TARTU UNIVERSITY

FACULTY OF SOCIAL SCIENCES

NARVA COLLEGE

STUDY PROGRAMM „INFORMATION TECHNOLOGY SYSTEMS DEVELOPMENT“

Aleksei Jekimov

CREATING AN ANDROID APPLICATION TO SERVE OFFER AND DEMAND FOR PRIVATE TUTORS AND COACHES

Diploma Thesis

Supervisor: Assistant Andre Sääsk

NARVA 2018

Olen koostanud töö iseseisvalt. Kõik töö koostamisel kasutatud teiste autorite tööd, põhimõttelised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

.....

Töö autori allkiri ja kuupäev

Non-exclusive licence to reproduce thesis

I, Aleksei Jekimov (date of birth: 16.03.1994),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright.

"Creating an Android application to serve offer and demand for private tutors and coaches" supervised by assistant Andre Sääsk

2. I am aware of the fact that the author retains the right referred to in point 1.

3. This is to certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Act.

Narva, **23.05.2018**

Creating an Android application to serve offer and demand for private tutors and coaches

Abstract: In this thesis, tutors and trainers are considered how people are looking for an opportunity to earn on their knowledge and skills. Some people need help in the fitting of a skill, and some people can help them with this. With this thesis, the Android application called Tutorbe was designed to meet the needs of both groups. Everyone interested can go to the application, look at existing ads or add their own and later contact the person they need with the help of an email message or call the phone number indicated in the announcement.

Keywords: Android, application, smartphone, tutor, coach, Firebase

Table of Contents

Introduction

The problem

The attitude towards such an occupation as tutoring has changed greatly in recent years. Not very long ago, studying with a tutor was a shameful thing. Unfair, lazy or hard-to-learn children had to deal with extra tasks and hire a tutor. Today things changed greatly to the extent that private tutoring became prestigious. Private tutors are an indicator of wealth and well-being of the family. A child, who takes private lessons, usually does well at school. Similarly, adults also often resort to the help of tutors and coaches. Overall, we can say that nowadays people hire private tutors to get high scores on exams, to get admission to a university or to get an interesting and highly paid work.

In the result of the aforementioned situation, there is a significant market for private tutoring in Estonia, but finding information is often not an easy task due to the lack of specialized services. People have to look for tutors and trainers in newspapers or ask friends and acquaintances, which greatly hinders the search process. These search methods are very outdated and do not work well, people need to take many unnecessary actions and, for this reason, sometimes abandon their studying plans.

The solution

Instead of using old and questionable ways of searching, such as buying newspapers and searching for acquaintances, who previously faced the same problem, people can now use their mobile devices instead. If there would be a dedicated application for finding private tutors, all a person would need to do would be downloading an application and start searching for a tutor among already existing posts or add their own post about finding a tutor or offering a tutoring service. In addition, such an application would help people earn money by sharing their knowledge and expertise.

The author hopes that with the help of this application finding a tutor in Estonia will become more convenient and effective, and qualified people will have the opportunity of additional earnings.

The aim

The aim of this work is to create a mobile application Tutorbe to help people look for a tutor or a coach, and to help tutors and coaches reach their audience.

The motivation

The author of this thesis seeks to study the technology of mobile application development, as well as to improve his general programming skills. In addition, the author is interested in developing such an application, because at this time it is unique for Estonia and can help many people, including the author himself.

Outline

The thesis consists from an introduction, two chapters and a conclusion. The first chapter covers the state of art of the technology used. In addition, there is an overview of already existing solutions in this field. The second chapter provides an in-depth overview of the development of Tutorbe and its features, covering both functional and non-functional requirements for the application. It also describes the design and architecture of the application, and explains the methodology used in the development. The third chapter describes tests conducted by users.

1 Implemented technologies and available solutions

1.1 Technologies

1.1.1 Smartphone

Smartphone is a mobile phone that includes advanced functionality beyond making phone calls and sending text messages. Most smartphones have the capability to display photos, play videos, check and send e-mail, and surf the Web. Modern smartphones, such as the iPhone and Android based phones can run third-party applications, which provides limitless functionality.

While mostly business users initially used smartphones, they have become a common choice for consumers as well. Thanks to advancements in technology, modern smartphones are smaller and cheaper than earlier devices. Users also have a much wider range of smartphones to choose from than before. While the RIM Blackberry dominated the smartphone market for many years, other manufacturers like Apple, HTC, and Samsung now offer a wide variety of smartphone options as well. This increase in smartphone availability has led to a corresponding decline in the usage of standard PDAs, which do not include phone capabilities.

Since smartphones have a wide range of functionality, they require advanced software, similar to a computer operating system. The smartphone software handles phone calls, runs applications, and provides configuration options for the user. Most smartphones include a USB connection, which allows users to sync data with their computers and update their smartphone software. (Smartphone definition 2010)

The author chose the smartphone as a development platform because of its convenience availability. The main competitors of smartphones are computers, but people take smartphones with them a lot more often, smartphones easily fit in pockets and are used as daily gadgets.

1.1.2 Android

Android operating system is one of the most widely used mobile Operating System these days. Android mobile operating system is based on the Linux kernel and is developed by Google. Android operating system is primarily designed for smartphones and tablets. Since Android is an open source it has become the fastest growing mobile operating system. Due

to its open nature, it has become favourite for many consumers and developers. Moreover, software developers can easily modify and add enhanced feature in it to meet the latest requirements of the mobile technology. Android users download more than 1.5 billion applications and games from Google Play each month. Due to Its Powerful development framework users as well, software developers are able to create their own applications for wide range of devices. Some of the key features of Android operating system are:

Application Frame work, Dalvik virtual machine, Integrated browser, Optimized Graphics, SQLite, Media Support, GSM Technology, Bluetooth, Edge, 3G, Wi-Fi, Camera and GPS. To help the developers for better software development Android provides Android Software development kit (SDK). It provides Java programming Language for application development. The Android software development kit includes a debugger, libraries, a handset emulator based on QEMU (Quick Emulator), documentation, sample code, and tutorials. (Singh 2014)

The main competitors of Android are iOS by Apple and Windows Phone. The author used the Android as the operating system because currently Android is by far the most popular mobile platform (Smartphone OS 2017). In addition, the author owns an Android phone, which simplifies the creation and testing of the application.

1.1.3 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as (Meet Android Studio 2018):

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Instant Run to push changes to your running app without building a new APK
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support

- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

Android studio has been used as an integrated development environment, because it is official default IDE for developing Android applications and it's very convenient. In addition, it based on IntelliJ IDEA, which the author used for coding before.

1.1.4 Java programming language

Java programming language is a general-purpose, concurrent, class-based, object-oriented language. It is designed to be simple enough that many programmers can achieve fluency in the language. The Java programming language is related to C and C++ but is organized rather differently, with a number of aspects of C and C++ omitted and a few ideas from other languages included. It is intended to be a production language, not a research language, and so, as C. A. R. Hoare suggested in his classic paper on language design, the design has avoided including new and untested features.

The Java programming language is strongly and statically typed. This specification clearly distinguishes between the compile-time errors that can and must be detected at compile time, and those that occur at run time. Compile time normally consists of translating programs into a machine-independent byte code representation.

Run-time activities include loading and linking of the classes needed to execute a program, optional machine code generation and dynamic optimization of the program, and actual program execution.

The Java programming language is a relatively high-level language, in that details of the machine representation are not available through the language. It includes automatic storage management, typically using a garbage collector, to avoid the safety problems of explicit deallocation (as in C's free or C++'s delete).

High-performance garbage-collected implementations can have bounded pauses to support systems programming and real-time applications. The language does not include any unsafe constructs, such as array accesses without index checking, since such unsafe constructs would cause a program to behave in an unspecified way.

The Java programming language is normally compiled to the bytecode instruction set and binary format defined in The Java Virtual Machine Specification, Java SE 10 Edition. (Gosling; Joy; Steele; Bracha; Buckley; Smith 2018)

The Java language was used because it is the main language used to develop Android applications, and the author has previous experience with this language.

1.1.5 XML

Extensible Markup Language, abbreviated XML, describes a class of data objects called XML documents and partially describes the behaviour of computer programs, which process them. XML is an application profile or restricted form of SGML, the Standard Generalized Markup Language [ISO8879]. By construction, XML documents are conforming SGML documents.

XML documents are made up of storage units called entities, which contain either parsed or unparsed data. Parsed data is made up of characters, some of which form character data, and some of which form markup. Markup encodes a description of the document's storage layout and logical structure. XML provides a mechanism to impose constraints on the storage layout and logical structure.

A software module called an XML processor is used to read XML documents and provide access to their content and structure. It is assumed that an XML processor is doing its work on behalf of another module, called the application. This specification describes the required behaviour of an XML processor in terms of how it must read XML data and the information it must provide to the application. (Extensible Markup Language (XML) 1.0 1998)

The author used XML in the development process because it is the default way for creating layouts used in Android Studio.

1.1.6 Gradle

Gradle is an open-source build automation system that builds upon the concepts of Apache Ant and Apache Maven and introduces a Groovy-based domain-specific language (DSL) instead of the XML form used by Apache Maven for declaring the project configuration. Gradle uses a directed acyclic graph ("DAG") to determine the order in which tasks can be run.

Gradle was designed for multi-project builds, which can grow to be quite large. It supports incremental builds by intelligently determining which parts of the build tree are up to date; any task dependent only on those parts does not need to be re-executed.

The initial plugins are primarily focused on Java, Groovy and Scala development and deployment, but more languages and project workflows are on the roadmap. (Gradle 2018)

The author used Gradle because it is the default build tool in the Android Studio, and it performs all the necessary tasks.

1.1.7 Backend as a service

BaaS providers allow subscribers to choose their own database and use preimplemented push-services or other applicational support and on the other hand some providers only offer one database and no preimplemented applicational support. Since frontend applications are fully independent of BaaS offerings and BaaS mostly provides universally important backend options, the BaaS offerings allow developers to only concentrate on the applications data. Under the control aspect the user controls the application and the BaaS provider controls everything else. The provider has to provide availability and other requirements important for cloud services. (Kohl n.d)

The author chose backend as a service because it is easier to work with than a real server, and also to learn a new tool that facilitates the work of the programmer.

1.1.8 Firebase

Firebase is a fully managed platform for building iOS, Android, and web apps that provides automatic data synchronization, authentication services, messaging, file storage, analytics, and more. (Build an Android App Using Firebase and the App Engine Flexible Environment 2018)

The author decided to use Firebase because Google provides the service free and it fits very well with the Android Studio. It is very easily implemented and easy to work with.

1.2 Available solutions

1.2.1 Coach.me – Instant Coaching

This application is an assistant for people who want to change their lives for the better. For example, throw a bad habit or start playing sports. In the application there are three levels of coaching, in the first one users set themselves tasks, perform them and receive rewards for them, in the second there is an opportunity to ask questions from other users and the third, paid level, the user is assisted by coaches.

This application, although somewhat similar, provides a different service compared to the one described in the problem of this thesis, so it is not a direct competitor for the author's project.

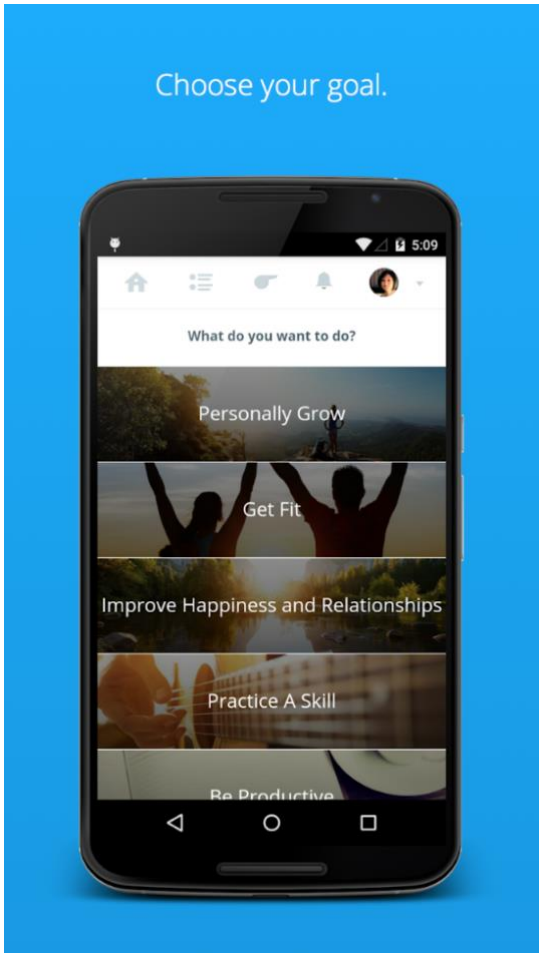

Figure 1. *Coach.me's main screen* [1]

1.2.2 Tutor.com To Go

This application has a similar goal with Tutorbe, but according to the author and people, who leave reviews in Google Play, its design is not very friendly and it has not been updated for a long time. In addition, one cannot enter this application without login or registration, which can only be completed through their site. E.g. in Tutorbe users have the option of anonymous login, which is performed in one click.

[1] https://play.google.com/store/apps/details?id=com.liftworldwide.lift

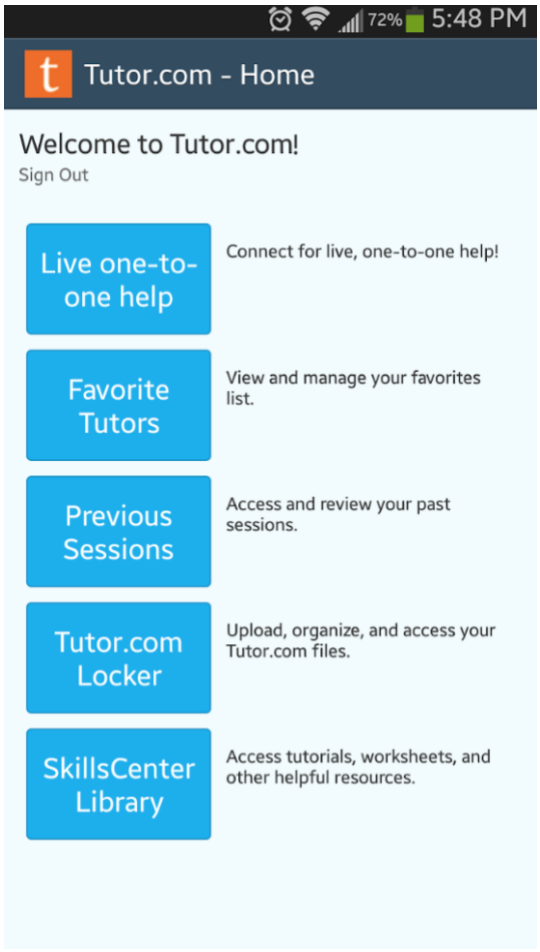

Figure 2. *Tutor.com To Go's main screen*[2]

1.2.3 Wyzant – Find Expert Tutors

This application is similar to what the author of this thesis wants to develop. With the help of this application, people can look for tutors to prepare for tests, to help with learning math, a new language, etc. In addition, there it has different levels of tutors and the feedback option. Wyzant has excellent reviews and ratings in Google Play, but its accessibility in Estonia is limited.

[2] https://play.google.com/store/apps/details?id=com.tutor.to.go.android

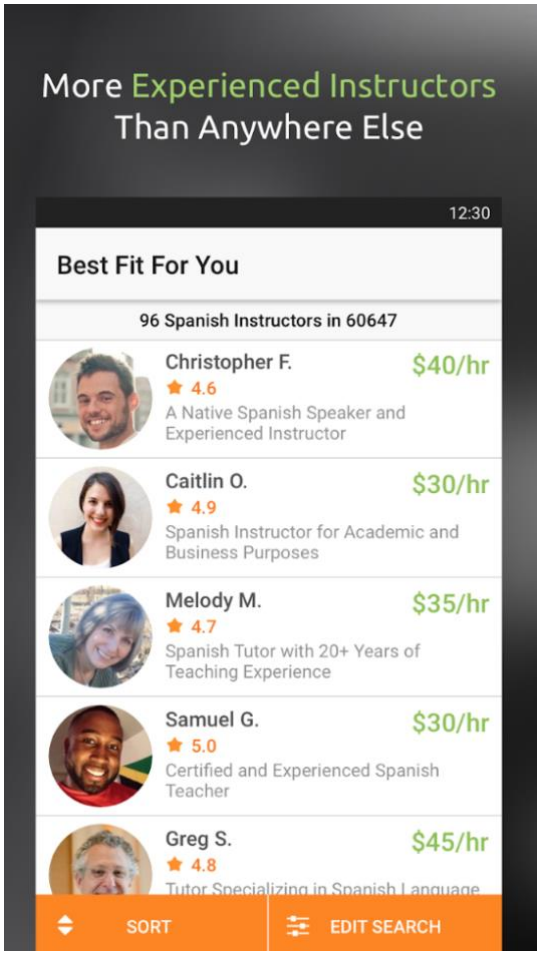

Figure 3. *Wyzant's main screen*[3]

1.2.4 Conclusion

After searching and analyzing applications in Google Play, the author concluded that most of these services use sites as their main platform for work, and their applications rarely become popular and used.

The author of this work sees Estonia as the main target of his application, as there is no solution specifically oriented to the Estonian market, or at least the author could not find any similar applications or web sites created for the Estonian market. Therefore, the author makes a hopeful conclusion that this application may find demand and acceptance among Estonian users.

[3] https://play.google.com/store/apps/details?id=com.wyzant.studentapp&hl=en_US

2 The design and architecture of Tutorbe mobile application

2.1 Methodology

During the development, the author used the Rational Unified Process (RUP) methodology, because it seemed the most suitable. It implies that it's impossible to take into account everything at the beginning, integration of functions is gradual, and software comes out with reduced functionality to occupy a niche in the market.

2.2 Requirements

2.2.1 Functional requirements

Functional requirements describe the services represented by the mobile application, its behaviour in certain situations, the response to certain input data and actions that the system will allow users to perform.

1. The user can create an account. The user must add his name, email, phone number and password to register the account.

2. The user can login if previously registered.

3. The user can enter under an anonymous account.

4. The user can add posts, which must contain a title, description, phone number or email.

5. The user can view the posts.

6. The user can delete his own posts.

7. The user can write an e-mail by clicking on the e-mail address in the post.

8. The user can call by clicking on the phone number in the post.

9. The user can sign out from his account.

2.2.2 Non-functional requirements

Non-functional requirements describe the characteristics of the system and its environment, rather than the behaviour of the system. Also, a list of restrictions imposed on the actions and functions performed by the system. They include time constraints, restrictions on the system development process, standards.

1. The application can run on all Android smartphones starting with version 4.1 and higher.

2. The user must have access to the internet for full use of the application.

3. The user should not have difficulty using the interface.

2.3 Application views

View class represents the basic building block for user interface components. A View occupies a rectangular area on the screen and is responsible for drawing and event handling. View is the base class for widgets, which are used to create interactive UI components (buttons, text fields, etc.). (View 2018)

The Activity class is a crucial component of an Android app, and the way activities are launched and put together is a fundamental part of the platform's application model. Unlike programming paradigms in which apps are launched with a main() method, the Android system initiates code in an Activity instance by invoking specific callback methods that correspond to specific stages of its lifecycle. (Introduction to Activities 2018)

The application consists of the following views:

1. User login

2. User registration

3. Open menu

4. Add post

5. Announcements board

6. Specific announcement

7. Splash screen

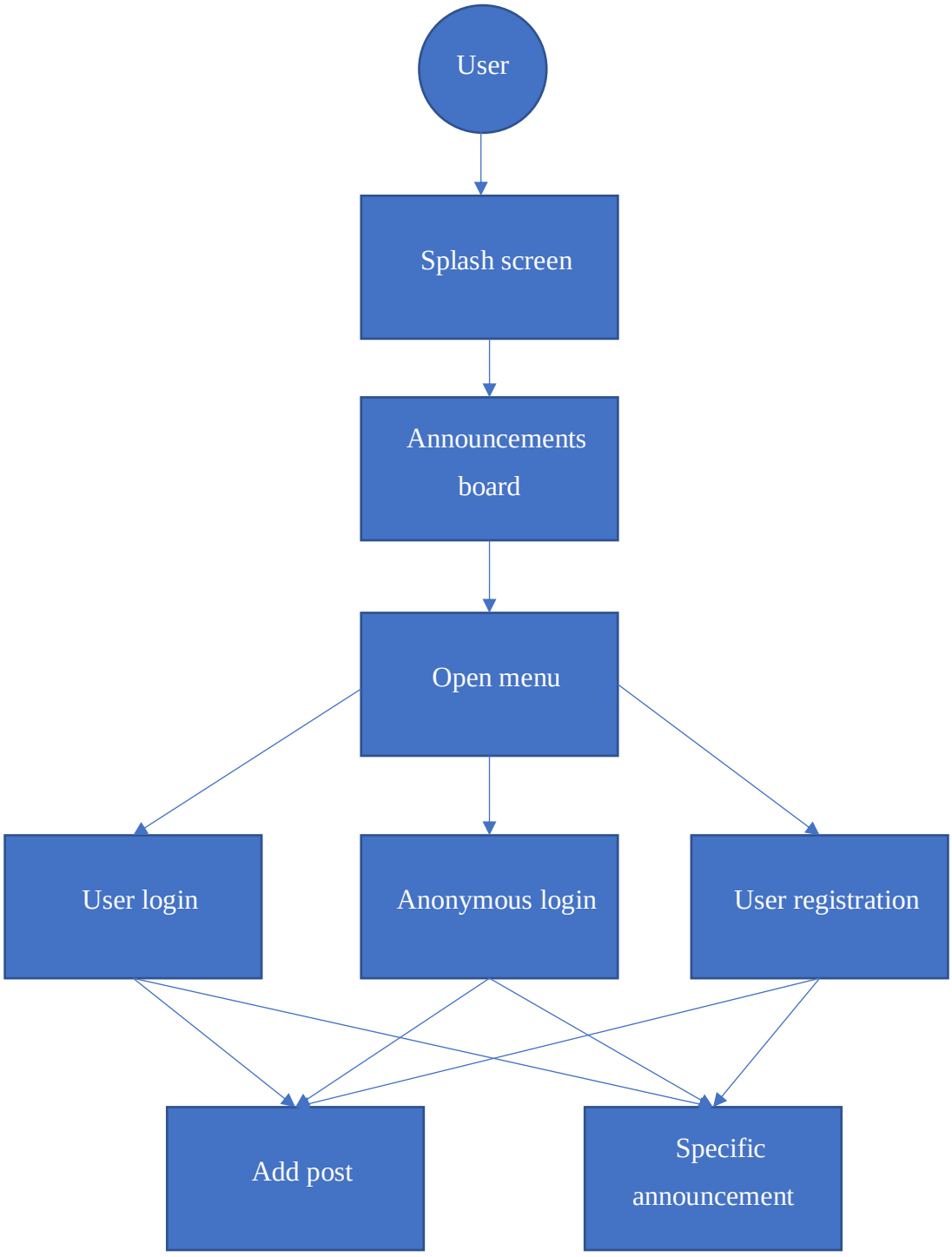

Figure 4. *View connections in Tutorbe*

2.3.1 LoginActivity

User login view is a view that is displayed if the user first uses the application or does not log in to the system. A user can log in to the system anonymously, or with an account created in the registration view. In the latter case, a user must enter an email address and a password.

If the user has not yet created an account, there is a button, which switches the user to the registration view.

The way the application will look was written in a document called activity_login.xml. This document is associated with the LoginActivity class in which Java code was written, with the help of which buttons and fields appear actions. The layout of the user login view is visible in figure 5.

LoginActivity – with his help the user logs in. It also includes checking whether the user exists, checking the correctness of the fields, anonymous login, login button and button to proceed to registration. The user authenticating through an anonymous login after the log out cannot delete his posts. About this user notifies the message appearing after the authentication.

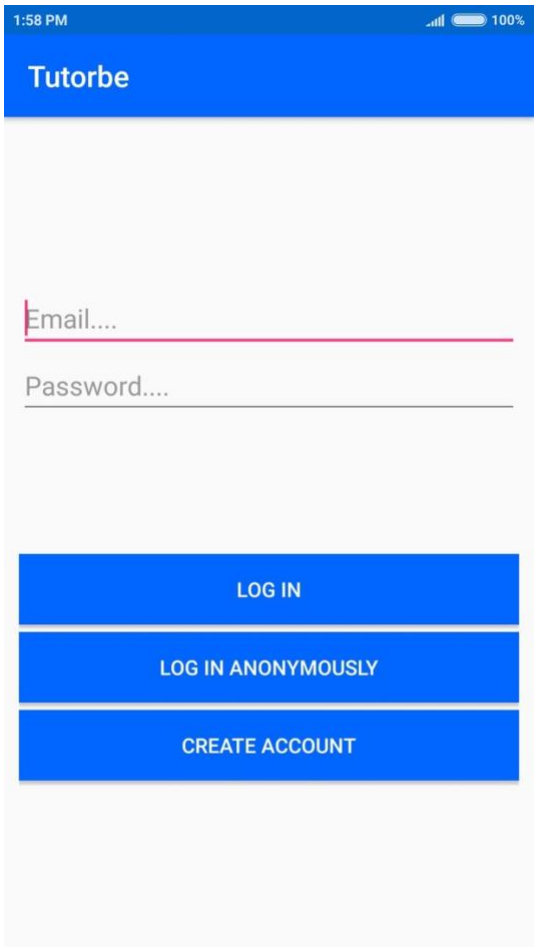

Figure 5. *User login view*

2.3.2 RegisterAcivity

User registration view allows the user to create a new account. The view is displayed after clicking the “Create account!” Text in the login window. When registering an account, the user must enter his name, email address, phone number, password and confirm the password. The e-mail address is used to enter the application, but it can also be used in conjunction with other users as a means of communication. All fields must be completed. The entered email is checked for correct input and must be unique. The password must be between 6 and 15 characters. The phone number must be between 6 and 8 characters.

The way the view will look was written in a document called activity_register.xml. This document is associated with the RegisterActivity class in which Java code was written, with the help of which buttons and fields appear actions. The layout of the user registration view is visible in figure 6.

RegisterAcivity – with its help, the user registers. It includes checking the correctness of filling the fields, the registration button and the login button.

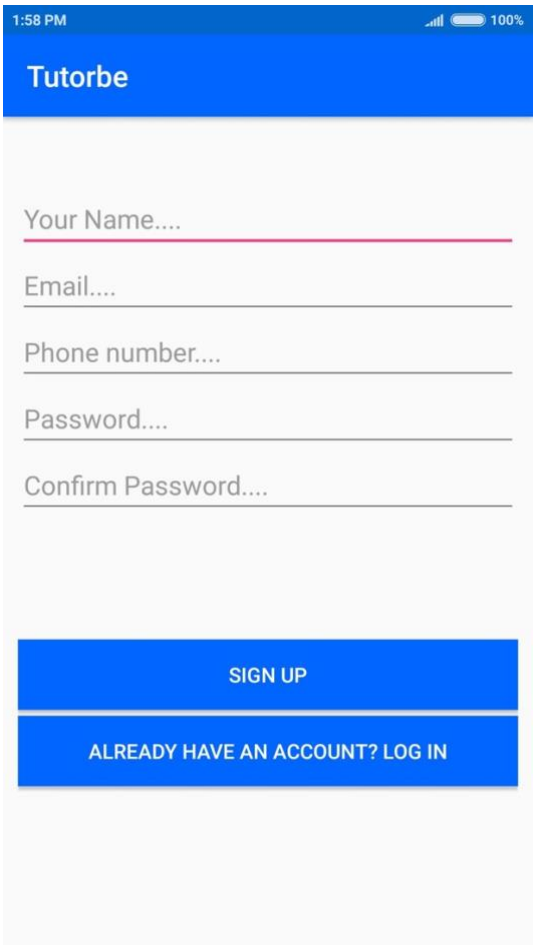

Figure 6. *User registration view*

2.3.3 MainActivity

After successful authorization, the user can use the menu, which is on the left and opens after clicking on the button from the top, or by swiping the finger on the screen to the right side. There are several items in the menu – “Add Post”, “Sign Up/ Log In”, “Log Out”.

The way the view will look was written in documents called activity_main.xml and nav_menu.xml. In activity_main.xml, the author used three widgets:

DrawerLayout acts as a top-level container for window content that allows for interactive "drawer" views to be pulled out from one or both vertical edges of the window. (DrawerLayout 2018)

NavigationView represents a standard navigation menu for application. The menu contents can be populated by a menu resource file. (NavigationView 2018)

RecyclerView is a flexible view for providing a limited window into a large data set. (RecyclerView 2018)

Nav_menu.xml contains the names of menus and icons embodying them. These documents is associated with the MainActivity class in which Java code was written, with the help of which buttons and fields appear actions. The layout of the open menu view is visible in figure 7.

MainActivity – with its help, the user can view ads and open and use the menu.

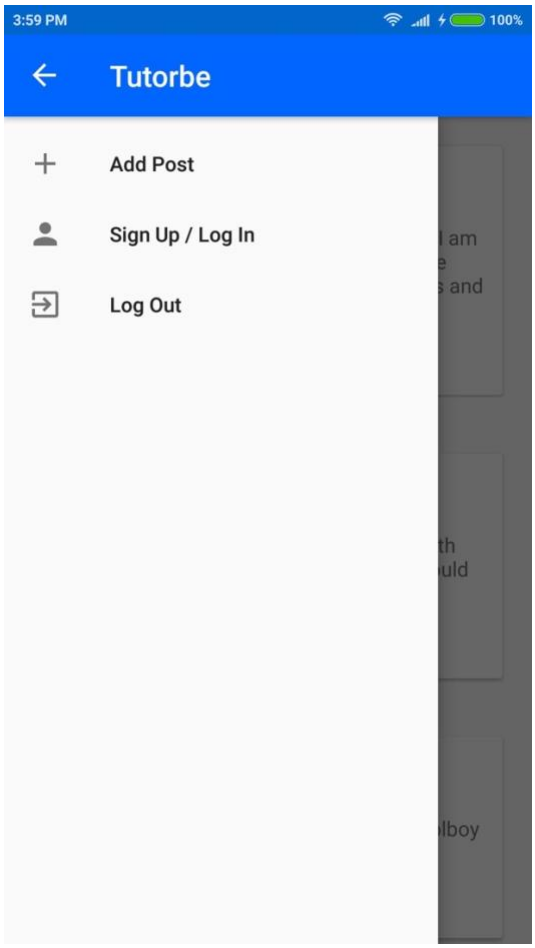

Figure 7. *Open menu view*

2.3.4 PostActivity

To add an post, user need to click on the button located in the menu "Add Post". A registered or anonymous user can add and remove their own ads. After clicking, the user will be redirected to the adding post view. When adding, user must enter a title, description, category, phone number or email, which will be used as contact data in the future.

The way the view will look was written in a document called activity_post.xml. This document is associated with the PostActivity class in which Java code was written, with the help of which buttons and fields appear actions. The layout of the add post view is visible in figure 8.

PostActivity – with its help, the user adds his post.

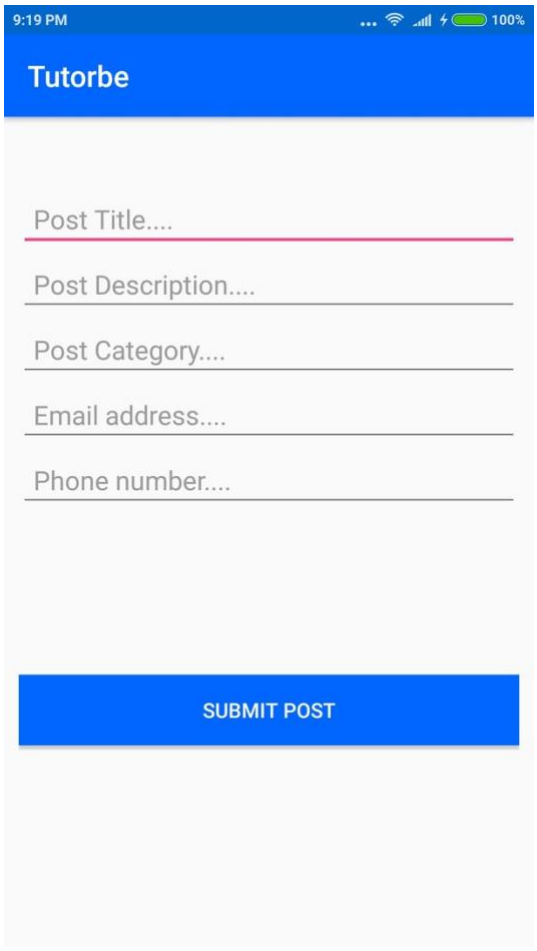

Figure 8. *Add post view*

The announcements board appears after the successful addition of the post in which a new entry appears. There will be displayed the title, description and user name that specified during registration. If the user logins anonymously and added an announcement, then instead of the name it will be written “Anonymous user”.

The way the view will look was written in documents called activity_main.xml and board_row.xml. Board_row.xml includes one widget:

CardView is A FrameLayout with a rounded corner background and shadow. (CardView 2018)

FrameLayout is designed to block out an area on the screen to display a single item. Generally, FrameLayout should be used to hold a single child view, because it can be difficult to organize child views in a way that's scalable to different screen sizes without the children overlapping each other. (FrameLayout 2018)

This widget helps to display individual news with the header, description and user name on the main page. These documents are associated with the MainActivity class in which Java code was written, with the help of which buttons and fields appear actions. The layout of announcements board view is visible in figure 9.

The board class is used to represent the users of the application and is associated with MainActivity. The class contains the following information about the user:

1) user's name

2) phone number

3) e-mail address

4) title of the ad

5) description of the ad

6) category

The name, phone number and e-mail address are necessary as a means of communication. The email address stored in the Board is not exactly the same as the one used to log on to the system. This is only the contact email address. Also, in this class are stored the title, description and category of the user posted advertisement.

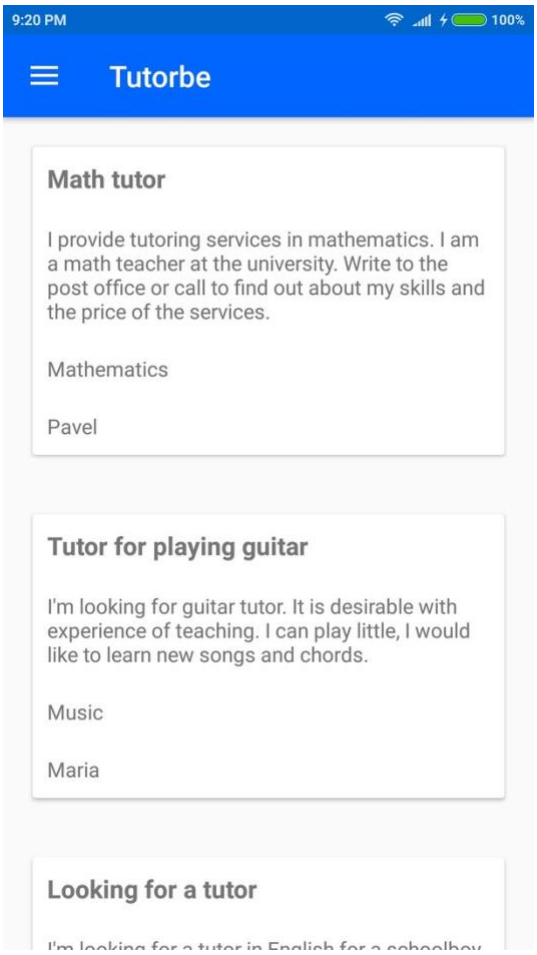

Figure 9. Announcements board view

2.3.5 BoardActvity

Specific announcement view is clickable. After clicking, the user will be directed to this post. In the announcement are the contact details of the user who added the post - email and phone number. After clicking on the email opens a standard Android dialogue to launch a mail application, after clicking on the phone number a standard Android phone call application is launched.

The way the view will look was written in a document called activity_board.xml. This document is associated with the BoardActivity class in which Java code was written, with the help of which buttons and fields appear actions. The layout of specific announcements view is visible in figure 10. Standard Android dialogue to launch a mail application is visible in figure 11. Standard Android phone call application is visible in figure 12

BoardActvity – with its help, the user can see how the post looks after clicking. It also includes a call to the number, after clicking on it, the button - Remove Post, which is visible only to the one who added the announcement.

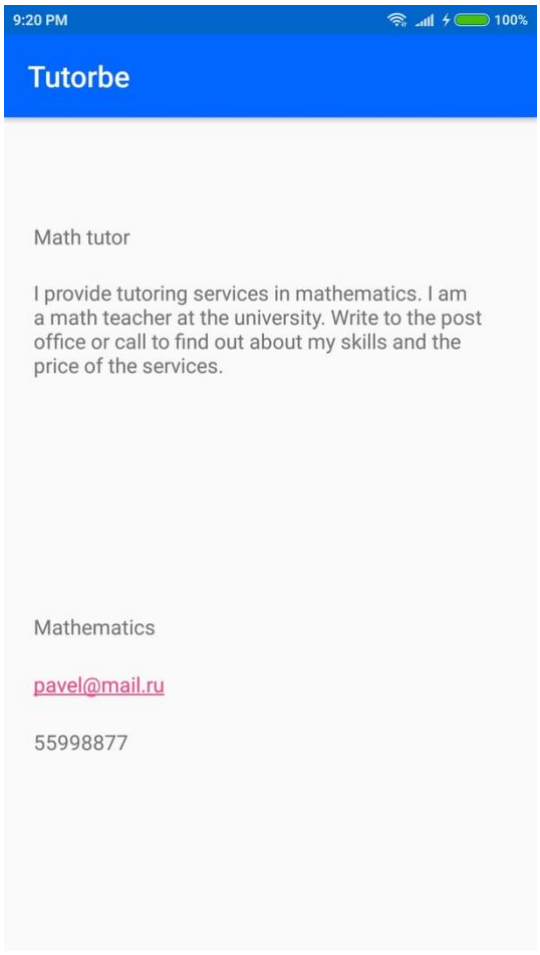

Figure 10. *Specific* announcements view

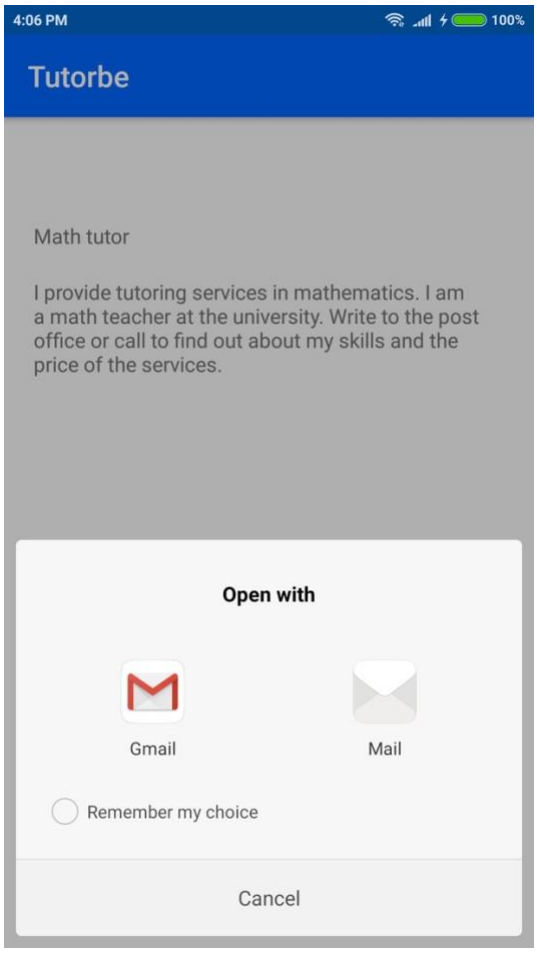

Figure 11. *Android dialogue to launch a mail application view*

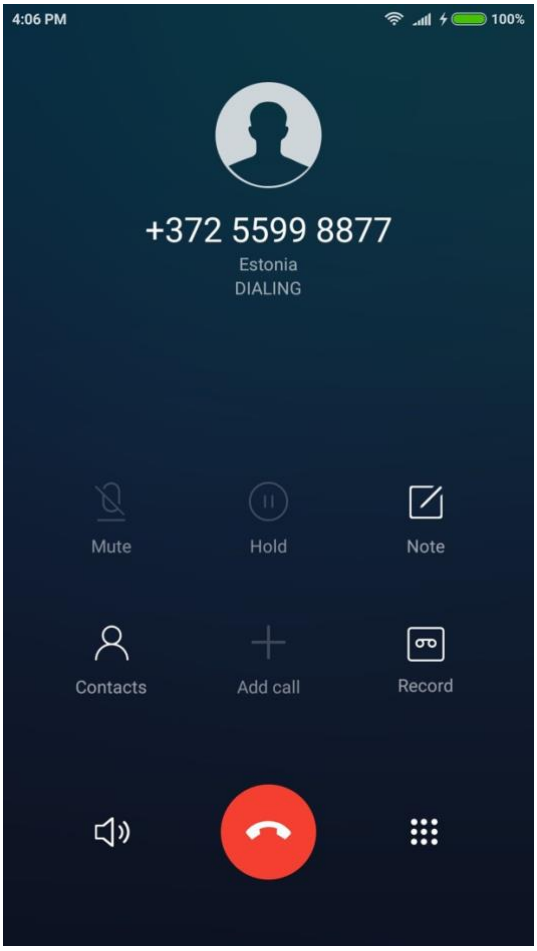

Figure 12. *Standard Android phone call application view*

2.3.6 SplashActivity

Splash screen is launched after clicking on the mobile application icon in the phone.

SplashActivity – with its help, the user avoids the delay, which is required when the application is launched. Instead, the application logo is displayed. The layout of splash screen view is visible in figure 13.

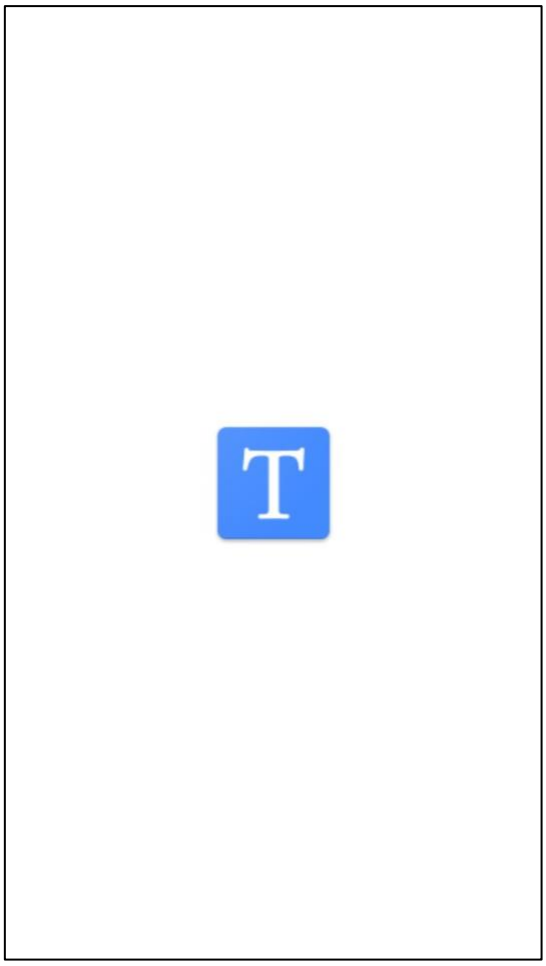

Figure 13. *Splash screen view*

2.4 Database architecture

The application consists of two components: the application itself and Firebase. The author uses Firebase Authentication and Firebase Realtime Database. Firebase

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more. (Firebase Authentication 2018)

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data. (Firebase Realtime Database 2018)

In the database there are two storages of the Board and Users. The Users field stores data specified by the user during registration. In the Board field, the ads they added. The ad and users are linked by user name.

Figure 14. *The database of Tutorbe*

3 User tests

The author of this thesis presented an application to a group of 8 people to test functional and non-functional requirements. All tests passed on one phone Xiaomi MI 5, Android 7.0, API 27. Most of the people being tested currently use the phone on the iOS operating system, but have experience using Android operating systems. The testers were all in the age group 19–25. Based on the requirements, the author set the tasks that each tester had to perform:

Run the application and get into the login form.

- From the login go to registration and register a new account
- Find and open the menu
- Add a new ad
- View existing ads
- Call and write to the email in the ad
- Delete your own post
- Sign out of your account

Each interrogated coped with the set tasks, no one had problems and did not cause difficulties to find the necessary buttons and fill in the necessary fields. Also, when testing, there were no errors. It was not obvious to several people that the fields with email and phone number are clickable, but in the end, none of the participants needed help in tasks.

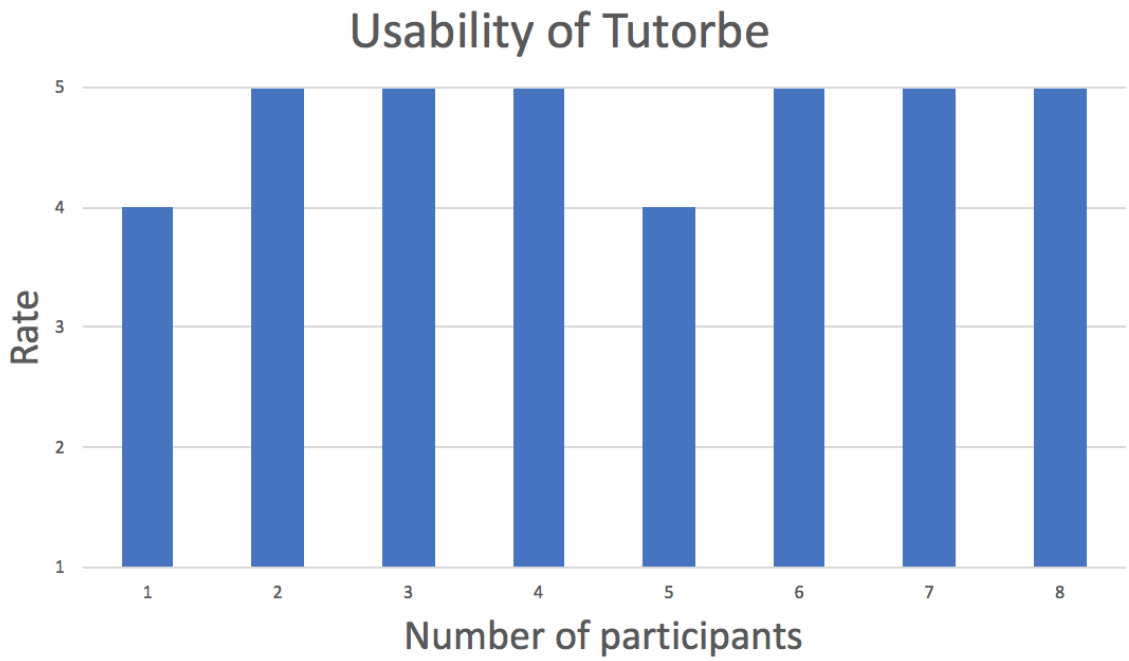

Figure 15. *Graph usability test of Tutorbe*

Conclusion

As a result, of the thesis, an Android application called Tutorbe was developed. In Tutorbe implemented login, registration. For successful login or registration all fields must be filled correctly, if an error occurs or the application is not filled correctly, the application will give an error. Registered users can add ads, view ads and delete their own ads. There is the possibility of anonymous login for users who do not want to spend time on registration or want to keep anonymity. Anonymous users have the same rights as registered ones, it is possible to add ads, view ads and delete their own ads. All ads can be clicked, after a click, user will be redirected to a new page, which will contain the full description of the ad, phone number, or email address of the user who published this ad. If user want to call the user click on the phone number specified in the ad. If user want to write to the e-mail address, click on the e-mail address specified in the ad and then choose how to send the message provided by phone.

All the basic functions needed in this application are present and ready to go. The application works stably, but has not been tested on other people, except the author and his computer, as well as the phone. At the end of this thesis, the author describes the future work that needs to be done to apply the application in production.

In the future I plan to add following features:

When adding an ad, users will have to choose the category to which their activity fits. It is necessary that in the future, users could search for ads by category, and not search for them manually. At the moment, Tutorbe exists only in English. In the future author would like to add Russian and Estonian. In Estonia most people communicate in Russian, but there are also Estonian-speaking users. Adding languages should help add an audience to the application.

Resümee

Väide tulemuses oli käsitletud Androidi rakendus nimega Tutorbe. Tutorbe'is realiseeritud login ja registreerimine. Eduka logimise ja registreerimise jaoks kõik väljad peavad olema õigesti täidetud, eksimuse tuvastamisel või vale täitmise korral rakendus annab vea. On olemas ka anonüümse logini võimalus nende kasutajate jaoks, kes ei soovi raisata aega registreerimisele ning säilitada anonüümsust. Anonüümsetel kasutajatel on samad õigused nagu tavakasutajatel - võimalus lisada ja ülevaadata kuulutusi ning enda kuulutuste kustutamine. Kõikide kuulutuste peale on võimalik vajutada ja pärast click'i teid suunatakse uuele lehele, kus on kuulutuse täiskirjeldus ja kuulutuse autori telefoni number või email. Soovil kohe helistada kuulutuse autorile, kasutajal piisab vajutusest antud numbrile. Soovil saata emaili kuulutuse autorile, kasutajal piisab vajutusest antud emailie valides saatmise võimaluse, mis on pakkutud kasutaja telefoni poolt.

Kõik vajalikud rakenduses põhifunktsioonid olemas ja valmis tööks. Rakendus töötab stabiilselt, aga polnud testitud teistel isikutel peale autori ja tema arvutit ning telefoni. Selle väide lõpus autor kirjeldab edaspidise töö, mis peab olema valmis rakenduse kasutamiseks tootmises.

Tulevikus mina kavatsen lisada järgmisi funktsioone:

Kuulutuse lisamiseks kasutajad peavad valima kategooria, millele nende tegevus sobib. On oluline, et tulevikus oleks võimalik, otsida vajalikku kuulutust kategooriate järgi, mitte käsiti. Praegu Tutorbe on olemas ainult inglise keeles. Pärast autor planeerib lisada vene ja eesti keelt. Kui Eestis (Ida-Virumaa maakonnas) enamus inimesi räägib vene keelt, aga on olemas ka eesti keele kõnelejaid kasutajad. Siis keelte lisamine peaks suurenema rakenduse kasutajate arvu.

References

Smartphone definition 2010. Available at https://techterms.com/definition/smartphone, accessed May 12, 2018.

Singh, Rajinder 2014. An Overview of Android Operating System and Its Security

Features. Available at https://pdfs.semanticscholar.org/11f4/b8efd1a9af746f17ac5e8d6a789bd3c3a9b7.pdf, accessed May 12, 2018.

Smartphone OS 2017. Available at https://www.idc.com/promo/smartphone-market-share/os, accessed May 12, 2018

Meet Android Studio 2018. Available at https://developer.android.com/studio/intro/, accessed May 12, 2018.

Gosling, James; Joy, Bill; Steele Guy; Bracha Gilad; Buckley Alex; Smith Daniel 2018. The Java Language Specification Java SE 10 Edition. Available at https://docs.oracle.com/javase/specs/jls/se10/jls10.pdf, accessed May 12, 2018.

Extensible Markup Language (XML) 1.0 1998. Available at https://www.w3.org/TR/1998/REC-xml-19980210.pdf, accessed May 12, 2018.

Gradle 2018. Available at https://en.wikipedia.org/wiki/Gradle, accessed May 12, 2018.

Kohl, Wilfried n.d. Backend As A Service Using the Example of Enginio a Cloud Service for Qt Available at https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.200/files/staff/K%C3%A4chele/kohl14backend-as-a-service-enginio.pdf, accessed May 12, 2018.

Build an Android App Using Firebase and the App Engine Flexible Environment 2018. Available at https://cloud.google.com/solutions/mobile/mobile-firebase-app-engine-flexible, accessed May 12, 2018.

View 2018. Available at https://developer.android.com/reference/android/view/View, accessed May 12, 2018.

Introduction to Activities 2018. Available at https://developer.android.com/guide/components/activities/intro-activities, accessed May 12, 2018.

DrawerLayout 2018. Available at https://developer.android.com/reference/android/support/v4/widget/DrawerLayout, accessed May 12, 2018.

NavigationView 2018. Available at https://developer.android.com/reference/android/support/design/widget/NavigationView, accessed May 12, 2018.

RecyclerView 2018. Available at https://developer.android.com/reference/android/support/v7/widget/RecyclerView, accessed May 12, 2018.

CardView 2018. Available at https://developer.android.com/reference/android/support/v7/widget/CardView, accessed May 12, 2018.

FrameLayout 2018. Available at https://developer.android.com/reference/android/widget/FrameLayout, accessed May 12, 2018.

Firebase Authentication 2018. Available at https://firebase.google.com/docs/auth/, accessed May 12, 2018.

Firebase Realtime Database 2018. Available at https://firebase.google.com/docs/database/,accessed May 12, 2018.

Appendices

Link to download Android application Tutorbe:

https://drive.google.com/open?id=1Y-ubl_6SuKwTHxY6kmSCesmsHNJqnO8H