
Figure 28 - Buildings map Screenshot.....	75
Figure 29 - Parking map Screenshot.....	75

## List of tables

Table 1 - Survey gender, age and student status results .....	36
Table 2 - Survey information searching device and smartphone OS .....	37
Table 3 - Survey internet access and receiving information preferred .....	37
Table 4 - Survey campus navigation preferred.....	38

# 1 Introduction

Using the information technology in the education system has become an integral part of life of all students at universities. All the study processes are reflected in university information systems: beginning with admission to the university and further accompaniment at all stages of the educational process, ending with the passing and testing of the final thesis and the last state examinations. The main task of the university's information systems is guidance, so that the student can track his progress and plan the learning process.

It is important to introduce the student to the university and its system. As a result, the student can easily navigate and easily follow the learning process. Universities certainly put a lot of effort in this introduction. The organization of excursions around the university, the arrangement of informal acquaintances with specific parts of the university are all accompanied by a set of printed brochures and documents, so that a student is provided by the necessary information.

The main motivation for this thesis was the simplification of the whole process for both the student and the university. The result of this work will be the application for mobile devices, which will replace all printed information for new university students. This will simplify the student's ability to easily access basic information about the university and the learning process. Alongside, it will be a great convenience for the university to provide the necessary information to new students without unnecessary efforts and resources to create printed brochures and documents.

## **2 Objectives and Methodology**

### **2.1 Objectives**

The main objective of the thesis is to analyze mobile application market and develop, design and deploy a mobile application for students of the Czech University of Life Sciences Prague.

Partial goals of the thesis are such as following:

- to make a literature overview of current mobile apps for university students and mobile development frameworks;
- to conduct user survey and develop the application prototype;
- to implement and test the mobile application and deploy it to a selected store.

### **2.2 Methodology**

The thesis will draw on an extensive literature review and secondary information desk research. In the practical part, a survey will be conducted to collect user requirements for the proposed application. Further, the methods of analysis, synthesis and deduction will be used to compare App Store and Play Market and mobile development frameworks. The application prototype will be developed by using software engineering methods such as User Centered Design, prototyping and usability testing.

## 3 Literature Review

### 3.1 Mobile application markets

According to the definition, a mobile application is an independent software product that is installed under the necessary operating system of a smartphone, tablet computer, or other mobile device. The most popular operating systems for mobile devices today are Android and iOS.

By purpose, apps can be divided into many different categories: games, news apps, for working with photos, text, and so on. It is quite difficult to provide a single classification, because some applications have a combination of several functions, which makes it impossible to classify them in a certain category.

The development of the mobile app market is facilitated, first, by the constant growth in the performance of mobile devices, and, secondly, by the increase in the availability of the Internet, including mobile, increasing the speed and stability of the connection. This is a crucial fact, because most mobile apps require a permanent network connection. Only in the Czech Republic according to Czech Statistical Office data of 2019, 97% of the Czech population uses a mobile phone and 70% of users prefer a smartphone. Mobile Internet is used more than twice by Czech citizens over the age of 16 compared to 2014, 65% of the total population versus 29 % (11).

Today, the mobile app market shows one of the most intensive growth rates among the smart product markets. According to "eMarketer" market research company, in 2019, more than 90% of the time spent on iOS and Android is occupied by mobile apps.

The main factors of the indicated dynamics are:

- The impact of social media;
- Development of modern mobile platforms and the emergence of new ones;
- Increased penetration of smartphones and tablet computers;
- Increasing the impact of mobile payment systems and mobile banking;
- Widespread penetration of high-speed Internet connection.

In this regard, several consequences can be identified:

- There has been a significant increase in mobile app downloads.
- Apps are increasingly integrated into social networks.

- increasing sales of smartphones and tablets, as well as related gadgets.
- Mobile games have become more profitable than games for portable consoles.
- Messengers are developing into a media platform.
- Significantly increased revenue from content sales.

Based on the given information, we can say about the current situation in the market:

- Marketplaces lower the entry threshold for new developers (Apple Swift, Android Studio).
- App ranking algorithms change, so that high-quality apps are noticed by users.
- Entertainment and games, messengers and commercial applications remain the most popular and profitable.
- Increase the role of Wearable Devices (Apple iWatch, Android Wear) and related applications.
- Strengthening the Freemium model for app monetization.
- New interface solutions, code simplification, shorter development times, and emphasis on cross-platform compatibility.

### **3.1.1 Google Play**

Google Play is the official online store from Google for Android apps and other digital media such as music, movies, and books. It is available on mobile devices and tablets running the Android operating system, supported by Chrome OS devices, and on the Internet. Users can access Google Play to view, purchase, and download software applications from Google and third- party developers.

Google Play Store comes pre-installed on Android devices running Google Mobile Services. It can be installed on compatible Chrome OS devices such as Chromebooks, Chromeboxes, and Chromebases. If the app is purchased on the Internet, the user can specify on which devices it should be installed. This includes devices such as Wear OS smartwatches and Android TVs. Purchases are stored in the cloud, so web and mobile accounts are synced, and content is available on all devices.

Malware protection is provided through Google Play Protect, a threat detection service that scans and verifies more than 50 billion apps every day. The service can also suspend applications that violate Google Play rules.

The Google Play was launched on March 6, 2012, combining the Android Market and other digital services such as Google Music and Google eBook store into a single offering. The online store is currently available in more than 190 countries.

#### 3.1.1.1 Google Play statistical numbers

To represent the extent of the impact of Google play and mobile apps on the global market, it is best to show the statistics. Below are the most important statistical performances of Google play during the period of 2019 and 2020.

**Total number of applications on Google Play.** According to the business data platform Statista, the number of available applications in Google Play by December 2020 is 3.14 million (10).

**Total number of applications downloaded daily.** Since there is a high demand for apps of all categories and niches, the number of downloads per day has also increased over the past two years. This is also because the market is experiencing rapid growth in demand for smartphones. In the Sensor Tower report, which studies Google Play statistics for 2019-2020, they estimated the number of downloads per day at 250 million (9).

**Total number of applications with and without ratings.** Google keeps an account of the applications with ratings or without ratings for users of the online store to determine whether to use an application. Google has calculated that there are 2 156 187 applications which have ratings and 858 200 apps without ratings.

#### 3.1.1.2 Requirements for publishing application on Google Play

Before you publish an application on Google Play and distribute it to users, you need to prepare it, test it, and prepare promotional materials. Google provides developers with detailed documentation (2) in which step-by-step actions are described.

This documentation helps the developer understand the publishing process and prepare for a successful product launch in Google Play. It lists some tasks that need to be completed before publishing an app to Google Play, such as creating a ready-to-release Android application package (APK), understanding the application requirements, and creating a

product page and graphics resources for the application. In order to start the publishing process, you must register a developer account to which the published app will be linked.

The following is a list and description of the requirements for publishing the mobile application.

**Understand the publishing process.** Before proceeding with the steps described in this list, you should read and understand the General publishing workflow and how it is set up. The developer will need to prepare the app for release using a process that is common to all Android applications.

**Understand Google Play policies.** The developer must make sure that he or she understands and follows the rules of the Google Play program adopted during registration. Google Play actively applies these rules, and any violations may lead to the suspension of the application or, in the case of repeated violations, to the closure of the developer account.

**Test for Core App Quality.** It is important to make sure that the app meets the basic quality requirements for all Android apps on all the devices it is targeting. You can check the quality of an application by setting up a test environment and testing it against a short set of basic quality criteria. Detailed quality requirements will be specified in the practical part of this work.

**Determine the content rating.** Google Play requires you to set a content rating for an app that informs Google Play users about its maturity level. Before publishing, you need to confirm which rating level to use. Available content rating levels are:

- Everyone
- Low maturity
- Medium maturity
- High maturity

**Determine country distribution.** Google Play allows you to control, which countries and territories, the app is distributed to. It is important to determine the exact distribution by country, because this may affect the following:

- Need for localized resources in the app
- Need for a localized app description in the developer console



- Legal requirements for the application that may be specific to certain countries
- Support for time zones, local prices, and so on.

**Confirm the applications overall size.** The overall size of an app may affect its design and how it is published on Google Play. Currently, the maximum size of an APK published in Google Play is 50 MB. If the app exceeds this size or you need to offer an additional download, you can use the APK extension files, which Google Play will place for free on its server infrastructure and automatically process the download to your devices. Possible to use up to two APK extension files, each up to 2 GB in size, for each.

**Confirm app compatibility ranges.** It is important to make sure that the app is designed to work correctly on the Android platform versions and device screen sizes that need to be configured.

**Decide on free or priced.** Free apps can be downloaded by any Android user in Google Play. Paid apps can only be downloaded by users who have registered a payment form in Google Play, such as a credit card or direct carrier payment.

**Consider In-app Billing.** If the app requires monetization by selling digital content, you must enable and configure a billing service. The in-app billing service allows you to sell one-time purchases and subscriptions within the app.

**Set prices for your apps.** If the app is rated or there are products sold in the app, Google Play allows you to set prices for your products in different currencies for users in markets around the world. You can set prices individually in different currencies, so you can flexibly adjust the price according to market conditions and exchange rates.

**Start localization early.** If the program is intended to be on several localizations, it is necessary to prepare the minimum three aspects:

- Localization of strings, images, and other resources in the app.
- Localize information about your app store in Google Play.
- Localization of the app's graphics resources, screenshots, and videos that accompany an ad in the store.

**Prepare promotional graphics.** There is a need to provide a variety of high-quality graphic assets to showcase the app. After publishing, these graphics appear on the

detail page of the product in the list of stores and search results, as well as in other places. These graphical resources are key elements of a successful product detail page that attracts and engages users. Screenshots and videos are also very important because they show what your app looks like and how it is used.

**Build the release ready APK.** The process of preparing a ready-to-release APK is the same for all applications, regardless of how they are distributed. Typically, this process includes basic code cleaning and optimization, building and signing with the release key, and final testing.

**Complete the product details.** You need to fill in information about the app in Google Play, which is displayed to users on the app details page. This information will help users decide if they need this app.

**Final checks and publishing.** When the app is ready to publish, there must be an action of logging in to the developer console and performing a final check of the following points:

- Developer profile, if it contains the correct information
- Version of the uploaded application
- All application details are filled in, including all graphics resources, screenshots, videos, localized descriptions etc.
- If the application is installed for free or paid.
- If the distribution country and currency are set correctly, if there are any prices.
- Compatibility of devices to which the application was developed.
- The correct link to the website and corrections of the email address of the support service.
- The application and its content cannot violate the content policy rules.

**Support users after launch.** After publishing an app or updating an app, it is critical to support your customers. Prompt and courteous support can provide a better experience for users, leading to better ratings and more positive reviews of the app.

### 3.1.2 Apple Store

The Apple App Store is a digital distribution platform where customers can buy and download digital software and applications. Applications purchased from the App Store are stored in iCloud for easy access from any device that is signed in. Apple claims to have a trademark on the term “online app store”, but this term is used to refer to any platform where apps are sold.

Apple opened the App store on July 10, 2008, the day before the launch of the iPhone 3G networks. Apple is first app Store for iOS but was later expanded to provide apps for Mac computers with the App Store for Mac OS in early 2011. One of the many significant changes to the App store came in 2014, when the focus of European regulators forced the company to change apps listed as “free” to “get” to reflect that some of them included in-app purchases. This practice of labelling has become a standard in 2017.

#### 3.1.2.1 Apple Store statistical numbers

Below are key figures showing the results of the research on business data platform “Statista” about the Apple App Store.

**Total number of applications on Apple Store.** By the end of 2019, Apple App Store was the second-largest app store with almost 1.84 million available applications for iOS devices.

**Total number of downloaded apps for 2019.** According to the data from 2019, users downloaded 204 billion apps on the Apple store platform, compared to 140.7 billion app downloads in 2016.

**Total Apple App Store iOS developer payout.** The total revenue of the Apple App Store for mobile app developers as of January 2019 is 155 billion US dollars. In 2018, this figure was 120 billion US dollars.

#### 3.1.2.2 Requirements for publishing application on Apple Store

In order to publish applications on the Apple Store, you first need to register an account for the developer. For this registration, you must pay a certain amount and get a confirmation from Apple, which is processed within two weeks. Only then you can start publishing the app on the Apple Store.

The Apple Store, as well as Google Play, has detailed documentation about all aspects of application development in order to publish it there (1). To successfully publish an application on the Apple Store, you must go through a large list of requirements and recommendations. If any of the requirements are not met, the request to publish the app will be rejected.

The following will describe the latest major Apple publishing requirements, which are updated over time. In order to make the application available on the Apple Store, you need to keep an eye on the requirement updates.

**Safety.** For users to be sure that the downloaded application is safe, the following criteria must be checked:

- The application does not contain objectionable content.
- The content used complies with all intellectual property conditions.
- If the application will be in the Kids category you need to meet the requirements of the app for children.
- Using the app does not cause physical harm to the user.
- Providing the information about the developer to contact support in case of problems with the app.
- Protect the user's data from misuse.

**Performance.** For the app to perform its main task efficiently and smoothly, it must meet the following performance requirements described in detail in the Apple Store documentation:

- App completeness,
- Beta testing,
- Accurate metadata,
- Hardware compatibility,
- Software requirements,

**Business.** If the app provides paid content or its purchase is paid, then you need to check the compliance of the following points:

- If the in-app purchase methods are not obvious, the explanation must be provided in detail in the application metadata and review.

- Adequacy of the app price or in-app purchase.
- Check the compliance of payment and subscription methods.
- Matching specific access to paid services and products in application.
- If the app provides advertising, then its limitations and appropriateness.

**Design.** Users of any smartphone when using applications on it expect high quality and are easy and intuitive to use. The App Store takes these criteria very seriously. No matter how great the app design is, it's important for the App Store that it meets their minimum requirements. If there is the slightest discrepancy, your site can be removed from the platform. Once your app has been approved, you should monitor for updates on demand to the app and update it if they change.

Minimum design requirements for the App Store, which are regularly updated and can be found in more detail in the official documentation for developers.

This documentation highlights the following points:

- Do not copy other application,
- Minimum functionality,
- Do not create multiple bundles of the same app,
- If the application hosts or containing extensions,
- Correct usage of Apple Sites and Services,
- Alternate applications icons,
- If the application contains HTML5 Games, Bots, etc.
- One of the login methods have to be Sign In Apple, if the application will stream games.

## 3.2 Mobile application types

As the number of mobile devices increases, so does the need for app development. Hundreds of mobile apps are created every day to provide more comfortable interaction with a mobile device. There are several types of applications that developers use:

- Native mobile applications
- Mobile web applications
- Hybrid mobile applications

### **3.2.1 Native mobile application**

These applications are called native because they are written in the native programming language for a particular platform. For Android, this language is Java or Kotlin, while for iOS objective-C or Swift.

Native applications are located on the device itself, which can be accessed by clicking on the icon. They are installed through the app store (Google Play on Android, App Store on iOS, etc.). They are designed for a specific platform and they can use all the features of the device – camera, GPS sensor, accelerometer, compass, contact list and everything else. They can also recognize standard gestures that are pre-installed by the operating system, or completely new gestures that are used in a specific application.

Since native applications are optimized for a specific OS, they fit seamlessly into any smartphone, featuring high speed and performance.

Native apps can access the device's notification system, and depending on the purpose of the native app, it can completely or partially run without an Internet connection.

Advantages of native apps:

- speed and performance;
- high security;
- advanced interface;
- as much functionality as possible;
- ability to work without the Internet;
- convenience for the end user.

Disadvantages of native apps:

- platform coverage;
- long development time;
- relatively high cost of development;
- the need to release an update for cosmetic purposes.

### **3.2.2 Hybrid mobile application**

Hybrid applications are a combination of web and native applications. This means that they are cross-platform and have access to smartphone functionality. These apps can only be downloaded from markets like Google Play and the App Store. However, they

have the option to update information offline, and they require an Internet connection. Without the availability of the latest web functions, they simply do not work.

Among many companies, the choice most often falls on the development of a hybrid application. This is explained by the fact that hybrid applications can combine the advantages of native with technological relevance, which is provided by the latest web technologies. However, unlike native ones, the cost of creating a hybrid is much lower, and its speed is higher. The affinity of hybrid applications with web applications, in turn, bears fruit in the form of the fact that they can be easily and quickly adjusted. In other words, developers don't have to replace the app in the store in order to fix errors in the previous version, as is the case with native apps.

The development of a hybrid application is also considered promising because it involves creating it for two platforms at once. As a result, this eliminates the headache associated with separate application development for each OS and it is a crucial factor.

Among other things, you should note that the quality and capabilities of hybrid applications depend primarily on the framework that the developer uses. It is also worth paying attention to the factors that make hybrid applications the preferred option over the rest.

Advantages of hybrid applications:

- cost and speed of development;
- a small number of developers;
- cross-platform;
- offline update option.

Disadvantages of hybrid applications:

- incorrect operation when there is no Internet connection;
- average speed of work against the background of native;
- minimalism in terms of visual elements.

### **3.2.3 Web application**

In fact, mobile web apps are not apps per se. After all, the fact is that a web application, in its essence, is a site that is adapted and optimized for any smartphone. In order to use it, it is enough to have a browser on the device, to know its address and to

have an Internet connection, thanks to which information is updated in this type of application.

When launching mobile web applications, the user performs all the actions which they perform when they go to any website. The user gets the opportunity to "install" them on their desktop by creating a bookmark of the website page as well.

Web applications are cross-platform, meaning they can function independently of the device's platform. The trump card up their sleeve is that they don't use its software. Because they are a mobile version of the site with advanced interactive content, web applications do not take up precious space in the smartphone's memory.

Web applications became widely popular at a time when HTML5 began to develop and people realized that they could access many of the features of native applications simply by accessing a website through a regular browser. Today, it is difficult to say exactly where the clear border between web applications and regular web pages is located, since the HTML5 functionality is growing every day and more and more sites are using it.

Web applications are developed using tools and frameworks that have become traditional. As a result, the process of their development has recently accelerated significantly.

The main drawback of a web application is the inability to work with them without the Internet. The second main disadvantage is their performance, which is at an average level, in comparison with other types of applications. Moreover, it depends on the Internet connection capabilities of the service provider.

Advantages of mobile web applications:

- full platform coverage;
- simple and fast development process;
- a small number of competent developers;
- no need to download from the app store.

Disadvantages of mobile web applications:

- required Internet connection;
- poor app interface;
- unable to send push notifications;
- performance and speed;
- unsatisfactory security level.



### 3.3 User Centered Design

The first attempts to design with the user's needs in mind were noticed in the 1980s, when designers and developers began to focus not only on the technical part of software development. It has become important for users that the technology is not only powerful and stable, but also easy to understand and use. The market has slowly but surely transformed from a person who adapts to the system to a system that adapts to the person. One of the best examples of that time and currently is Apple. Steve jobs once said: "You've got to start with the customer experience and work back toward the technology — not the other way around" - "You must begin with customer needs and return to the technology, not vice versa".

#### 3.3.1 What is UCD and how do recognize it?

User centered design is a design, which considers the needs at each stage of design. User-oriented designers understand the people they design for, come up with interesting solutions, and adapt to their needs (7). Many companies have incorporated UCD into their organizational strategy. Among them are IKEA, Lego, Google, Facebook, Microsoft and Apple. They bet on the emotional relationship between the product and the consumer.

The UCD process includes 6 steps:

1. **Collecting information about user needs.** Identify the people who will use the product, what they will use it for, and the conditions under which they will use it. Designers should try to adapt their products to the people they are intended for.
2. **Collecting technical information about the product.** Once you have identified the users' needs, it is necessary to understand how technically feasible it is and how to make the decision financially feasible. It is extremely important that design solutions increase the number of regular users and attract investment.
3. **Development of unique design solutions.** In this step, designers generate ideas, test them, and improve them. There may be too many ideas: some will be too sophisticated to work on others will be a mistake not to try to implement them.

4. **Usability testing with real users.** The sooner the designer shows the solution to actual users and gets feedback from them, the easier it will be to look at the design from the user's point of view. Evaluation is an integral part of the user-centered design process, as is testing in software development.
5. **Implementation of the developed concept.** At the implementation stage, the designer brings his solution to life. The person looks at how it functions in the real environment.
6. **Refinement as users ' needs evolve and change.** User-oriented design is always adaptive. It changes depending on new user needs.

### 3.3.2 Main features of UCD

User centered design is a team effort between designers, developers, testers, and other professionals when needed. Therefore, the first and most important characteristic of UCD is the so-called “empathy”, which means the ability to put yourself in the user's position. The second characteristic of user-oriented design is interactivity. Designers design, test, and Refine based on surveys, interviews, and user testing results, and this cycle repeats several times. A third characteristic is the interdisciplinary approach. Designers, psychologists, ethnographers, and other specialists, who know the end user well, can work on UCD (7).

### 3.4 Development tools review

For the convenience and quality of application development, the following tools were used, which will be described in more detail below:

- Trello for task management.
- GitHub for versioning.
- Microsoft Visual Studio IDE for programming and coding.
- Expo Tools to run applications on Android during development, debugging and building applications of the project

### 3.4.1 Trello

Trello is one of the most popular online project management systems, which is in demand among small companies and startups. It allows you to effectively organize the work according to the Japanese methodology of Kanban boards.

It was created by Fog Creek Software in 2011 based on MongoDB, Backbone.js and Node.js. The main advantages that allowed Trello to achieve popularity are:

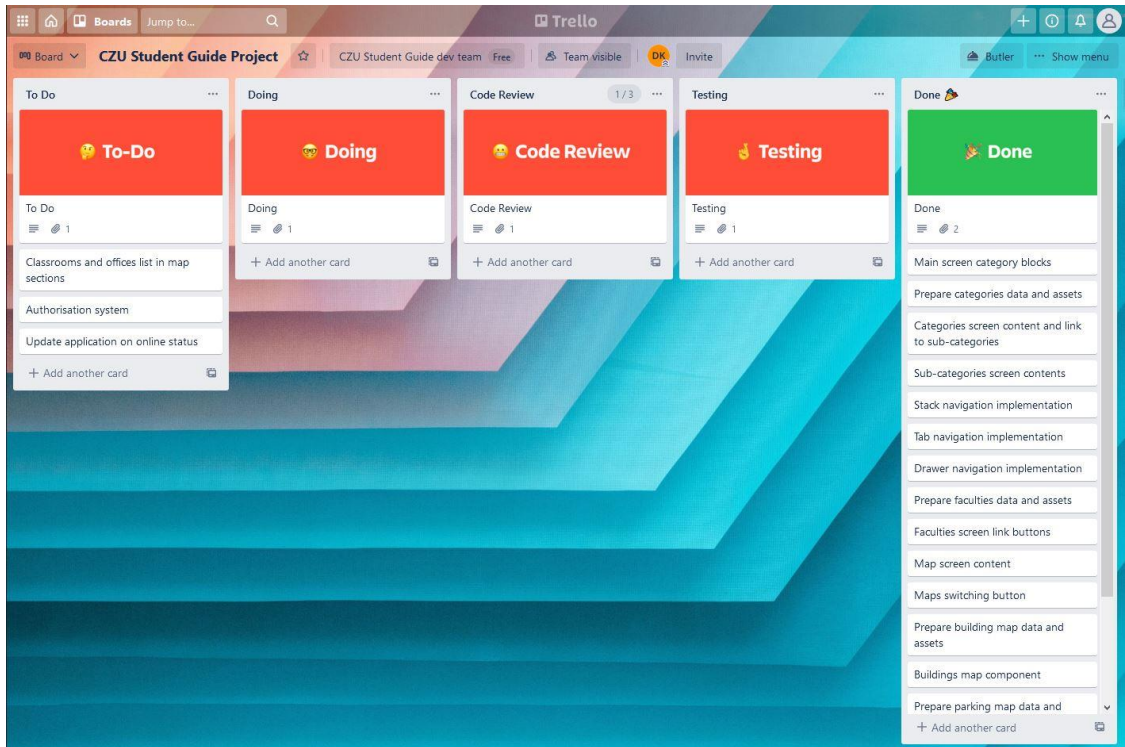
- simple interface;
- almost unlimited free access;
- easy to use and easy to integrate with other popular tools for online work.

For our application it will be used mainly for that to create tasks. Each task went through a specific cycle of processes according next list:

- To-Do – tasks to implementation.
- Doing – tasks in the implementation process or coding.
- Code Review – tasks to check code quality.
- Testing – tasks to test implemented features.
- Done – tasks successfully tested and implemented.

The simple Kanban board interface makes it easy to transfer tasks from one status to another, both forward and backward.

Below in **Figure 20**, you can see in general at what stage the application development is and what else needs to be implemented in the future.



**Figure 1 - Trello project management board**

### 3.4.2 Versioning

For versioning of source-code I used Git. Git is a kind of VCS (Version Control System). And VCS is a program for working with constantly changing information. Such software can store multiple versions of the same file (document) and revert to an earlier state (version).

For example, the code is written by several programmers. They work in different places, from different computers, at different times. VCS saves all changes made by all programmers, and at any time you can go back to the version created three days or a month ago.

What VCS are there exists:

- Git itself;
- SVN;
- Mercurial;
- CVS (Concurrent Versions System);
- Team Foundation Server.

All of them are divided into two types:

- distributed (Git, Mercurial) – changes are stored in the local storage of computers and are synchronized with other computers from there;
- centralized (CVS, SVN) – all data is stored on a central server, and from there each local computer receives updated data.

Git refers to distributed systems, so it does not depend on the central server where the files are stored. Git stores the data in a local repository. What is a Git repository? This is a directory on the hard disk of the programmer's working computer. For greater stability and faster synchronization of different versions of the project, the local repository is stored online in special services: GitHub, Gitlab, Bitbucket.

As a centralized store for versioning in our project the GitHub service was used.

### **3.4.3 Project versioning**

An IDE (integrated development environment) is not just a text editor. While text editors for code, such as Sublime or Atom, offer many convenient features, such as syntax highlighting, a customizable interface, and advanced navigation tools, they only allow you to write code. To create functioning applications, at least you need a compiler and debugger.

The IDE (integrated development environment) includes these components, as well as several others. Some of them come with additional tools for automating, testing, and visualizing the development process. The term "integrated development environment" means that everything is provided that is necessary to turn the code into functioning applications.

For development of our application we used Microsoft Visual Studio. It is an integrated development An IDE (integrated development environment) is not just a text editor. While text editors for code, such as Sublime or Atom, offer many convenient features, such as syntax highlighting, a customizable interface, and advanced navigation tools, they only allow you to write code. To create functioning applications, at least you need a compiler and debugger.

The IDE (integrated development environment) includes these components, as well as several others. Some of them come with additional tools for automating, testing, and visualizing the development process. The term "integrated development environment"

means that everything is provided that is necessary to turn the code into functioning applications.

For development of our application we used Microsoft Visual Studio. It is an integrated development environment. Many versions of this IDE can create all types of programs, ranging from web applications to mobile applications, video games. This line of software includes many tools for testing compatibility. Thanks to its flexibility, Visual Studio is a great tool for students and professionals alike.

Supported programming languages: Ajax, ASP.NET, DHTML, JavaScript, JScript, Visual Basic, Visual C#, Visual C++, Visual F#, XAML, and others. Since our application is developed based on the React Native Framework, the Java Script programming language supported by this IDE is important for us.

Features of Visual Studio IDE is next:

- A huge library of extensions that is constantly growing;
- IntelliSense.
- Customizable panel and dock able windows.
- Simple workflow and file hierarchy.
- Real-time performance monitoring statistics.
- Automation tools.
- Easy refactoring and code snippet insertion.
- Split screen support.

Error list that makes debugging easier.

#### **3.4.4 Expo Tools**

Expo is a set of tools that simplify the creation of React Native applications. With Expo, developers can create React Native apps without having to worry about installing and configuring software dependencies such as Android Studio, XCode, and other tools that are needed to develop and run React Native applications.

Expo is a framework for fast-growing native React applications. It's like Laravel or Symphony for PHP developers, or Ruby on Rails for Ruby developers. Expo provides a layer on top of the React Native API commands to make them easier to use and manage. It also provides tools that simplify the initial creation and testing of React Native applications. Finally, it provides user interface components and services that are usually

only available when installing third-party native React components. All of them are available through the Expo SDK.

Before using this tool, it is necessary to know the requirements of the application being developed, in order to consider some of the limitations of the Expo tools (8):

- **Expo apps do not support background code execution.** This means that you cannot, for example, run code that listens for location changes when the app is closed.
- **Expo apps are limited to the native APIs that the Expo SDK supports.** This means that if your application has very specific tasks, such as exchanging data with a Bluetooth device, the only option to implement this functionality is to use a simple React Native, or write your own code using the Expo Kit library.
- **Expo binds you to the use of its tools.** This means that you cannot just install and use most of the great tools available for React Native development, such as command-line tools, helper libraries, and UI frameworks. But the good news is that the Expo SDK is compatible with simple React Native applications, so you won't have any problems if you extract the application from Expo.
- **Offline executables of Expo applications can only be built if there is a network connection.** Expo provides command-line tools called Expo CLI. This allows developers to run the project build process on the Expo servers. After the build is completed, a URL link will be available to download the .apk or .ipa file.

### 3.5 Similar Existing Solutions

The aim of this thesis is to develop a mobile application for new coming students. It is important to start with the examination of several similar existing solutions. There are a few mobile applications on AppStore and Google Play, which aim to help students to get guidance information about university.

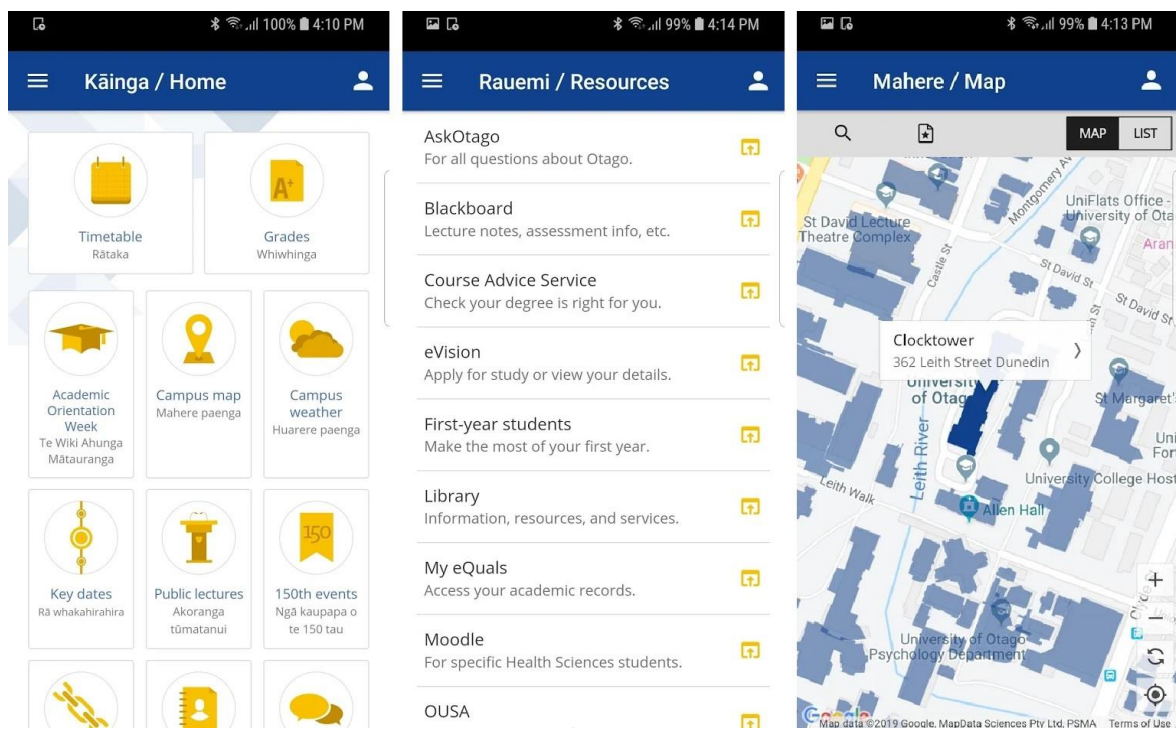
#### 3.5.1 University of Otago

The first similar application on Google Play is for University of Otago students in New Zealand. The application has a four-star rating from most of the users and more than 5000 downloads on Google Play.

Main features of application are the next points:

- Personalized study timetables,
- Quick access to personal grades,
- An interactive campus map,
- Live campus weather,
- Daily event listings during Academic Orientation Week,
- Easy access to key dates and upcoming public lectures.

This app is designed for already enrolled students. A permanent Internet connection is required to run the app. In the section where you can find any detailed description of the University, faculty or specific curriculum, the full information is a link to the University's web pages. Interactive campus maps integrated with standard Google Maps functionality and a list of all university buildings pointed on it.



**Figure 2 - University of Otago app screenshots**

### 3.5.2 European University Cyprus

Another mobile application for android devices is for EUC students. The application has a four-star rating from five users and more than 1000 downloads on Google Play.



According to the description, the main features are listed as:

- About the University,
- Programs of Study,
- Employability and Career,
- Schedule of Classes,
- Academic Calendar,
- Student Services,
- Emergency.

Application works offline and automatically updates if the device is connected to the internet. However, most of the content is not available without internet connection and implemented as a simple link which goes to university web pages. The campus map is implemented in the simplest way with standard Google Maps functionality and list of buildings.

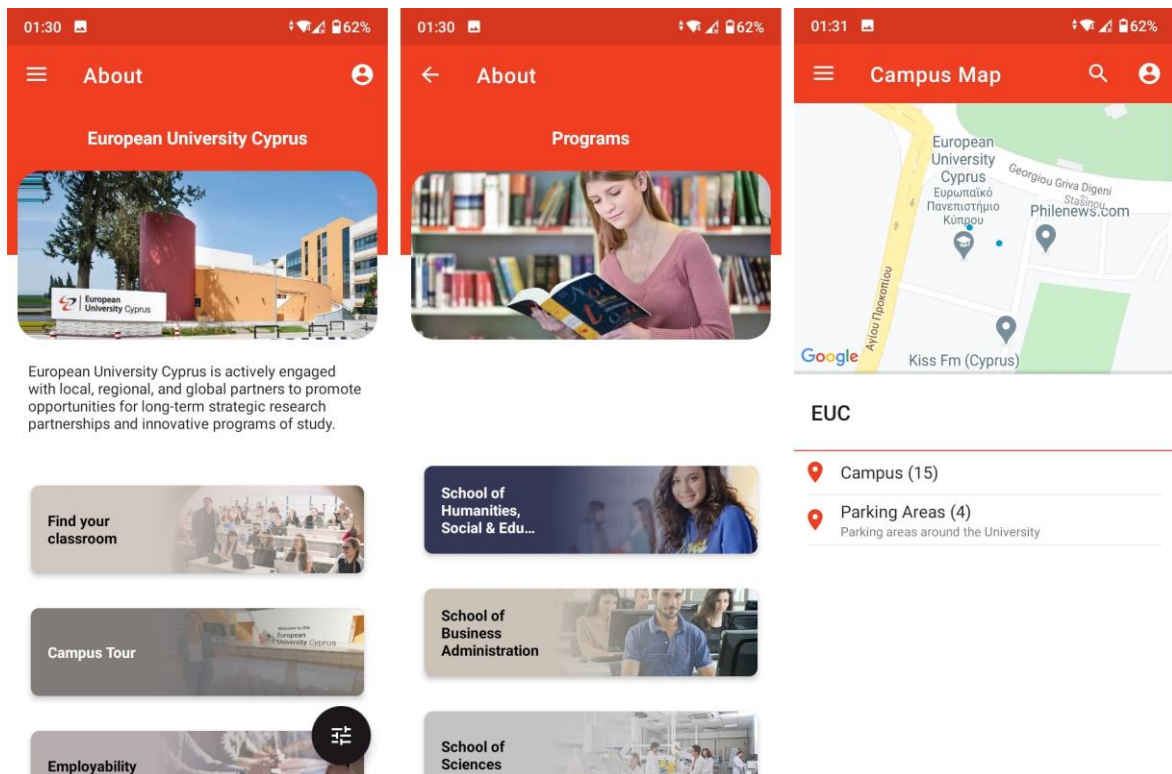


Figure 3 - European University Cyprus app screenshots

### **3.6 Literature review summary**

In the theoretical part of the work, first, we studied the mobile app market and provided statistical data to show how fast these markets are developing. We also studied the main requirements for the application being developed, which will help guide the development process as it is necessary for the main requirements of each mobile application markets.

After that, we made review of mobile application types and some of modern tools that will help at different stages of our application development in practical part of this work.

In the last chapters we gave an example of existing applications and review their features in order to understand which of them could be highly useful for our application users. Because we will use user centered designing methodology, what was also reviewed in theoretical part.

Now, armed with all needed information we can start working on designing and developing of our application for students.

## 4 Practical Part

### 4.1 Introduction

The main goal of the practical part of this work is to develop an application for university students. This application will help both the student to get and the university to provide detailed information about the university.

The developed application will contain all the information that the university now provides as a bookable brochure. The mobile application will not only allow you to be a static medium of information but will also allow you to expand the functionality of obtaining information by students. One such possibility would be to use navigation on a map to locate specific buildings on campus. Also, all links in text to external websites could be opened by clicking on the default browser on a mobile device. Contacts as email and phone numbers are also clickable. On email clicking will trigger an action on the mobile device to send new email with filled email by default mailing app. In case of clicking on phone numbers it will open a phone app with a filled phone number.

In the future, the application can be expanded in its functionality for more personal use by students. For example, add an individual lesson schedule, show academic achievement, take personal study notes, and many other opportunities as needed for a student and university.

In order to achieve the main goal of the practical part, the following works will be carried out:

- Conduct end-user survey to determine application development technology
- Define basic functional and non-functional requirements
- Analyze application logic and functionality in detail with Use Case Analysis, Activity Diagrams, Wireframes, Explaining of navigation structure
- Split and define online and offline parts of the application
- Description of minimum application programming interface (API) capabilities for online functionality
- Detailed description of application components with code examples
- Testing of final application
- Description of deploying process of it to chosen market platform

## 4.2 Survey

The purpose of this survey is to learn information from a future or already enrolled student, which would decide what technologies to use to develop a mobile application. The survey was designed for different age, gender, social categories. Google Forms were used to create the survey. Social media groups and education forums were used to disseminate the survey.

Link to survey form: <https://forms.gle/d7Z3oKmc7fPbxt8Z9>

All questions were required except for questions about age and gender. Most of the questions had “other” option, so that the user could add his own answer option.

### 4.2.1 Summary

In total, 89 people participated in the survey, which is a sufficient indicator to summarize statistics and decide on the choice of technology for developing the application. Below are the results of the survey in the tables.

<b>Gender ratio</b>	
Female	42.3%
Male	57.7%
<b>Age ratio</b>	
10 – 18 years old	6.7%
19 – 25 years old	49.4%
26 – 30 years old	32.6%
Older than 30 years old	11.2%
<b>Student status</b>	
Student in secondary school	5.6%
Finished secondary school	13.5%
University student	65.2%
Other	15.7%

**Table 1 - Survey gender, age and student status results**

<b>Device mostly used to search an information</b>	
PC or laptop	48.8%
Tablet	16.3%
Smartphone	71.3%
<b>Operating system on respondent's smartphone</b>	
Android	55.1%
iOS (Apple)	41.6%
Windows mobile	3.4%

**Table 2 - Survey information searching device and smartphone OS**

<b>If always have access to the internet on their smartphones</b>	
Yes	56.2%
No	27%
Maybe	16.9%
<b>If always had access to open internet in visited universities</b>	
Yes	55%
No	26.2%
Maybe	18.8%
<b>How respondents prefer to receive an information about universities</b>	
University website	50.6%
University mobile application	41.6%
Hard printed brochures	6.7%
Not interested in this information	1.1%

**Table 3 - Survey internet access and receiving information preferred**

<b>How respondents would navigate in a large university campus</b>	
Physical indicators on campus	33.7%
An maps application on smartphone	41.6%
Hard printed map of campus	7.9%
On the advice of other people	15.7%
Other	9%

**Table 4 - Survey campus navigation preferred**

According to the results of the survey, the following conclusions can be drawn:

- The gender ratio is approximately equal, 51 women participated in the survey, 36 men and 2 abstained.
- The age ratio of respondents showed that people aged 19 to 25 years 49.4% and 26 to 30 years 32.6% took the most part.
- Most of the respondents are university students, this is 65.2% of the participants
- 71.9% of participants use smartphones to search for certain information.
- 55.1% use smartphones on the Android operating system and 41.6% use smartphones with the operating system iOS.
- 56.2% of survey participants have constant access to the Internet, and 27% do not always have constant access to the Internet on their smartphones.
- 61.8% had access to the open Internet at the universities they visited.
- 50.6% of respondents prefer to receive information on the university's websites and 41.6% would be happy to use a mobile application to obtain information about the university, the rest prefer paper media.
- To question what method of navigation you would prefer on large campuses of the university, respondents answered that 41.6% would use the map application on a smartphone, 33.7% would be oriented on physical markings on campus, 15.7% on the advice of other people and 7.9% on printed campus maps.

### **4.3 Defining development technologies**

Based on the results of the theoretical analysis and the results of the survey, it was decided to use the technology of a hybrid mobile application because the use of both operating systems is android and iOS very high.

The main reasons for this are:

- First, it will allow you to simultaneously develop an application for the Android and iOS operating systems.
- The second is to save time and resources for developers.
- Third, the application's functionality does not require the capabilities of the native mobile app.
- The web mobile application provides a permanent Internet connection, and our app can work both online and offline.

I decided to use React Native Framework as a main technology for developing the hybrid application.

The main reasons why I chose this technology are as follows:

- To use the React Native framework, you need to know the JavaScript programming language, which I have been using professionally.
- This technology is widely used and supported by large IT companies such as Tesla, Facebook, Salesforce, Uber Eats, Walmart and many others.
- Detailed and well written documentation.

For application development, this work will focus more on the android operating system and placing the app on the Google Play platform.

### **4.4 Requirements**

This application is being developed to help the university provide all basic information about the university and the educational process in a more accessible form, thereby giving another opportunity to obtain information for its applicants and students.

The development will be carried out by one developer and will take several months of work on it. The application development part will not include the following, but will be added in future updates to the application:

- Content Management System all data will be static on mobile device
- Content Update on online mode
- Integration Sign Up with Facebook or Google accounts
- List of all offices, rooms, classes to find on map section and to navigate to the map.

#### 4.4.1 Functional requirements

According to the high workload, the following functional requirements were formulated:

1. **Content Navigation** - users can navigate the content structure, which is divided into sections and subsections.
2. **Parking map** - users can see on the Google map location of university parking places and different entrances and by clicking on markers and colored territories can get additional information.
3. **Campus map** - users can see on the Google map location of university buildings and by clicking on markers and colored buildings can get additional information.
4. **Clickable URL links** - users can click on the highlighted link in the text and open it in the default mobile device browser app.
5. **Clickable emails** – users can click on any email address in app content and it will open sending new email with filled to send email address by default mobile device emailing app.
6. **Clickable phone numbers** - users can click on any phone number in app content and it will open the default mobile device phone app filled to call the phone number.

#### 4.4.2 Non-Functional requirements

1. **Mobile Application API** - application will work with a simple server hosted API to use online features of the application.
2. **Android platform** - the application will work on smartphones and tablets and Android operating system with version 4.1 (API 16) or newer.
3. **Usability** – the application must comply with Android or iOS platform standards and follow the basic usability principles.



## 4.5 Use Case Scenarios

Use Case Scenarios is mainly used to cover the application functionality with a detailed scenario of the action when using the application. For our application, we have defined 6 functional features. For the analysis to be successful, we need to make at least a one-use case scenario for each application functionality.

Following will be Use Cases for each functionality of our application:

**UC1 Get Information about the CZU shop.** Users in this section can find information about the CZU shop.

1. On the main screen, the user clicks on the block labeled “Study information”.
2. Application shows a screen with buttons that allow the user to go to other sub-pages, the page can be scrolled down.
3. User scrolls down the screen and clicks on the button labeled “CZU shop”.
4. Application shows a screen with information about the CZU shop.
5. After reading the information, the user can return to the main screen by clicking on the home button in the bottom navigation.

**UC2 Get parking information on map.** On this screen, users can view the location of car entrances and parking areas.

1. User clicks on the middle map button in the bottom navigation of the app.
2. Application shows a page with a map of the university buildings.
3. User clicks on the car icon in the right part of the top navigation.
4. Application shows a screen with a map of the location of car entrances and parking areas.
5. User swipes on the map in a certain area and clicks on the marker icon or the area highlighted in different colors on the map.
6. Application shows information in a modal window about the clicked area or marker icon on the map.
7. User clicks on the “Ok” button in the modal window.
8. Application closes the modal window and shows campus parking map.

**UC3 Get buildings information on map.** On this screen, users can view the location of the university building.

1. User clicks on the middle campus map button in the bottom navigation.
2. Application shows a screen with a map of the university buildings.
3. User swipes on the map in a certain area and clicks on the marker icon or the area highlighted in different colors on the map.
4. Application shows information in the modal window about the clicked area or marker icon on the map.
5. User clicks on the “Ok” button in the modal window.
6. Application closes the modal window and shows a map of the university building.

**UC4 Find and open faculty home page on browser.** All URL links in the application content are highlighted in bold and in a special color. When you click on this URL link, the application opens it in the default mobile device browser application.

1. User on the main screen clicks on the block with the label “Faculties & the Institute”.
2. Application shows a screen with a list of buttons labeled with faculty name.
3. User clicks on the button labeled “Faculty of Economics and Management”.
4. Application shows screen with the information of the selected faculty.
5. User scrolls down the content and clicks on the name of the faculty's website in the “Contacts” area highlighted in the faculty's color “pef.czu.cz”.
6. Application opens the mobile browser on the official page of the university faculty.

**UC5 Find email and write new email to chosen contact.** All email addresses in the application content when clicked open in the default mobile emailing application and creates a new email sending with the fulfilled recipient.

1. On the main screen, the user clicks on the block labeled "Faculties & the institute".
2. Application shows a screen with a list of buttons labeled with faculty name.
3. User clicks on the button labeled “Faculty of Forestry and Wood Sciences”.
4. Application shows screen with the information of the selected faculty.
5. User scrolls down the page.

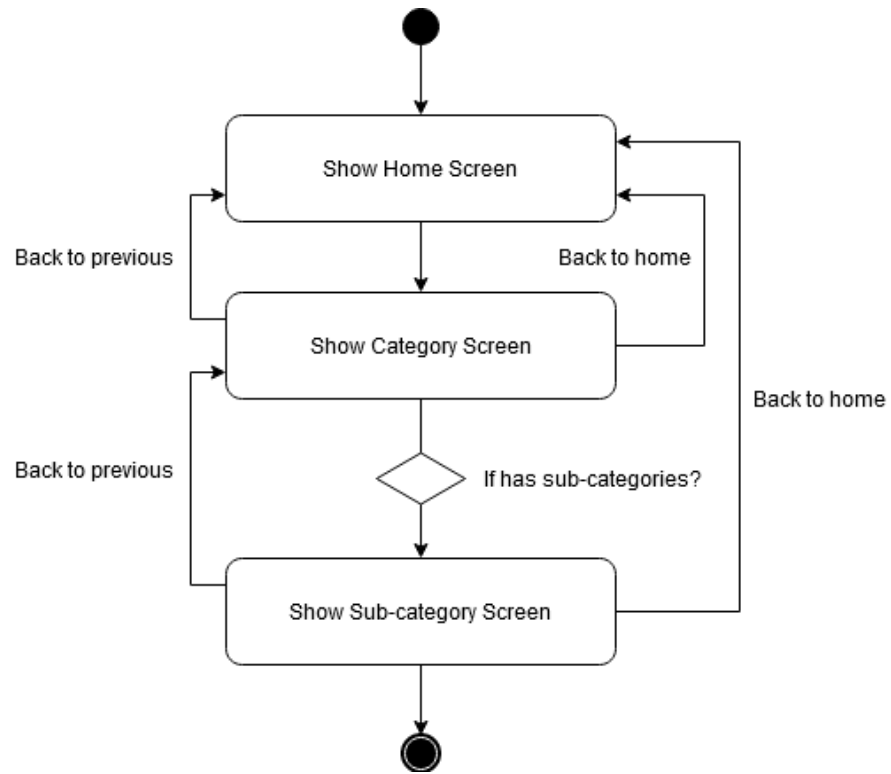
6. Finds a person from the International Relations Office and clicks on the person's email.
7. Application opens the mobile mailing application to send a new email with a fulfilled recipient.
8. User fills in the body of the message and the subject of the message and sends the mail.

**UC6 Find phone number and make a call to chosen contact.** All phone numbers in the application content when clicked open in the default mobile phone application and fill in the to call field with the clicked phone number.

1. User clicks on the block labeled “Study information” on the main screen.
2. Application shows a page with a list of sub-categories buttons.
3. User clicks on the button labeled “Security at CZU”.
4. Application shows a screen with information about the Security department.
5. In the highlighted green area for important information, click on the phone number of the Security department.
6. Application opens the mobile phone application fills phone number.
7. User clicks on the call button.

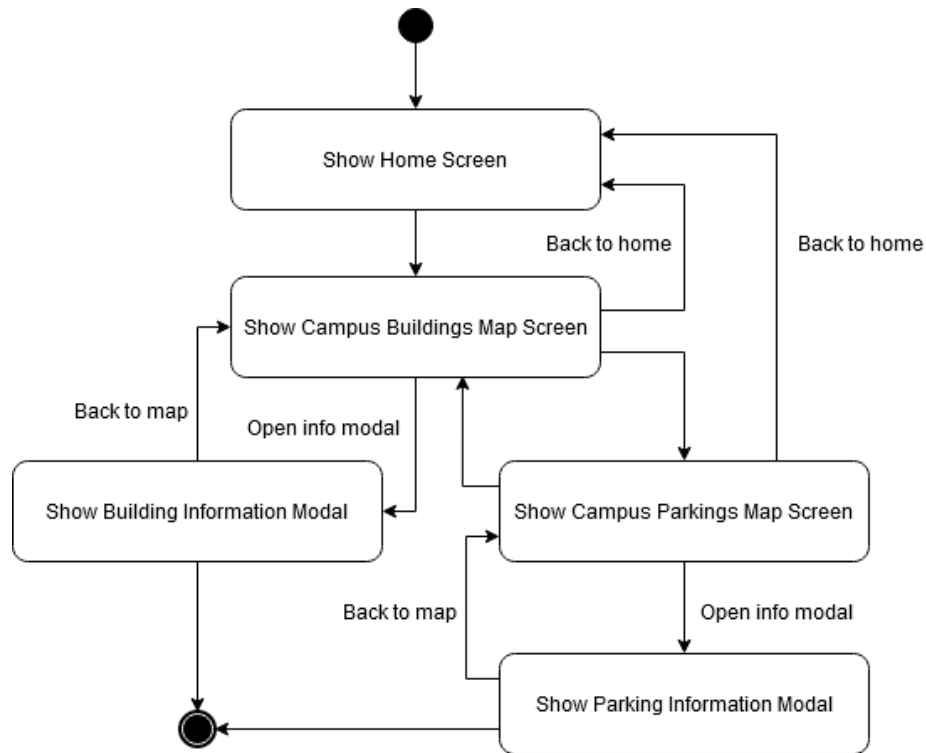
## 4.6 Activity diagrams

Use Case Scenarios is the best method to explain applications from the user's point of view, but in order to show how the system is integrated, the best method is activity diagrams. Below in **Figure 12**, you can see how the user's actions proceed when searching for certain information.



**Figure 4 - Main content activity diagram**

Below in **Figure 13**, you can see what actions are taking place in order to use the map sections in the app.



**Figure 5 - Campus map activity diagram**

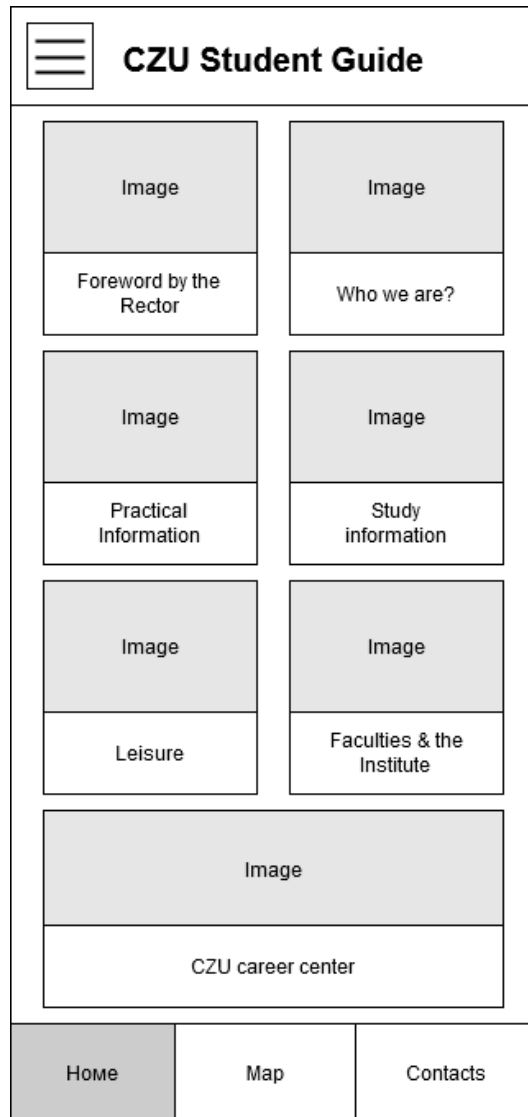
## 4.7 Wired models

The wired model well explains and shows the functionality and features for future application development. It is especially important when designing a wired model to consider the features of the User Interface (UI) and User Experience (UX) for the convenience of using applications by the end user. The following wired models will show the main features of the application and a detailed description of them.

In the first model, **Figure 14** shows the main screen of the application. The top part of the model shows the navigation of the main content. There you can see the changing title of each section, in this case the application on the main screen and title is the name of the application. The icon on the left side of the top navigation is another navigation button. When you click on it, the side navigation will open and there will be the list of secondary information screens as “About Application”.

In the middle, largest part of the screen, there are buttons in the form of blocks. These blocks are divided into categories of main information. Images for these blocks will be selected by subject for each category.

At the bottom of the screen is the bottom navigation, which will be displayed on all main information screens. In this navigation, three buttons are displayed: "Home", "Map" and "Contacts". The "Home" button takes the user from all sections or subsections of the application to the main screen. The "Map" button will take you to the campus maps section. The "Contacts" button will show a screen with the main contact information of the university.

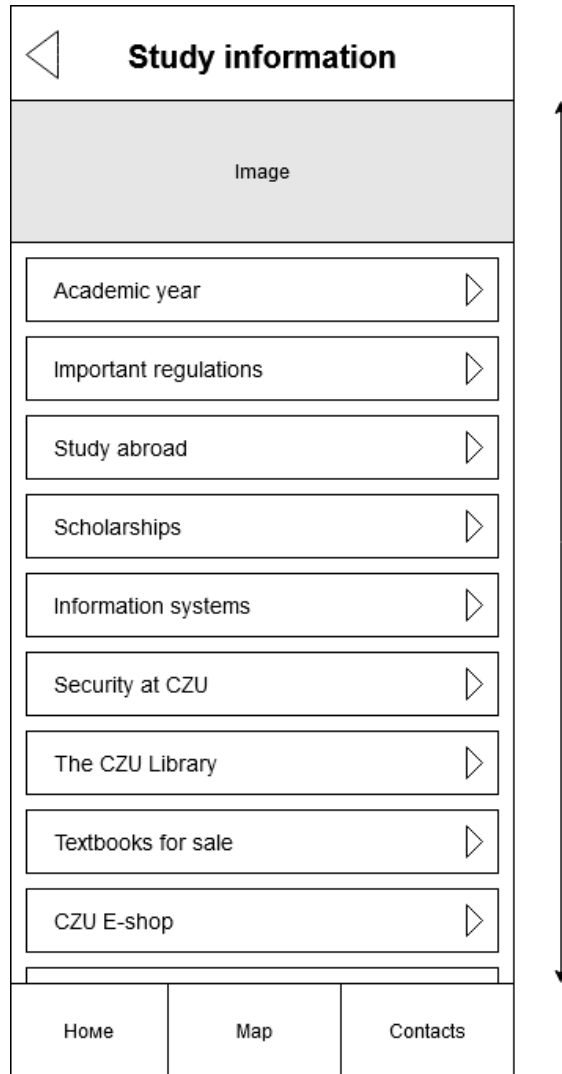


**Figure 6 - Main screen wired model**

**Figure 15** shows the model of the category screen that the user went to from the main screen by clicking on the selected block button. In this case, the model of the "Study information" category screen is shown. If there is more information on the page than fits on the screen of the mobile device, it is possible to scroll through it.

In the top navigation, we can see how the screen name changed to the current category and a button appeared on the left side to return to the previous screen, in this case to the main screen.

In the main part of the screen, the category image and the list of sub-categories appeared in the form of buttons, which will take you to a screen with information about the selected sub-category.



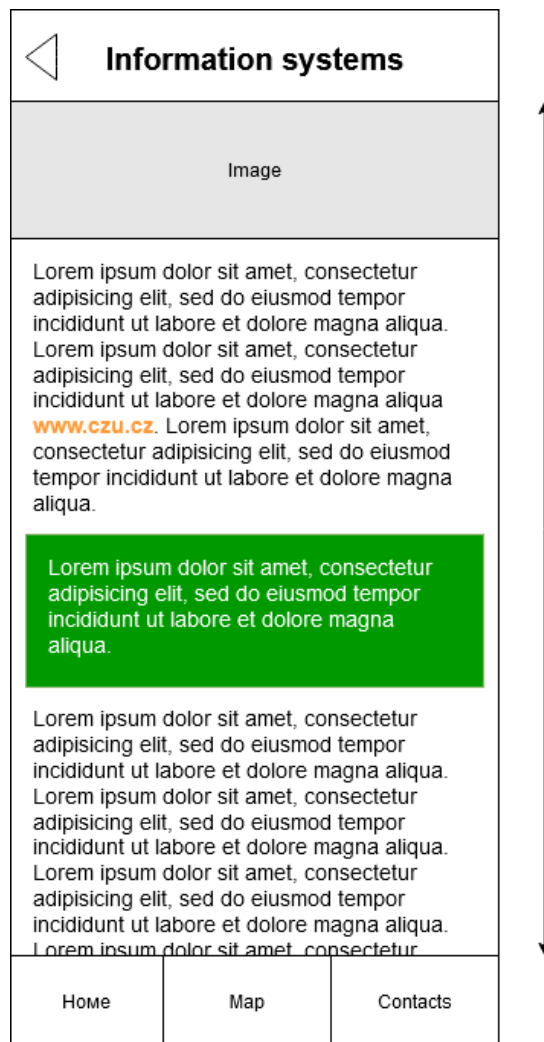
**Figure 7 - Category screen wired model**

**Figure 16** shows how each information screen with the main content will look like.

The top part also changes the name of the current subcategory and the back button to go to the previous parent screen.

In the middle part of the screen, we see the main information. As you can see in the text there is a part highlighted in bold text and a bright color, when you click on it, you can go to the link of a certain page in the browser application of your mobile device. Also, in the main content there are sections highlighted in green, where more important information is indicated. If the screen content is larger than the screen size, then you can scroll through the page.

When you click on the “Home” button in the bottom navigation, the application will take you directly to the main screen.



**Figure 8 - Sub-content screen wired model**

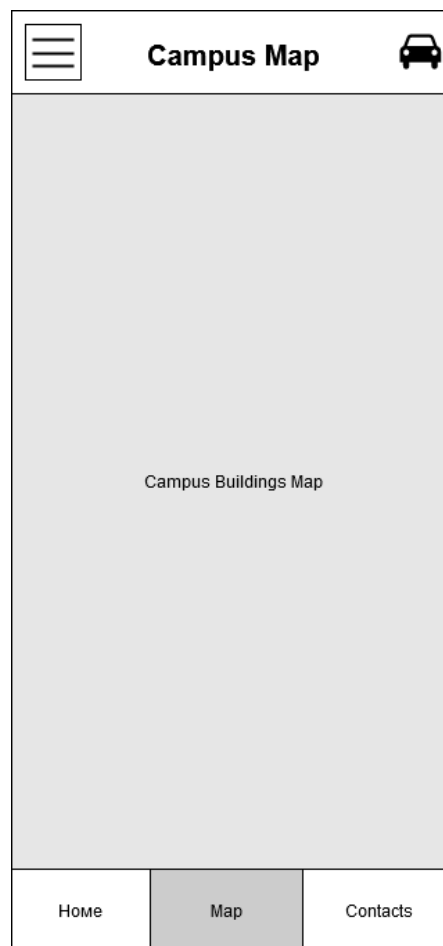
The next **Figure 17** will be a screen with a map of the campus with the designated university buildings. The map can be scaled to better see the location of buildings and passageways to them. You can click on each marker icon and area highlighted in a certain



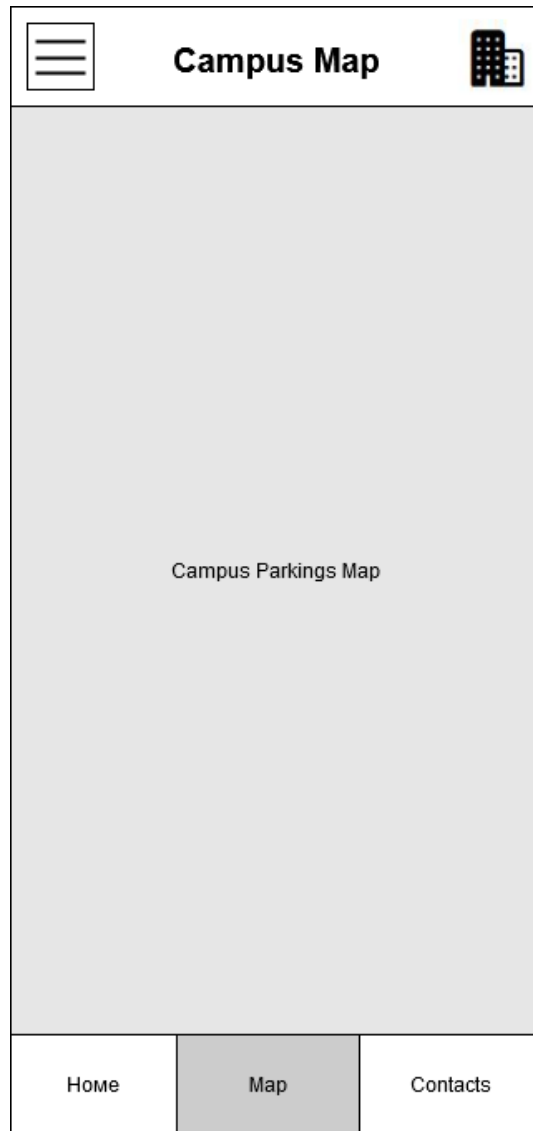
color and open a modal window with information about this object, as you can see in **Figure 19**.

When you go to the map screen, you can see the "car" icon on the right side of the top navigation. By clicking on this icon, you can go to a screen with a map of parking spaces on the campus and entrances to them. When you go to this screen in **Figure 18**, you can see that the icon in the top navigation has changed to another icon "building". These icons are switches between the buildings map and the parking map.

The screen with the map of parking areas and entrances has the same functionality as the map of the buildings. When you click on the highlighted areas and marker icons, it will show the information in a modal window, as shown in **Figure 19**.



**Figure 9 - Campus buildings map wired model**



**Figure 10 - Campus parking map wired model**



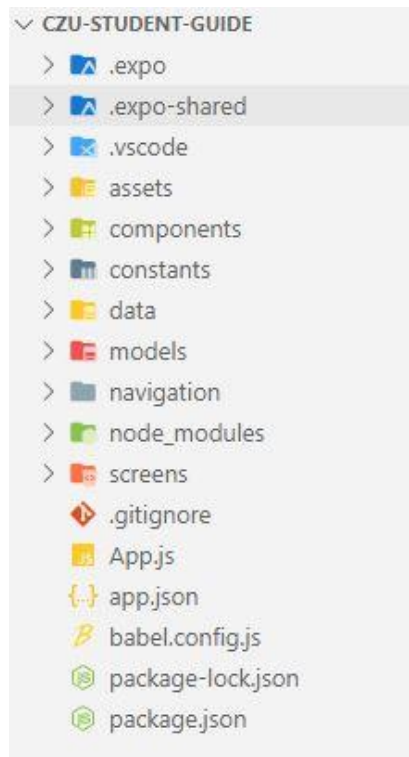
**Figure 11 - Campus map object information modal wired model**

## 4.8 Project description

In this chapter, we will look at the project structure using the Expo tools and dependencies. First, let's go through the external structure of the project, which can be seen in **Figure 21**.

- **/.expo** and **/.expo-shared** folders contain configurations and libraries for the Expo tools to work properly. In this folder, everything is generated from commands on the command line, and we do not change any files there manually.
- **/.vscode** folder contains the Visual Studio IDE settings for the current project.
- **/assets** folder stores all images and icons used locally in the application.
- **/components** folder contains reusable custom UI components which are available from all project files.
- **/constants** folder contains JavaScript files with constant variables which could be used in all projects.
- **/data** this folder contains files with structured data and could be used in all projects
- **/models** folder contains data model files which describe definite data as objects to make more processes on them.
- **/navigation** here is files with whole application navigations logic and structure.
- **/node\_modules** folder contains all dependent external libraries used in the project.
- **/screens** folder contains files which represents each screen of application with separate logic and functions
- **/.gitignore** file specifies intentionally untracked files that Git should ignore.
- **/App.js** file is the starting point of our application.
- **/app.json** file is main application configuration which is important for building application process
- **/babel.config.js** contains configuration information for babel toolchain that is mainly used to convert newer JavaScript code into a backwards compatible version of JavaScript in current and older browsers or environments.
- **/package.json** file at the root of a JavaScript/Node project, that holds metadata relevant to the project and it is used for managing the projects dependencies, scripts, version etc.

- **/package-lock.json** is automatically generated for any operations where npm modifies either the node\_modules tree, or package.json file



**Figure 12 - Project source-code structure**

As a next step, we can take a closer look at some of the internal structure of the application itself and go through the examples of the written code.

#### **4.8.1 Navigations**

In our application user three types of content navigation:

- Stack navigation – top navigation with title and back button for main content.
- Tabs navigation – bottom navigation with icon buttons to switch between screens Home, Campus map and Contacts.
- Drawer navigation – left side drawer navigation for secondary content screen.

To make this native navigation the React Navigation library was used. In our application we are implementing all logic and configuring it in **/navigation/MainNavigation.js** file.

At first, we made separate 4 stack navigation for Secondary content, Main content, Campus map and Contacts screens. To do that we used `createStackNavigator()` function, as an example you can see a short code below for campus map navigator with default navigation options.

```
const CampusMapNavigator = createStackNavigator(  
  {  
    CampusMap: CampusMapScreen,  
  },  
  {  
    defaultNavigationOptions: {  
      headerStyle: {  
        backgroundColor: Colors.primaryColor,  
      },  
      headerTintColor: 'white',  
      headerTitle: 'A Screen',  
    },  
  },  
);
```

To create bottom tabs navigation, we used `createMaterialBottomTabNavigator()` function, you can see in next code:

```
const BottomTabNavigator = createMaterialBottomTabNavigator(tabScreenConfig,  
  {  
    activeTintColor: Colors.accentColor,  
    shifting: true,  
    barStyle: {  
      backgroundColor: Colors.primaryColor,  
    },  
  },  
));
```

As you can see first arguments in function `creatMaterialBottomTabNavigation()` is “tabScreenConfig” was separated into next object variable where “screen” property contains our previous created stack navigation:

```
const tabScreenConfig = {
  Home: {
    screen: CategoriesNavigator,
    navigationOptions: {
      tabBarIcon: (tabInfo) => {
        return (
          <Ionicons name="ios-home-outline"/>
        );
      },
      tabBarColor: Colors.primaryColor,
      tabBarLabel: Home,
    },
  },
  Maps: {
    ...
  },
  Contacts: {
    ...
  },
};
```

To create left side drawer navigation, we used the function `createDrawerNavigator()` with just 2 screen ends Home and About. Into Home we already put bottom navigation which contains main content, campus map and contacts stack navigations.

```
const MainNavigator = createDrawerNavigator(
  {
    Home: {
      screen: BottomTabNavigator,
      navigationOptions: {
        drawerLabel: 'Home',
      },
    },
    About: SideNavigator,
  },
  {
    contentOptions: {
      activeTintColor: Colors.accentColor,
    },
  }
);
```

Now we have compounded all navigation in one MainNavigator object. To import and use it in our application we have put it in a container. To do that we need to use createAppContainer() function, as in next code:

```
export default createAppContainer(MainNavigator);
```

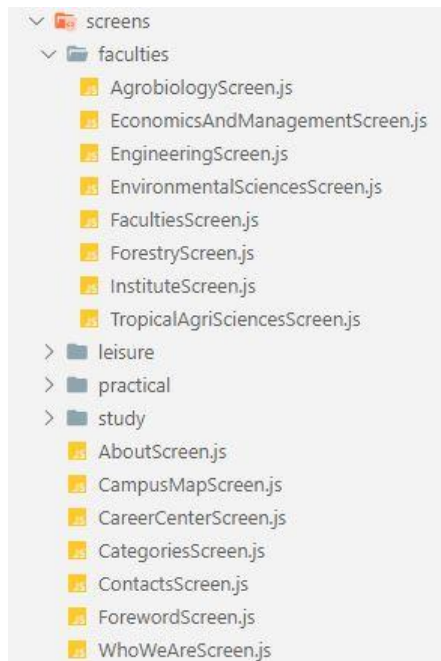
Now MainNavigator could be imported and used in our App.js file as a component in example code below:

```
import React from 'react';
import MainNavigator from './navigation/MainNavigation';

export default function App() {
  return <MainNavigator />;
}
```

#### 4.8.2 Screens

In the screens folder you can see a file for each content screen, because for the first version of the application all content information is static. Screens which have the sub-screen separated in folders as an example you can see faculties folder below on **Figure 22**.



**Figure 13 – Screens Structure**



### 4.8.3 Components

In **/components** we have reusable UI elements which could be imported in any screens. For example, to loop in some data objects and show a few same structured elements on screen. In code below you can see component which used to populate main screen clickable blocks to main contents:

```
import React from 'react';
import {
  TouchableOpacity,
  View,
  Text,
  StyleSheet,
  ImageBackground,
} from 'react-native';

const CategoryGridTile = (props) => {
  return (
    <View style={styles.gridItem}>
      <TouchableOpacity style={{ flex: 1 }} onPress={props.onSelect}>
        <View style={styles.container}>
          <ImageBackground
            source={props.image}
            style={styles.bgImage}
          >
            <View style={styles.titleContainer}>
              <Text style={styles.title} numberOfLines={2}>
                {props.title}
              </Text>
            </View>
          </ImageBackground>
        </View>
      </TouchableOpacity>
    </View>
  );
};

const styles = StyleSheet.create({
  ...
});

export default CategoryGridTile;
```

In `/screens/CatecoriesScreen.js` file component `CategoryGridTile` used with help of React Native Flatlist component, see code below:

```
import React from 'react';
import { FlatList } from 'react-native';
import CategoryGridTile from '../components/CategoryGritTile';
import { CATEGORIES } from '../data/main-info';

const CategoriesScreen = (props) => {
  const renderGridItem = (itemData) => {
    return (
      <CategoryGridTile
        title={itemData.item.title}
        image={itemData.item.image}
        onSelect={() => {
          props.navigation.navigate({
            routeName: itemData.item.route,
            params: {
              categoryId: itemData.item.id,
            },
          });
        }}
      />
    );
  };

  return (
    <FlatList
      data={CATEGORIES}
      renderItem={renderGridItem}
      numColumns={2}
    />
  );
};

CategoriesScreen.navigationOptions = (navData) => {
  ...
};

export default CategoriesScreen;
```

In the above example you see that we import data from `/data/main-info.js`. This file contains model objects which are defined in `/models/category.js`.

#### 4.8.4 Maps integration

To integrate google maps into our application we need to prepare geo-data, for that was used by Google My Maps service. This service gives the possibility to make your own styled maps with custom markers and highlighted with custom color areas. After finishing the custom map of our university campus, all data was exported in a KML file.

KML is a file format used to display geographic data in an Earth browser such as Google Earth. KML uses a tag-based structure with nested elements and attributes and is based on the XML standard.

To facilitate working with geo data, KML file was converted into JSON data files, which gives us preprocess it in format we needed to integrate our map. Below you can see code example from one of the geo data files in `/data/geodata/buildings/library-czu.js`:

```
export const LIBRARY_CZU = {
  'type': 'FeatureCollection',
  'name': 'Library CZU',
  'crs': {
    'type': 'name',
    'properties': { 'name': 'urn:ogc:def:crs:OGC:1.3:CRS84' },
  },
  'features': [
    {
      'type': 'Feature',
      'properties': {
        'Name': 'Library CZU',
        'description': 'CZU in Prague',
        'gx_media_links': '...',
      },
      'geometry': {
        'type': 'Point',
        'coordinates': [14.3730746, 50.1288874],
      },
    },
  ],
};
```

Most important data for us is “geometry” information to show on map in current case at some point on map by “coordinates”. And for showing information about clicked on map points we use “Name” and “description”.

After the data preparation action, we used the Campus Map screen external React Native Maps library to populate all our data on the application map. This library can

process not only Points on the map, but also can create highlighted areas with help of processing Polygon “coordinates” from our geo-data files. With the help of the Polygon component from React Native Maps library on the application map were highlighted university buildings and campus parking areas. With the help of Marker components from the same library were populated all custom university icons on the application map.

## **4.9 Testing**

### **4.9.1 Developers Testing**

During development, according to the Trello task management system, each task was tested using the Expo Go tool and no errors were found. This tool allows you to test applications on any mobile device on which the Expo Go mobile app is installed.

After finishing the development of application with help of Expo tools was generated czstudentguide.apk file. With this file application was installed on 2 different android smartphones.

The first device was Xiaomi Mi A3 smartphone with Android 11 OS and 6.09 inches screen size. Application testing on this device was successful without issues.

The second device was Xiaomi Redmi 4x with older Android 7.1.2 (Nougat) OS and 5.0 inches screen size. On this device testing was also without issues.

#### **4.9.1.1 Use Case Scenario Testing**

As we use user centered design methodology, I asked two of my colleagues to install an application with czstudentguide.apk on their android smartphones. After they get one of the testing scenarios from Use Case Scenario section above and test it according this scenario with commenting on each tested step.

Also, I asked few questions to them after testing:

1. How old are you?
2. Which android device did you use to test the application?
3. How well do you know your android device? (bad, good, very good)
4. Did you understand the scenario and if it is close to reality?
5. Did you feel that the application is user friendly?
6. How quickly did you understand the navigation logic in the application?
7. Did you like the way the application content was organized?

8. Do you have recommendations to improve the application?

#### 4.9.2 Use Case Scenario Testing Results

**First tester results with comments.** For the first tester I give Use Case 2 “Get parking information on map”.

1. User clicks on the middle map button in the bottom navigation of the app.
  - **Tester:** It was easy to find on screen a familiar icon from google maps application.
2. Application shows a page with a map of the university buildings.
  - **Tester:** Map could be more zoomed on campus.
3. User clicks on the car icon in the right part of the top navigation.
  - **Tester:** Car icon looks good, but for me it will be more obvious if there will be a normal parking sign with P.
4. Application shows a screen with a map of the location of car entrances and parking areas.
  - **Tester:** Zooming on the map is good because all icons and highlighted areas are visible.
5. User swipes on the map in a certain area and clicks on the marker icon or the area highlighted in different colors on the map.
  - **Tester:** Map functionality works perfectly as in original google maps application.
6. Application shows information in a modal window about the clicked area or marker icon on the map.
  - **Tester:** Works good and useful information.
7. User clicks on the “Ok” button in the modal window.
  - **Tester:** Better to use “close” or “back” button text.
8. Application closes the modal window and shows campus parking map.
  - **Tester:** Works good.

**First tester questions answers.**

1. How old are you? – 21
2. Which android device did you use to test the application? - Samsung Galaxy A41

3. How well do you know your android device? (bad, good, very good) – good
4. Did you understand the scenario and if it is close to reality? -Yes understood, yes it could be useful in reality
5. Did you feel that the application is user friendly? – Yes
6. How quickly did you understand the navigation logic in the application? - Very quick because navigation looks like in many other applications.
7. Did you like the way the application content was organized? – Yes
8. Do you have a recommendation to improve the application? - Application works good and simple

**Second tester results with comments.** For the second tester I give Use Case 2 “Find email and write new email to chosen contact”.

1. On the main screen, the user clicks on the block labeled "Faculties & the institute".
  - **Tester:** Easy to find.
2. Application shows a screen with a list of buttons labeled with faculty name.
  - **Tester:** Buttons looks great.
3. User clicks on the button labeled “Faculty of Forestry and Wood Sciences”.
  - **Tester:** Easy to find.
4. Application shows screen with the information of the selected faculty.
  - **Tester:** Will be great to get more info about faculty on this page
5. User scrolls down the page.
  - **Tester:** Done.
6. Finds a person from the International Relations Office and clicks on the person's email.
  - **Tester:** I thought that I will do copy paste stuff with email, but this feature is great.
7. Application opens the mobile mailing application to send a new email with a fulfilled recipient.
  - **Tester:** Works great.
8. User fills in the body of the message and the subject of the message and sends the mail.
  - **Tester:** Sure, it will work, but I don't want to send this email.

### **Second tester questions answers.**

1. How old are you? – 25
2. Which android device did you use to test the application? - OnePlus 6
3. How well do you know your android device? (bad, good, very good) - Very good
4. Did you understand the scenario and if it is close to reality? - Yes, yes
5. Did you feel that the application is user friendly? – Yes
6. How quickly did you understand the navigation logic in the application? - Quick, because on the top menu which opens the left side menu with only home and about.
7. Did you like the way the application content was organized? - Yes, but faculties button on home page could be on top to see it first
8. Do you have a recommendation to improve the application? - Good application, no recommendations

### **4.9.3 Use Case Scenario Testing Summary**

Apart from the small comments from the testers, the test results indicate that the testing of the first version of the application was successful. In both tests, even though the smartphones were of different brands, the testers did not have any errors or failures in the application.

From the comments of the testers, we need to analyze and, if necessary, make changes to the following points:

1. On opening buildings, the screen map could be more zoomed on campus.
2. Use the parking sign icon instead of car icon in the map switching button.
3. Change map information modal “Ok” button text to “Close” or “Back”.
4. Add more information about faculties.
5. Analyze the need for a left side drawer navigation.
6. Analyze the need to move the “Faculties & the Institute” block button to the upper position on the main screen.

## 4.10 Deployment

Finally, after all the stages of development of our application, we are ready to show it to the world. In order to publish our app on the Google Play Market platform, I have prepared a step-by-step description of all the actions.

Before you start this process, you need to create an account on the Google Play Console service.

The first thing we need to publish an application is to collect the required information.

### **Required information:**

1. Application name.
2. Short description.
3. Long description with application features and functionality.
4. URL with support information about your app (can be a website or email).
5. Rating of application. By answering a few questions about your project and, based on your answers, the rating will be set.
6. Bundle identifier. For our application by standards is “com.czustudentguide.app”.
7. Profile (brand, organization, customer) name for a store.
8. Information about the publisher.
9. Application category. Our application under education category.
10. Application icon with sizes 512x512 pixels.
11. Link to application privacy policy.

After preparing the required information, we proceed to build the application itself. At the end of the build process, we will get a .apk file that can be downloaded to the Google Play Market platform using the Google Play Console service. The Expo tools that we use to develop our app also greatly simplify the process of building the .apk file. In order to use this feature, we need to create an account on the Expo service web pages.

### **Steps to build application:**

1. Configure app.json for Expo Tools. All information about configuration parameters are in Expo Documentations. In our configuration we will use part of previously prepared information.



2. Publish our application at the Expo. For that we just enter Expo-CLI tool command “expo publish” in the command line console. This process requires credentials to the Expo account.
3. Enter “expo build:android -t apk” in the command line console to get our .apk file. Building an application can take several minutes.

After finishing the building process Expo-CLI will provide a link to Expo server, where we can download our final .apk file.

Once we have prepared the .apk file, we are ready to upload it to the Google Play Market platform. To do this, you need to log in to the web console of the Google Play Console service and follow these steps:

1. Press "Create new app" in “All apps”.
2. Fill all the required information.
3. Upload application, our czustudentguide.apk file.
4. Submit for the review.
5. Wait for approval. This process can take from one to several days.

After approval we can easily download our CZU Student Guide application in Play Market application on android mobile device.

## 5 Result and Discussion

### 5.1 Installation

CZU Student Guide application could be downloaded and installed on Google Play Market by URL address [play.google.com/store/apps/details?id=com.czustudentguide.app](https://play.google.com/store/apps/details?id=com.czustudentguide.app) or by scanning following QR code with QR code reader on mobile devices.



Figure 14 - CZU Student Guide application QR code

The minimum requirements are Android 4.1 and above. The application does not require any special permissions on Android devices. All content in the application accessible offline.

### 5.2 Accessing contents

To view the contents of the app, you can use three types of navigation: Top navigation bar, Left side drive navigation and bottom navigation. You can also use clickable blocks on the main page and clickable buttons on category pages.

### 5.3 Using campus map

You can open the campus map by clicking on the middle icon in the bottom navigation. You can switch the map to display university buildings or entrances and parking spaces by clicking on the car or building icon in the upper navigation on the right side. The map can be scaled with a two-finger swipe to the center or out and you can move on the map by sliding to left, to right, to top and bottom on the mobile device screen. Sectors and icons marked with different colors are clickable. When you click on them, a modal window opens with information about the object, and when you click on the " Ok " button, you can close it and return to the map.

## **5.4 Discussion**

In theoretical part we reviewed two mobile applications of University of Otago and European University of Cyprus, which has same purposes as our CZU Student Guide mobile application. The main difference and advantage of our app over them is the availability of all content and functionality in offline mode. For example, most of the information about the university and the study information in both applications is available only in the form of links to the web pages of universities.

University of Otago application has own advantage, it is designed more for already enrolled students, because it has features as Personalized study timetable, Quick access to personal grades, Daily event listings during Academic Orientation Week and Easy access to key dates and upcoming public lectures. All these features authorization functionality, what makes application personal for each student. These features also could be implemented in future into CZU Student Guide application. And it can be useful not only to get acquainted with the university, but also to accompany it during the entire academic period.

Other main features of all three applications are university campus map. In our application is most simple version of this functionality, because it does not show device location on map and there is now search bar with list of campus objects. But during the implementation of the map in our application, the development was carried out considering the fact that these two map functionality extensions could be easily implemented in future.

## **5.5 Recommendations**

From experience of writing this thesis and working on the CZU Student Guide mobile application project, there is few recommendations:

1. During application designing use activity diagrams, data flow diagrams, make good use case analysis. All this method will show you what is more important and needs more attention in your project.
2. Always think about who the application is intended for and how to provide a certain functionality of the application in the most user-friendly way.
3. When choosing a technology for developing an application, always consider the possible expansion of functionality in the future.

4. After selecting a technology for development, study its documentation in detail, this will greatly facilitate and speed up the development process.

## 6 Conclusion

The first main objective of the thesis was to analyze the mobile application market which we did in the theoretical part of this work. Author looked at the trends in the development of the two most important mobile app markets, got acquainted with their statistics in recent years and reviewed the requirements for publishing the application.

The second main objective was to develop, design and deploy a mobile application for students of the Czech University of Life Sciences Prague, what we did in the practical part of the thesis. Where we started with a survey of potential future application users, which helped us determine what technologies we would use for development. Then we made design analysis by defining application requirements and making Use Case Scenarios, Activity Diagrams and Wired Models. After we looked at tools used to help our development process.

The description of the project showed us in detail the structure of the source code and code examples of some application functionality. During testing, we made sure that the application works without problems on different mobile devices. We also received the first feedback and comments from our testers, which helped us determine how to make the app even more convenient to use for our end users. And we finished the practical part with a detailed explanation of the main aspects of the application deployment and its publication on the Google Play Market platform.

## 7 References

1. Apple Developer Documentation. In: Apple Developer [online]. One Apple Park Way, Cupertino, CA 95014, USA: Apple, 2009 [cit. 2018-09-02]. Dostupné z: <https://developer.apple.com/documentation/>
2. Google Play Developer API [online]. 1600 Amphitheatre Parkway, Mountain View, California 94043, United States: Google, 2018 [cit. 2018-09-09]. Dostupné z: [https://developers.google.com/android-publisher/getting\\_started](https://developers.google.com/android-publisher/getting_started)
3. HERMES, Dan. Learn Android Studio 3: efficient Android app development. 3rd edition. New York, NY: Springer Science Business Media, [2018]. Wrox programmer to programmer. ISBN 978-148-4231-555.
4. HERMES, Dan. Xamarin Mobile Application Development: Cross-Platform C# and Xamarin.Forms Fundamentals. 3rd edition. California: Apress, [2015]. Wrox programmer to programmer. ISBN 978-148-4202-142.
5. MCWHERTER, Jeff a Scott GOWELL. Professional Mobile Application Development. 3rd edition. India: Wiley India Pvt., [2012]. ISBN 9788126538249.
6. MUKHERJEA, Sougata. Mobile application development, usability, and security. Hershey PA: Information Science Reference, [2017]. ISBN 978-152-2509-455.
7. STILL, Brian a Kate CRANE. Fundamentals of User-Centered Design: A Practical Approach. New York City: CRC Press, [2017]. ISBN 978-1498764360.
8. Expo Docs [online]. Palo Alto, California 94301, US: Expo, 2020 [cit. 2021-02-25]. Dostupné z: <https://docs.expo.io/>
9. The Q4 2020 Data Digest Report: Get the Latest Mobile Trends in 2020 [online]. Sensor Tower Reports Website: Sensor Tower, 2021 [cit. 2021-03-10]. Dostupné z: <https://go.sensortower.com/q4-2020-data-digest-report.html>
10. Number of available applications in the Google Play Store from December 2009 to December 2020 [online]. The Statistics Portal Statista: Statista Research Department, 2021 [cit. 2021-02-27]. Dostupné z: <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store>
11. Využívání informačních a komunikačních technologií v domácnostech a mezi jednotlivci - 2019 [online]. Portal Český statistický úřad: Český statistický úřad, 2019 [cit. 2021-03-31]. Dostupné z: <https://www.czso.cz/csu/czso/vyuzivani-informacnich-a-komunikacnich-technologii-v-domacnostech-a-mezi-jednotlivci-2019>

## 8 Appendix

### 8.1 Questions

The survey questions and answers:

1. Your gender:
  - a. Female
  - b. Male
  - c. Other
2. Your age:
  - a. 10 – 18
  - b. 19 – 25
  - c. 26 – 30
  - d. 30 and more
3. What is your current student status?
  - a. Student in secondary school
  - b. Finished secondary school
  - c. University student
  - d. Other
4. What device you mostly use for searching an information?
  - a. PC or laptop
  - b. Tablet
  - c. Smartphone
  - d. Other
5. Which operating system is on your personal smartphone?
  - a. Android
  - b. iOS (Apple)
  - c. Windows mobile
  - d. Other
6. Do you always have access to the Internet on your smartphone?
  - a. Yes
  - b. No
  - c. Maybe

7. Which mobile app are you most using on your smartphone?
  - a. Online
  - b. Offline
  - c. Both
8. Was there access to the open Internet in the universities you visited?
  - a. Yes
  - b. No
  - c. Maybe
9. How do you prefer to receive information about universities of interest to you?
  - a. University website
  - b. University mobile application
  - c. Hard printed brochures
  - d. Other
10. How would you navigate in a large university campus?
  - a. Physical indicators on campus
  - b. A maps application on smartphone
  - c. Hard printed map of campus
  - d. On the advice of other people
  - e. Other



## 8.2 Screenshots

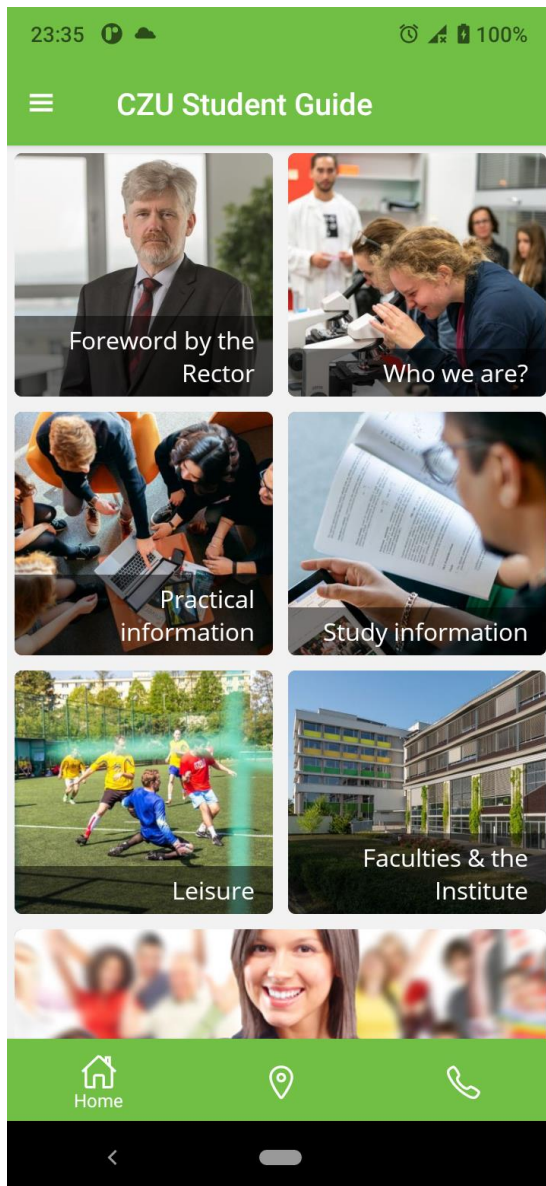


Figure 15 – Home Screenshot

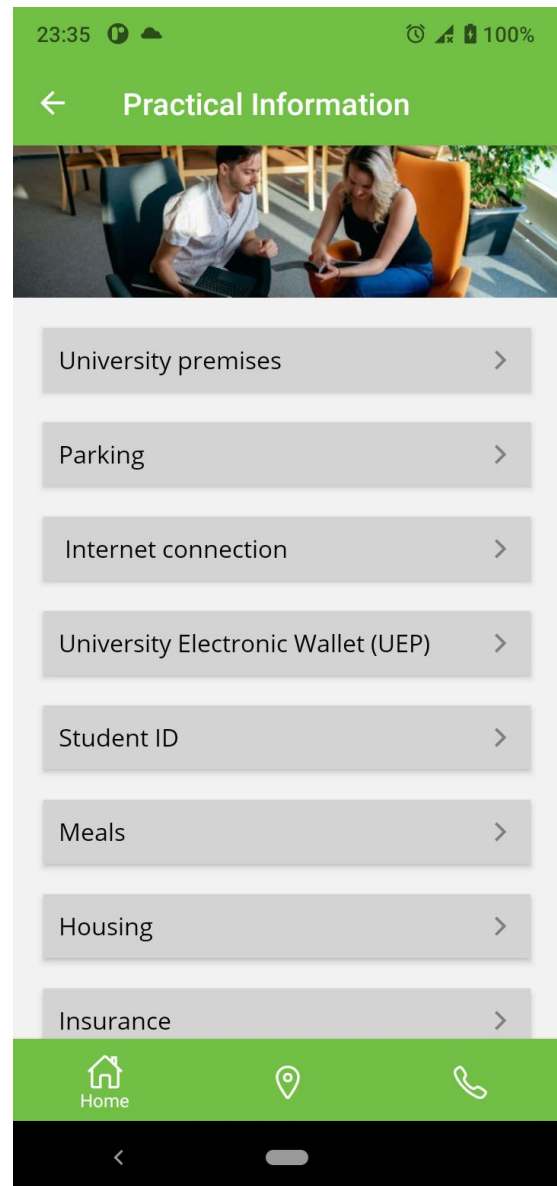
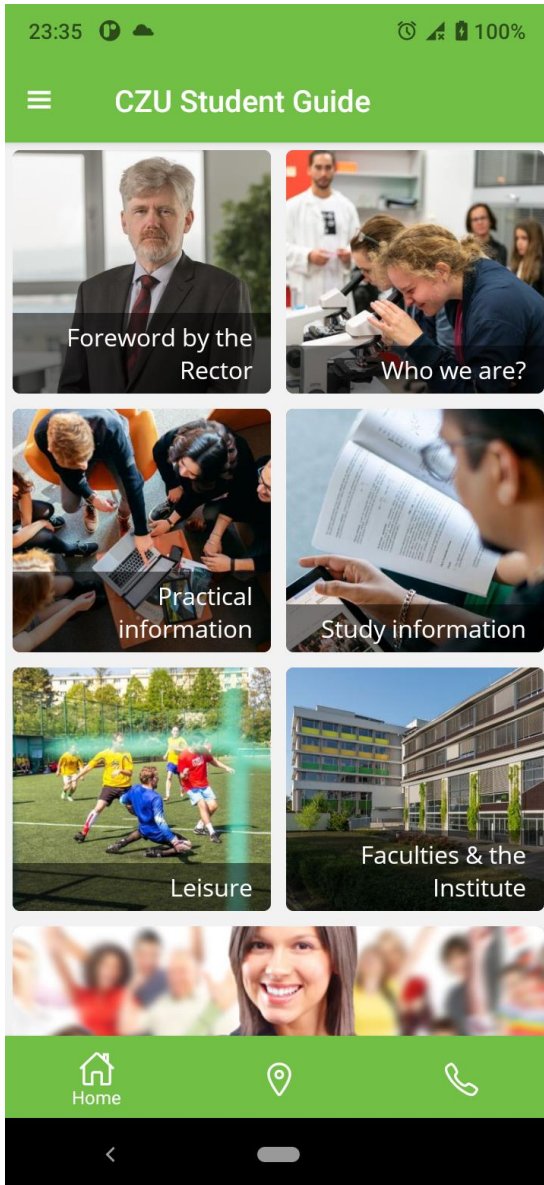
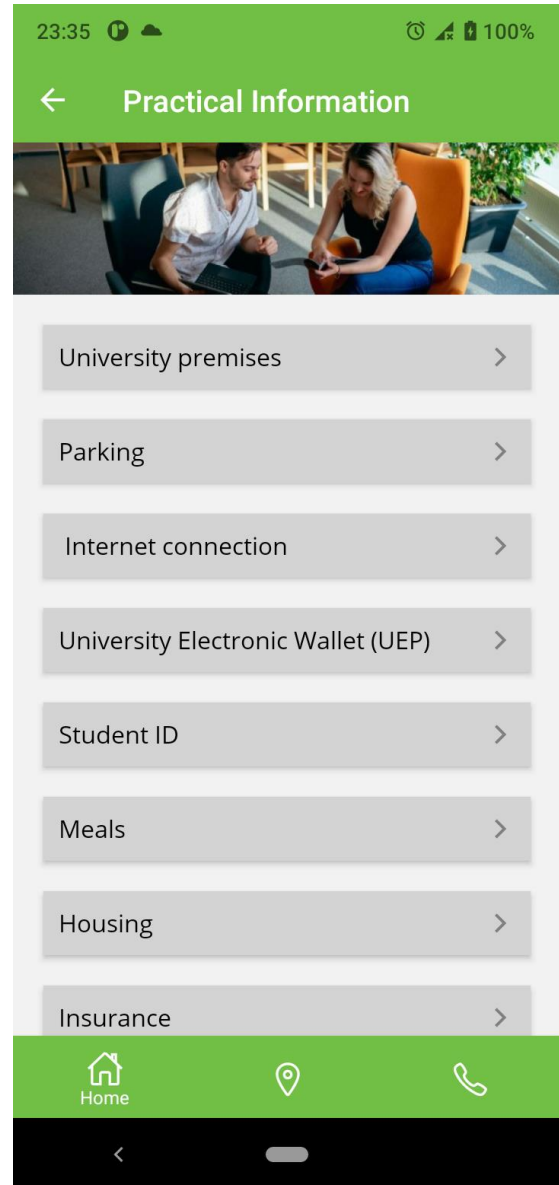


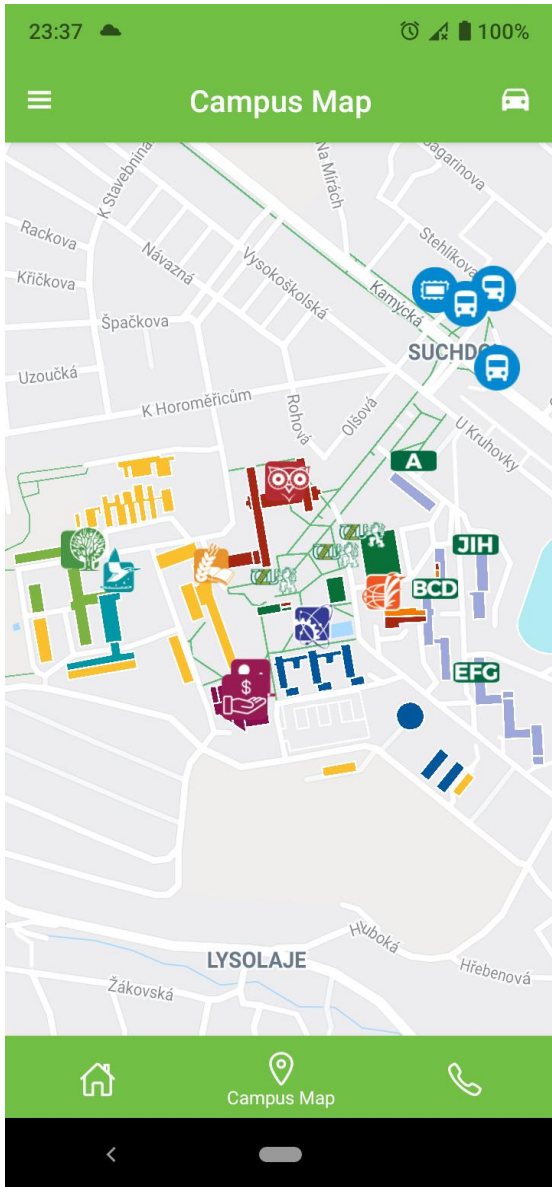
Figure 16 – Practical information Screenshot



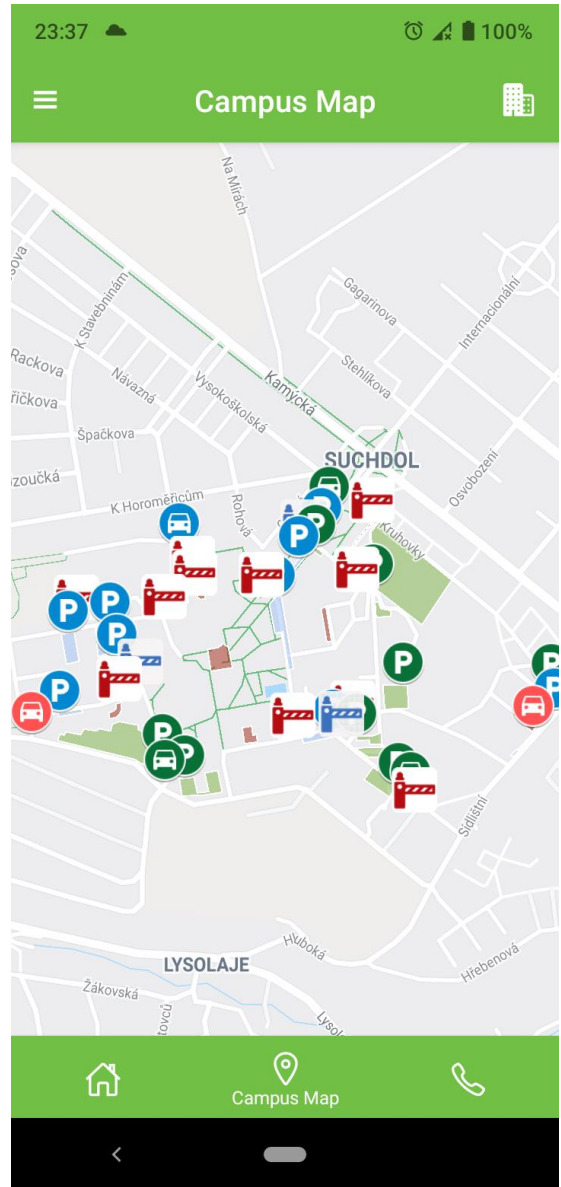
**Figure 17 – Home Screenshot**



**Figure 18 – Practical information Screenshot**



**Figure 19 - Buildings map Screenshot**



**Figure 20 - Parking map Screenshot**