UNIVERSITY OF TARTU

FACULTY OF SOCIAL SCIENCES

NARVA COLLEGE

INFORMATION TECHNOLOGY SYSTEMS DEVELOPMENT

Santeri Pohjaranta

CREATING AN ANDROID APPLICATION FOR ACQUAINTANCE WITH HISTORICAL BUILDINGS AND PLACES

Diploma thesis

Supervisor: Andre Sääsk, M.Sc.

Narva 2021

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, Santeri Pohjaranta (date of birth: 19.08.1998),

1. here with 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 for acquaintance with historical buildings and places” supervised by Andre Sääsk M.Sc.

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, 17.05.2020

TABLE OF CONTENTS

TERMS AND ABBREVIATIONS

OS – Operating System, software which connect user interface and hardware of computer and schedule task of computer system.

AR – Augmented Realty, technology which generate image by computer or another technological device on a user's view of the real world.

API - application programming interface, it is interface for calls. If someone like to use data from server, then API key tell program which data it can collect and use.

URL - Uniform Resource Locator, represent itself as web address which show path until specific place in computer network.

SDK - software development kit, is a set of development tools used to develop applications.

XML - Extensible Markup Language.

Data - are units of information, often numeric, that are collected through observation. In a more technical sense, data are a set of values of qualitative or quantitative variables about one or more persons or objects, while a datum (singular of data) is a single value of a single variable. (Data 2021)

Field – Field is places where user can write specific data in specific place which programmer set up.

OOP – Object oriented programming.

Pop-up –screen which show additional information on main screen.

DFD – Data Flow Diagram.

Card – card show user image of place in tour and by clicking on that user receive information about place if this point is next in user tour.

Navigation Drawer – drawer which show menu of navigation to different screens in application.

Status Bar – Bar where user can find status icons on phone.

Navigation Bar - menu that appears on the bottom of your screen, show user screen name in application.

INTRODUCTION

The Problem

In our technological world we have features which help learn different things in short time. Today people can learn history through internet by reading different articles or science works. But with all this stuff sometimes some historical place does not get attention by tourist or another people which like to receive information about history of town. In World War Two many historical places and architecture was destroyed or been damaged and today we cannot see them as they were. This is not only circumstance which show what happened with historical places and buildings, but this is great example how they change appearance in a period of world history. Towns and cities across the world have lost valuable history due to various circumstances. As an example, Narva, in Estonia's eastern region has lost its old town, which result a low level of tourism to such a historically important place.

The solution

Using existing technology, author can create application which show people interesting places in town by tour and give user interesting information about historical place or building. Also, people can receive information from tour in gaming format, as example when user start tour there can be questions or letters about places which are not marked on map and when user reach that point, he receives letter or question which give him points.

The aim

Author aim is creating application in which user can choose historical tour and receive information about historical place or building with AR, text, image, audio.

This application should do:

- Login into system.
- Register user in database.
- Login anonymously.
- Application needs to show user different tours.

- Need to show user location on map.
- Need to set automatic path between user and points on map.
- Need to show user story about building or place by clicking on place card when user is 50 meters nearby.
- Registered user can choose favorite tours.
- Show to user in AR 3D models.
- Need to show questions which user can answer and receive points for each right answer.
- Language support: English, Russian, Estonian.

The motivation

The author has interest to raise cultural and historical level of native people. Also, author of this thesis like to apply his knowledge in field of mobile application creation.

Tasks

To reach the aim of this diploma thesis, the author must complete the following tasks:

- Choose appropriate tools and technologies for developing the application.
- Choose an API which can read a map.
- Create test tours which show how the application would work.
- Create architecture of the database.
- Create user interface mockup.
- Develop the application.
- Test and debug the application.

Outline

Theses consist of three chapters. First part is introduction problem description, resolving of problem and what is author goal and which tasks author need to reach in this work.

Second part describe technology which was used in application creation, instruments which help to integrate technology in application, similar solutions which is already on market but can be improved by integrating functions which is not yet used in similar applications.

Third part introduce functional and not functional requirements, early prototyping of application, data flow diagram and application mockup, main work with code and comments which describe each method and class which application use in process of work and conclusion about work which described in diploma work and final review of work by author.

1. IMPLEMENTED TECHNOLOGIES AND AVAILABLE SOLUTIONS

1.1 Early Solutions

1.1.1 Quest city

Quest city is application for creation and finding for user tour which he/she like or quest rooms in his/her town. This application help user to find interesting tours in your town and give opportunity to create their own tours and receive from that money.

This application has some similar idea with this application, but the author of application directed for education in history. Also now Quest City application does not work, and author do not know if they plan to place new application for tours or not.

1.1.2 Lokimo

Lokimo is app where user can make tours of historical and interesting places in town. User who likes to go through tour which he/she is interested download that tour and start it. Also, this application provides tours which user can go through in his/her home.

This application is directed competitor of author application. Lokimo application has an excellent system to add tours, but author application is directed to tell people story about historical place and show with AR how this place or building was in old times.

1.1.3 ShortWalk

ShortWalk is application which give user random tour on map with time which user set.

Application has similar functions with author application, but ShortWalk application give user already created map with points based on time which user set in application, but in author application user can choose which tour he/she like to go through. Also, from discussion with creator of this application author understand that application is not working, and additional support is stopped.

1.2 Technologies

Now we have some examples of applications which have same idea but have different realization. Now we can talk about technologies which can be used in application, and which help author to reach thesis goal.

1.2.1 Android

Android is OS which was developed by Google company for smartphone and tablet devices. Today this is one of the most popular OS for smartphones.

Considering that Android is one of the most popular operating system for mobile devices such as smartphones and tablets and that Google company give instruments for developing application on Windows OS like Android Studio for devices with Android OS it is an obvious and logical choice to pick it for the development of the mobile app. Also, on global market Android OS have nearly 72% of usage base on statista.com research, that give opportunity for testing application on more users.

1.2.2 Android Studio

Android Studio is integrated development environment (IDE) for developing mobile application on Android OS which can be used on Windows, Linux, and Mac OS. Itself Android Studio is built on JetBrains' IntelliJ IDEA software but designed for developing applications on Android OS.

Author chose Android Studio as development platform to create application for android because this platform officially supported by Google and author already know how to work with it.

1.2.3 Java

Java is object-oriented programing (OOP) language which can be used on every device which support this language.

Object-oriented programming is a programming paradigm based on the concept of "objects", which can contain data and code: data in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods). (Object oriented programming, 2021)

Java is one of two supported programing languages which android use in app development. Second is Kotlin. For now, author use Java as main programming language because author already know how to work with this programing language. Java is popular object-oriented language and Android Studio community already have many solutions for resolving problems in code on Java than on Kotlin.

1.2.4 Firebase

Firebase is platform for development environment for application on Android OS. Firebase can host database and storage for app and give developer instruments for testing and deployment application on Android applications market.

Author used Firebase to host application and receive access to database which will hold information about user, information about tours and their default settings such as: time of went through tour, picture of tours and color. Also, author use analytics to collect data about users. Firebase has already database and storage hosting, also their integration in app and working with them is simple because: first this is Google official product which can be simple integrated in android app, second android community have already tutorials and different information how to work with it.

1.2.5 Google API

Author used Google API because it has big library of different type of data which can be used in application. Also, it gives author a good price for application testing with Google API.

1.2.6 XML

Extensible Markup Language (XML) is text-based markup language which base function mainly focuses on transfer of data through system and represent result which system give.

XML is base language to create layouts in Android Studio. Because of well-structured format XML code is easy to read and write.

Author chose XML because it is base language for creating layout on Android.

1.2.7 Picasso

Picasso is a plugin for image rendering in Android application by writing them in XML file.

Author does not recommend for using this plugin because it has a problem with automatic sizing of image file in application.

1.2.8 Glide

Glide is an Image Loader Library for Android developed by bumptec. It has been used in many Google open-source projects including Google I/O 2014 official application. It provides animated GIF support and handles image loading/caching.

Author user this plugin because this plugin has automatic sizing in of image in application and have updates which indicate about supporting this plugin and working on newer versions for Android Studio.

1.2.9 SparkButton

SparkButton is library that allows you to create a button with animation effect, such as heart or like button.

SparkButton can specify both active and inactive image of the button. If only active image is specified SparkButton will behave as a normal button, otherwise as a switch.

Author used this plugin for creating like button, because it has simple functionality and images which can be used as like button or another button which need animation requirements.

2. Design and architecture of application

2.1 Requirements

2.1.1. Functional requirements

Functional requirements describe a service which give output or input from system based on situation which it is been programmed:

1. User can login or register in system. Need to write email and password.
2. User can go anonymously into app.
3. Show current location of user.
4. When user is near than 50 meters with place on map, he/she can see information about historical place.
5. User receive questions about text which he/she read about place or building. User receive 2-4 answers which he/she can choose which is correct. If user answer correctly, then he/she receive points on account and go to next point. If user answer wrong, then it receives second chance to answer. If second time is wrong, then user go to next point and did not receive points.
6. User can see 3D models in AR. When user open information about point on bottom of text placed button for AR function. Buttons appear if point have link to 3D model in database. If user click on that button, then application send user to screen where user first need to download model and then click on place where user like to place it.
7. See favorite tours which user like to go throw again.
8. Create menu where user can choose different options like favorite tours, change language.

2.1.2. Non-functional requirements

Non-functional requirements describe how application should look. Also, it includes suggestions and restrictions which client give to developer to develop application for best user experience on client wish.

1. Application needs to work on Android smartphone which has version 5.0 and higher.
2. Should not contain Status Bar and Navigation Bar.
3. Interface need to be simple and understandable for user.
4. Should show information on English, Russian and Estonian.

2.2. Design

In this section author show how application is designed and which methodology is used for creating application.

2.2.1. Use Case

Use Case diagram is diagram which show relationship between different actors with system.

Use Case diagram does not show how system would work in totally it only summarizes some of the relationships between use cases, actors, and systems.

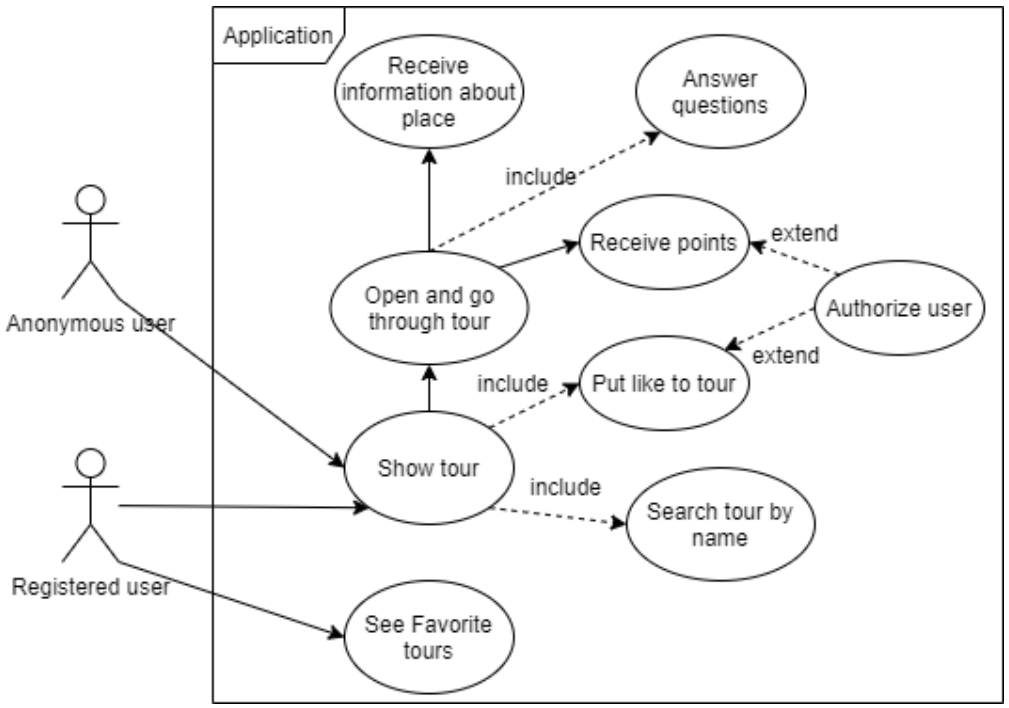

Figure 1 Use case diagram.

In this Use Case diagram author represent relationship with application and with its functions. Here we have authorized user and unauthorize user as actors which use this application. Both users can see tours from database, but only registered user can see favorite tours which is on his account. When user see tours on main screen it has additional functions such as searching tour by name or put like to tour but this function can use authorized user. When user choose tour from list user open tour and go through tour and receive information about places. For additional in tour can be questions which user receive. For each point in tour user receive points but his points can receive only authorized user. Diagram is represented in **Figure 1** Use case diagram.

2.2.2 Data Flow Diagram (DFD)

Data Flow diagram show process of work in system by each step. In references you can see rectangle, rounded rectangle, rhombus, and arrows which represent each element in system. Rectangles represent work of application, rectangle with rounded angles is user inputs or user activity in application. Rhombus represents “if” statement and arrows application next step. Because of complex representation of system diagram was split to three parts. First is login and registration to system, second is main screen with tours and other functions which can be found in there. Third function

represent situation when user choose tour and start it. Diagram is represented in **Figure 2**, **Figure 3**, **Figure 4**.**Figure 4** Data flow diagram of tour system.

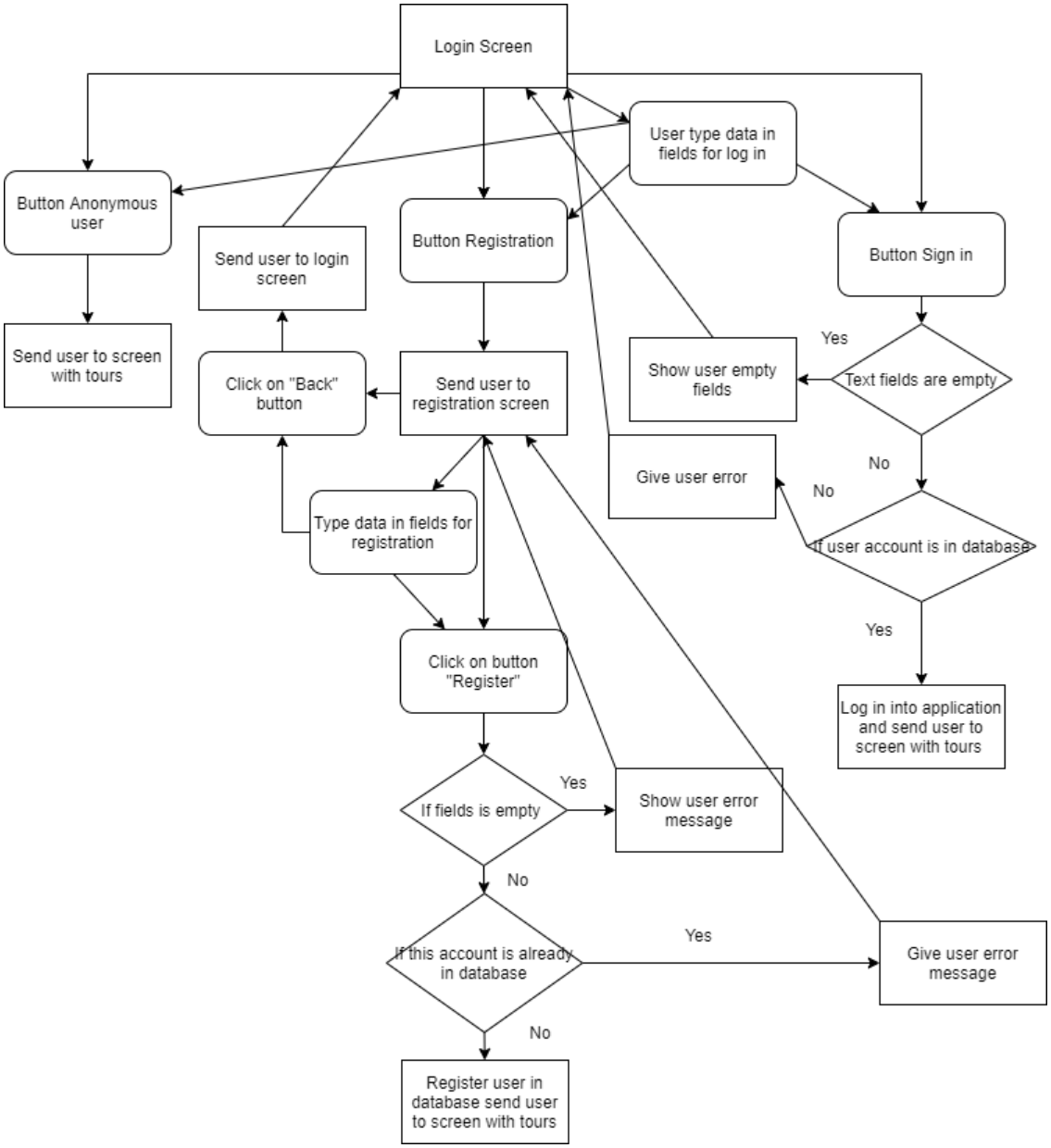

Figure 2 Data flow diagram of log in and registration system.

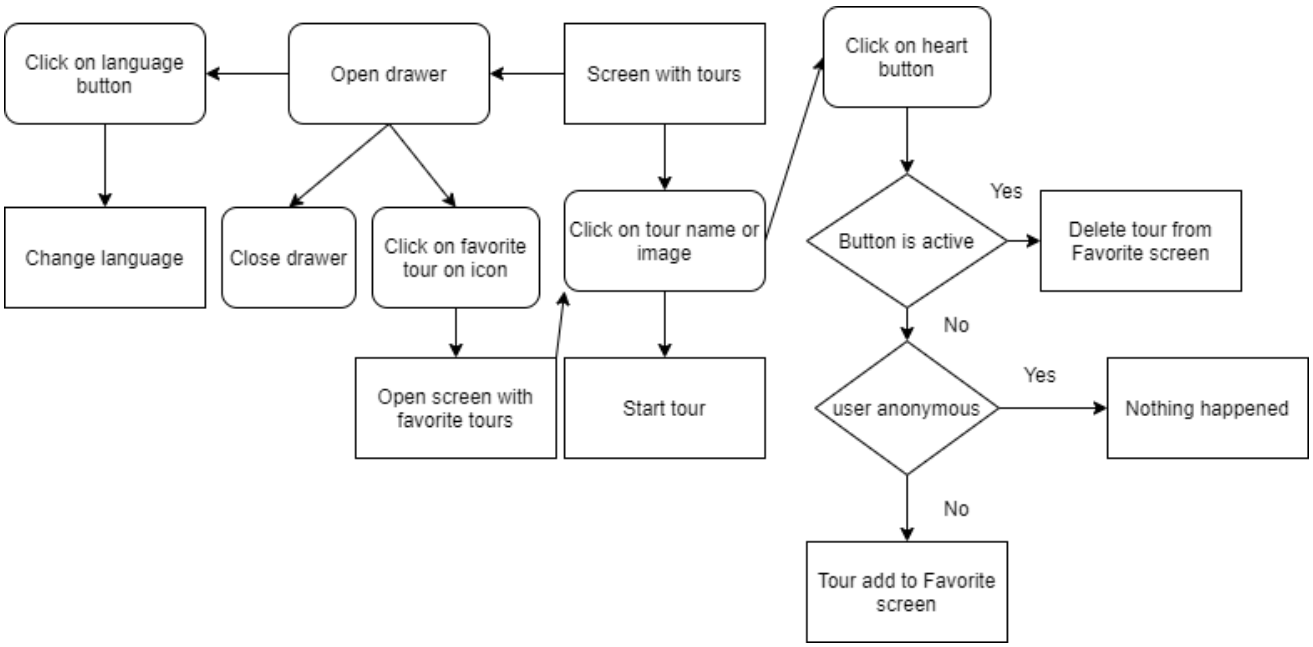

Figure 3 Data flow diagram of choosing tours system and favorite tours system.

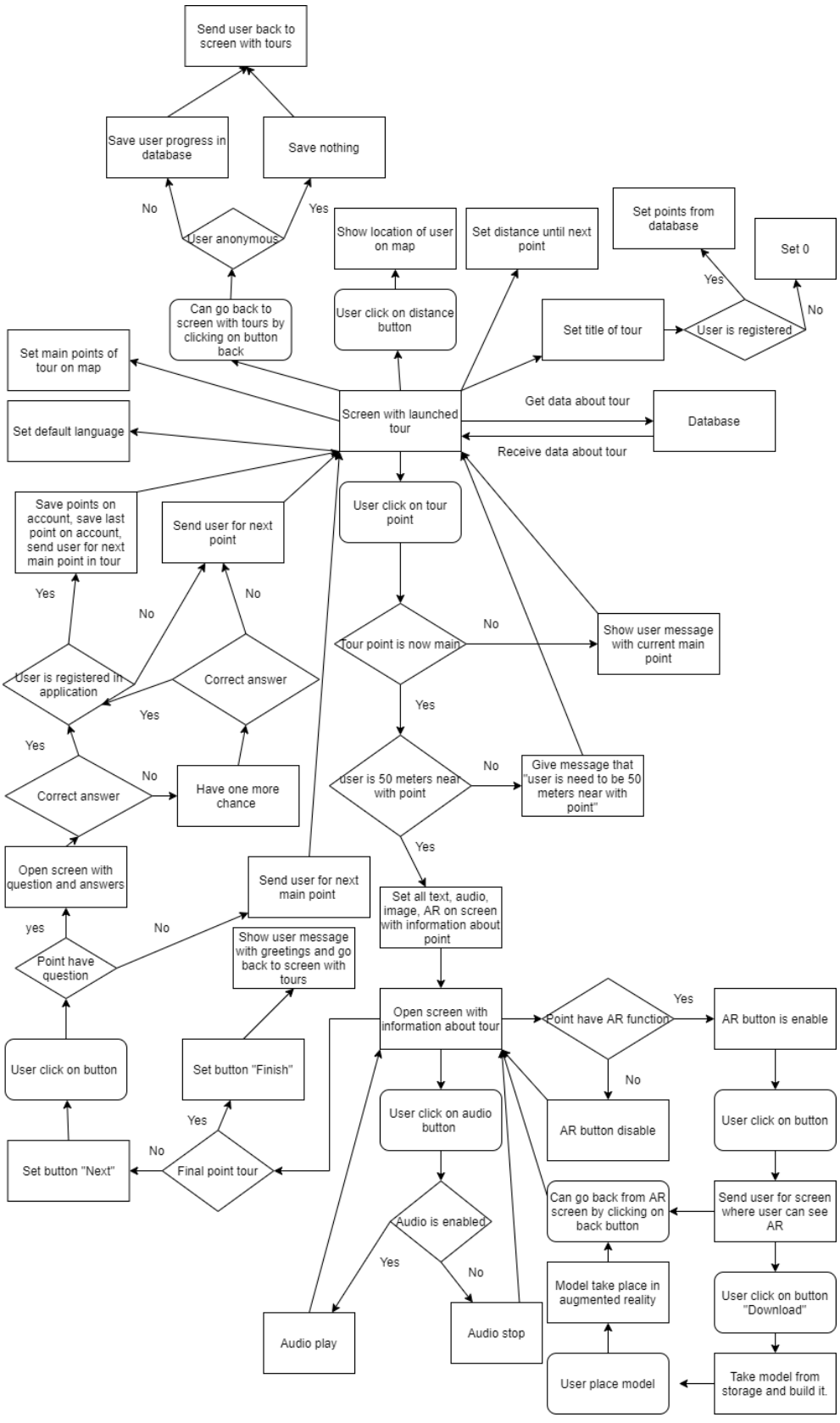

Figure 4 Data flow diagram of tour system.

2.2.3 Database design

Database design show how database is built and how data is linked with other tables. This is needed for understanding where data would hold and how data would be placed in system.

Database has two main tables this is tables with tours which hold information about tour and table with user which have information about users and tour names which user have started or gone through. Table with tour hold main coordinates table which hold coordinates of main points of tour, and all needed to represent user information about place when he/she reach it. With that database have SubCoordinates table which represent coordinates for creating path between points, Question table with questions which user need to answer if main point has question number in table. Languages table need for audio and text translation on tour different points.

User table need for collection of data about user and create saving for user progress in database. Database representation shows on **Figure 5**.

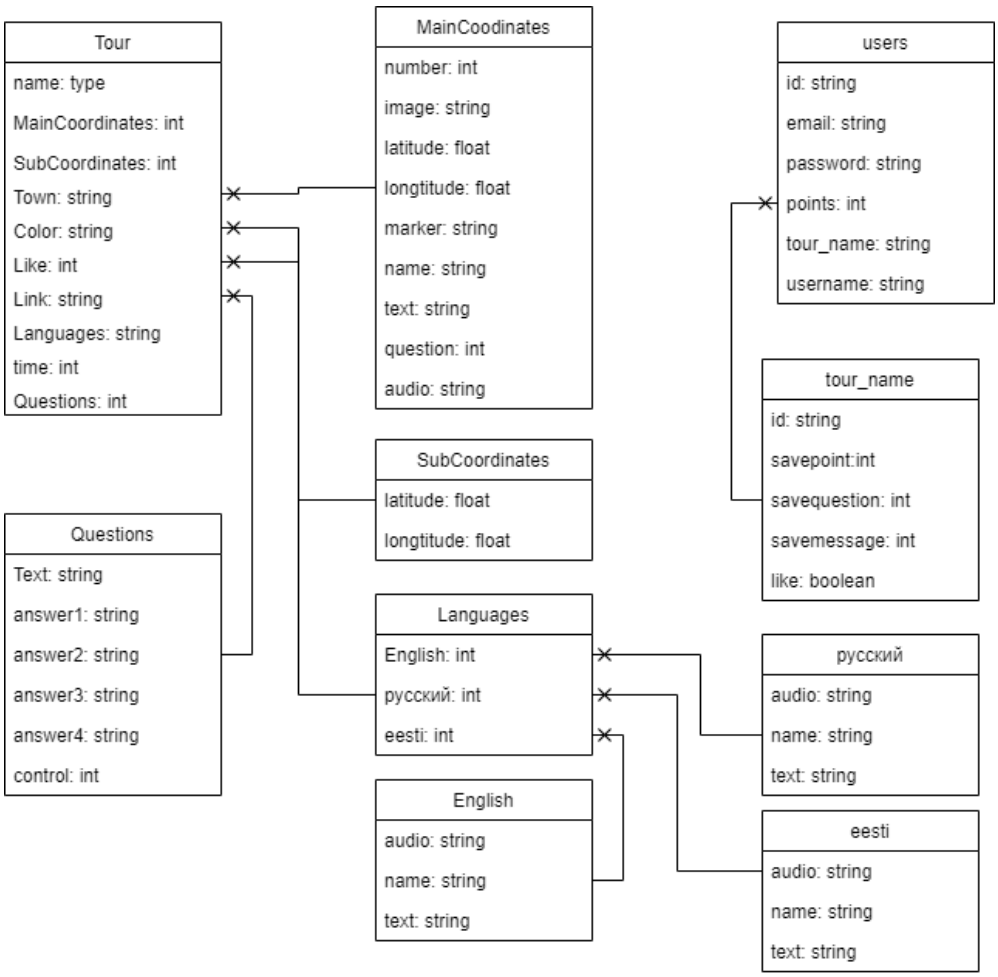

Figure 5 Application database design.

2.2.4 Application Mockup

Now when author have understanding how relationship with users work in application, now author can start to work on mockup of application to visualize application work.

Application Mockup show application design and visualize how app prototype can look and work in different situation.

Here we start from screen with app logo and then go to login screen, in another situation if user already did login previous time, then user login automatically. On login screen we have 3 options: login into system, register into system and go as anonymous user or with other words as not registered user. If user login into system, then user need e-mail and password which is already registered in system. If boxes with email or password is empty when user click on login button, then application show user

error with boxes which are empty. If user put all data right, then user go to main screen with tours. If user like to register in system, then user click on register button and user go to registration screen. If user put data and click on button to register itself into system, then if user with same data already registered into system, then application give error, if boxes with text is empty then user show error message with boxes which are empty. If all data is right, then system register user and send it to tour screen. Next time user can login with registered account. User can also choose to login as anonymous, and application send user to screen with tours with limited functionality. Also, on that screen user can choose change language by clicking on language button. When user open application first time application set phone language as default language. When user change language in application then it also saves it in application system. Representation of mockup can be found on **Figure 6**.

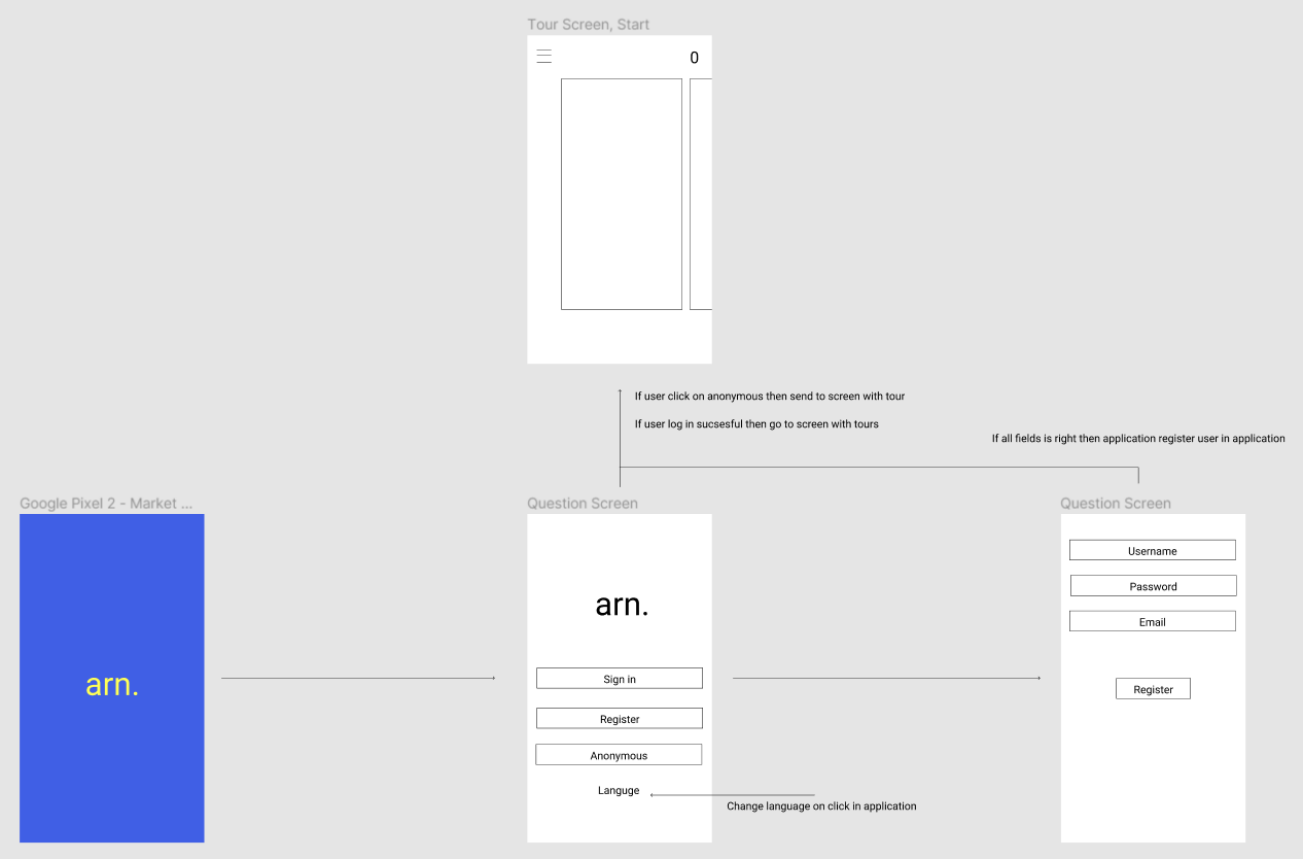

Figure 6 Mockup of log in and registration screens.

On screen with tours user can choose tour from list. Authorize user can choose to like tour and application send information about that tour to favorite tour screen. This tours user can see when user open navigation drawer and choose “Favorite tour” screen. Also, in navigation drawer user can found

button for changing language and log out button. When user click on default language then application change language which is in list next. By clicking on “Favorite tours” button in navigation drawer user can see tours which he liked by like button. If user click on like button again then application remove tour from “Favorite tour” screen. Access to this screen have only registered users. From this “Favorite tours” screen and from screen with tours user can choose a tour and click on tour image or name. Representation of mockup can be found on **Figure 7**.

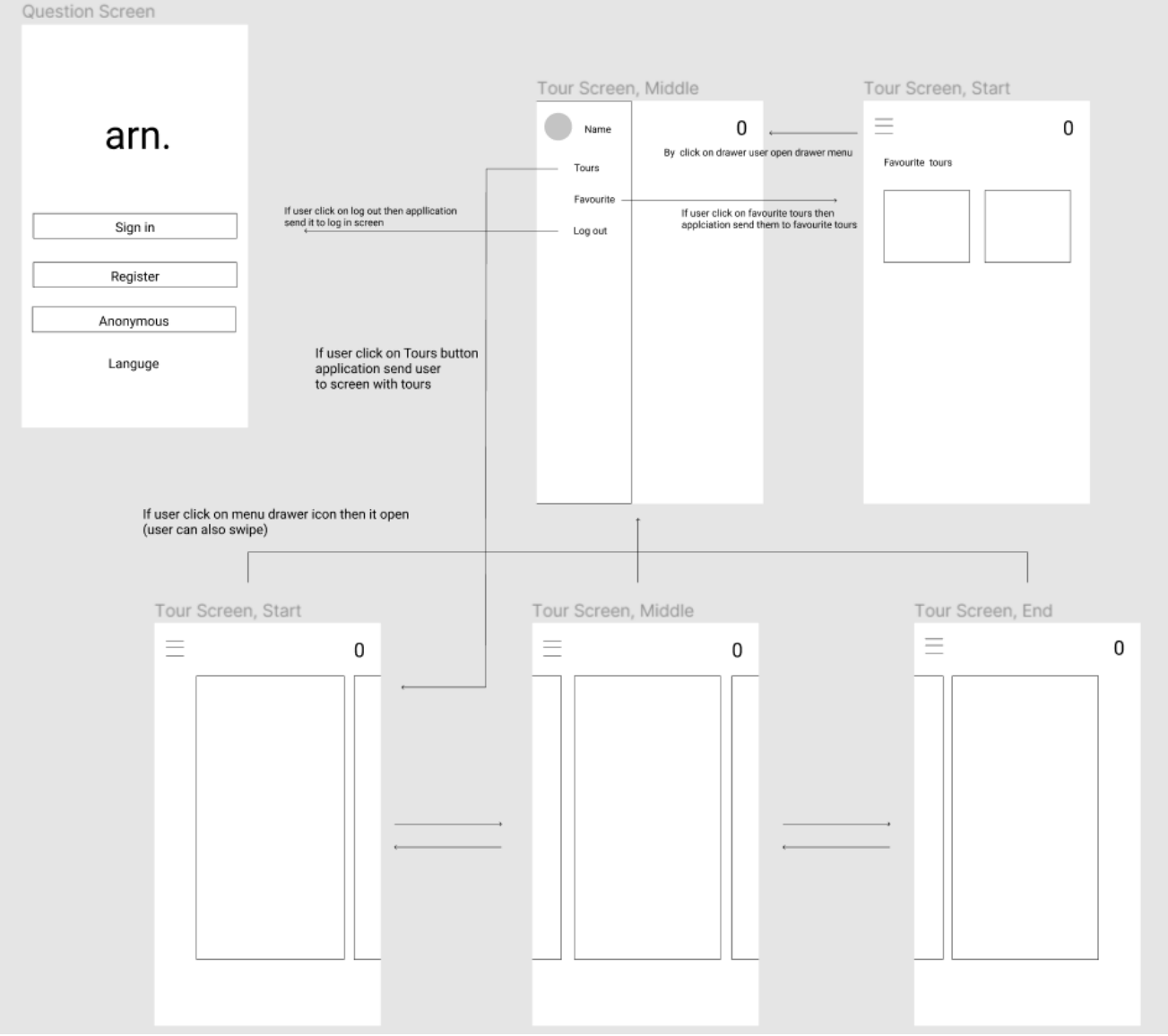

Figure 7 Mockup of screen with tours and navigation drawer.

When user click on tour image or name, application change screen and show user map with points of tour and path between them. With that on top of screen user can see title of tour, points which user

receive by activities in tour and current point where user need to go. When user reach one of tour point then by click on that point image application open pop-up with information about that point with audio, image, and text. If point of tour has button with eye in right bottom corner, then user can click on that and application send it to screen with AR functionality. When user is there, user needs to click on button to download model. When model is downloaded in application user can place it on surface where he/she likes. The model stay on place where user is set it and user can go around that model and see it from different corners. When user click on button next then it shows user next location where user need to go. If point of tour has question, then application open on-screen pop-up with question which user can answer. If user choose wrong answer, then user receive message that he/she have one more chance to answer question. If question answer is correct, then user receive point if user answer wrong user does not receive points. Representation of mockup can be found on **Figure 8**.

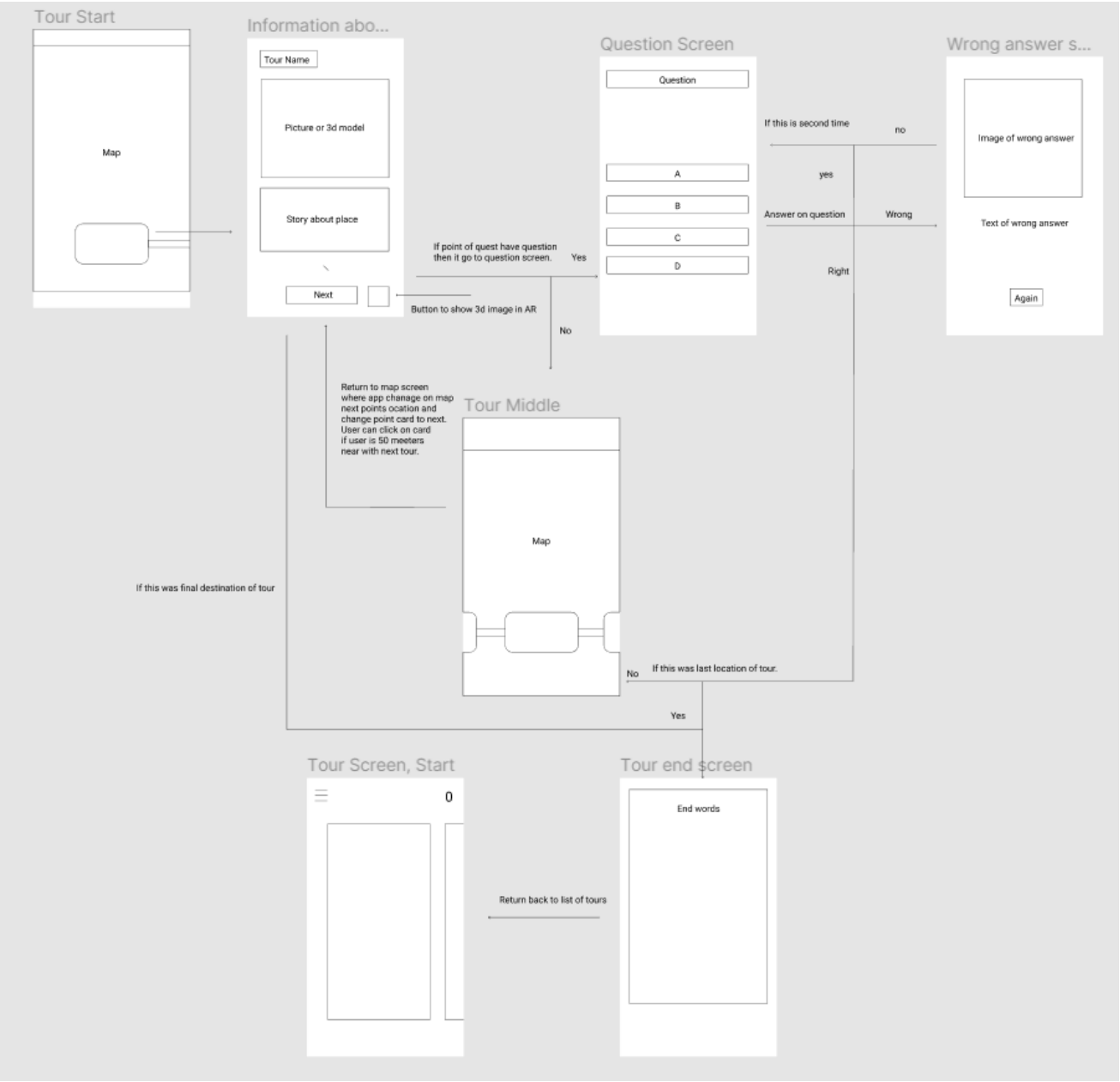

Figure 8 Mockup of tour start.

2.2 Early prototype of application

2.2.1 Login View

On early stage for application prototyping was used Adobe XD. Adobe XD is good application for prototyping because when you create application mockup you can test it with android and IOS on how it looks and test base functions.

On early-stage login in screen contains text with hello words to user, two text field for e-mail and password to login into system, button which login user if e-mail and password was correct and if user forgot password, then use can click on it and then it sends user to view where user can write his e-mail to send password to e-email if database have account linked with this e-mail. How prototype looks like you can see **Figure 9** User login view prototype.

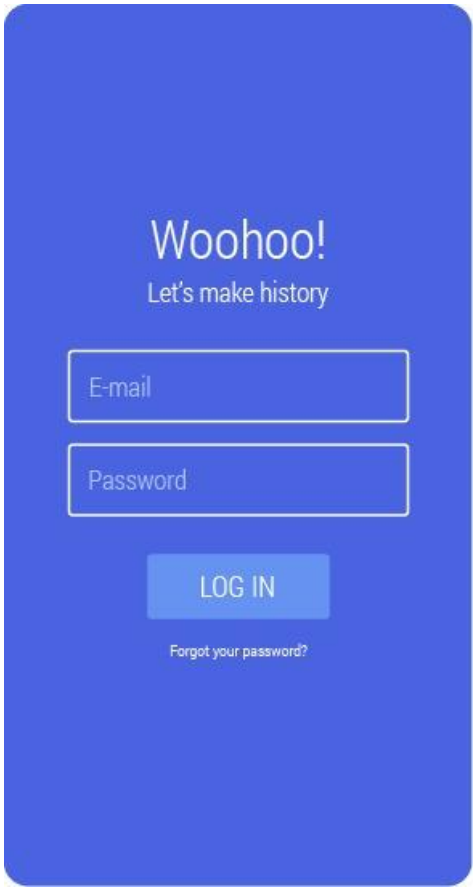

Figure 9 User login view prototype.

2.2.2 Map View

On early stage of prototyping map screen there was google map and automatic paths between points, name and how much kilometers need to go. When user is near with point map automatically change size and show name of place and history of this place. Also, on this screen author planned to put buttons for home screen, then for tours screen and for profile screen. Layout of this screen can see on **Figure 10**.

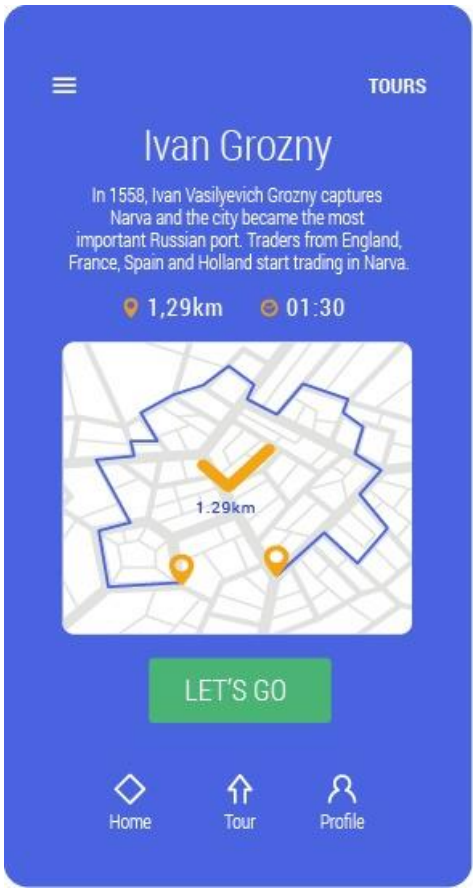

Figure 10 Map prototype.

2.2.3 Slider view

On early stage of slider author planned to put images which symbolize functions of application, name of slide, layout for dots that user can see on which slide he/she is. On first prototype author use slider as start screen when user login in application to show features which application can give. Slider screen can be found on **Figure 11**.

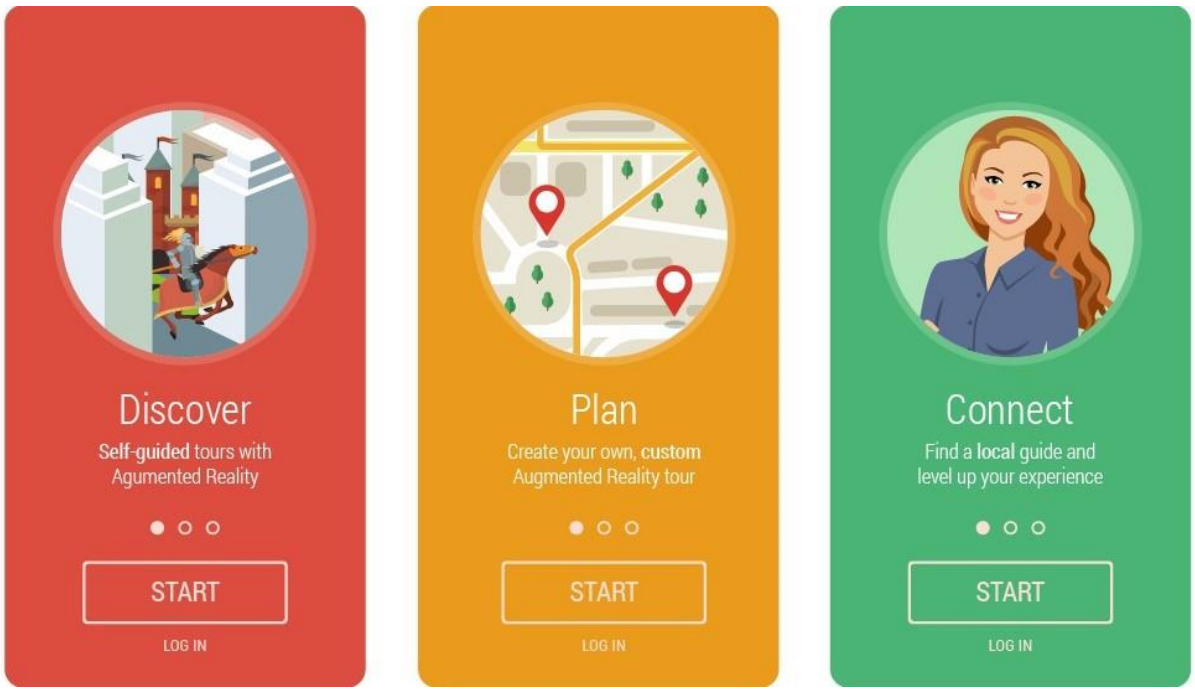

Figure 11 Slider prototype.

2.3 Application Screens and functions

Author's application uses different screens and functions. Some of them are complex and without proper documentation understand code is not simple. In this section author say about application work in details and describe functions which application use for proper work.

2.3.1 Main Screen

Application starts with class "MainActivity.java" which have functions for login into system by registered and anonymous users. Also have function which send user for registration screen and function for changing language for Russian, Estonian and English.

When application start it create layout or screen, here class create login screen. When class create screen, it also uses two custom functions which hide mobile keyboard and navigation bar. Next application check from user phone if permission for his location is granted for this application. It is needed for receiving user coordinates from his phone but all information about user location stays in application. If application is not granted access for user location, then application ask it from user. Next application gets button id from xml file where locate all design for screen. Xml file can be found in

app/src/main/res/layout, it has same name as class. Application set button on language change by “changelanguage” method. When user click on button it changes all text elements which is needed for translation and save language in application system, that means if application send user to next screen then it changes language automatically and if user restart application then on application start system take saved language from settings and set it as default language.

Next is functions which application needed for user login into system or registration it. If user already been on his account in application and close it then in next time user automatically login into application. If user need to login into system, then application take id of button which have function for user login and set conditions that if user have empty email or empty password field then application give user message that this or that field is empty. If all information was set into fields, then by clicking on login button application connect with Firebase authentication system if user already registered into system and user account is in Firebase authenticator and e-mail and password is correct then application send user to main screen where he/she can see tours which is now possible to go through. If something goes wrong, then user see information that “something went wrong please try again”.

For user registration screen have button “Registration”. If user click on button, then it activates method “openRegister” which send user for registration screen.

Also, there in class you can found old methods such us “openSlider”. This method was used in old version of application and author like to save them in application for further plans on application. Representation of screen can be found on **Figure 12**.

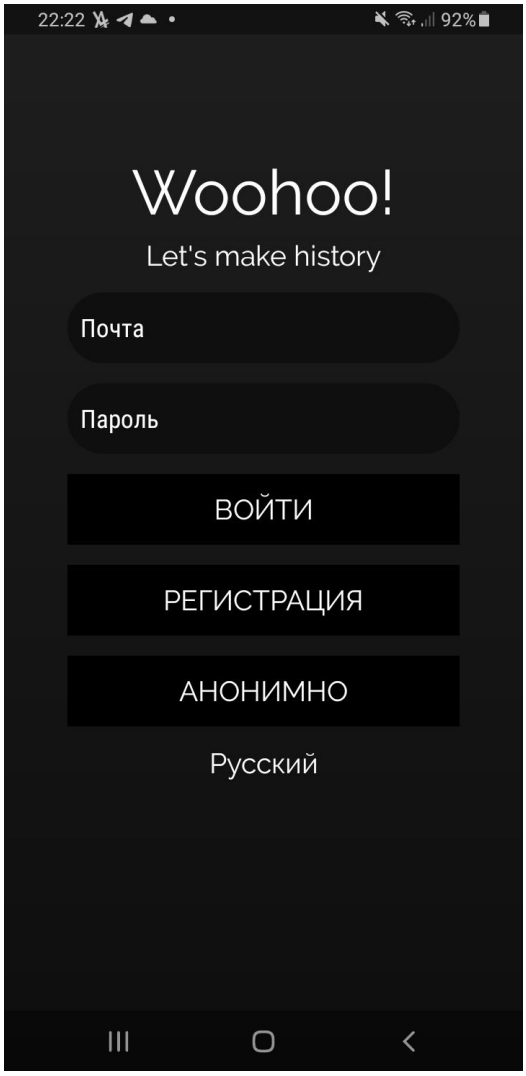

Figure 12 User login view.

2.3.2 Registration screen

Application registration screen is in Register class and all design for registration screen you can found in regisatrtion.xml.

Itself screen need for user registration into system. When user put all needed data for registration and click on registration button then application send user directly to main screen with tours and user data send for saving into Firebase Authentication system, in database application save nickname of user and user id.

When class start it check language which is set in system and change all text to preferred language. As in previous screen it also has method for hiding navigation bar and mobile keyboard. Next class take button id which is set in register.xml, then when user click on button application check that if all data which is needed written in textboxes application send all information in database by “addUser()” method and create account in Firebase Authentication system. If something goes wrong, then user receive message that “Error Occurred!”. If user like to go back on Login screen, then for that screen have “back” button which send user to previous screen. Representation of screen can be found on **Figure 13** Register view.

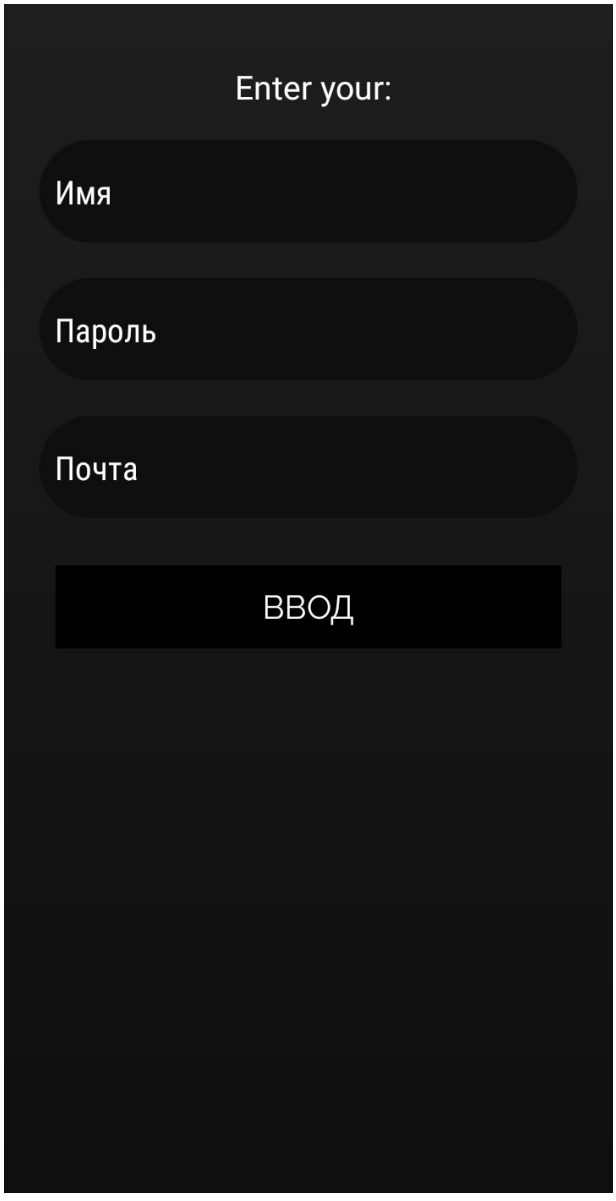

Figure 13 Register view.

2.3.3 Tours and Search

Tour screen is created as main screen when user login into application. Tour screen is needed to show user tours, navigate to different fragments by navigation about which I say later.

When user go from Registration screen or Login screen to Tours screen application open activity “nav.java” which base class for Tour screen where is written all code for correct work of screen. First class set layout for screen which have name “nav_layout.xml”. Also, when user go to Tour’s screen application load preferred by user language and translate text on this screen by method “languagecontrol”. Application takes from file system preferred language which was set by user and replace text to another language text. Then application set first fragment on screen. Main difference between fragment and activity which show screen of application that activity show screen, with which user can interact and fragment represent behavior of user interface so fragment can be collection of different elements of screen which can be changed by different fragment with different elements but fragment itself cannot show itself without activity which build screen.

Fragment class “Tours.java” need to show tours to user by parsing data about tour from firebase database and placing them into list which later can be used in two places: first is placing different tours on tours screen, second is placing them into list of tours where user put “like” which means that this tour is user favorite tours. When all data about tour was placed in list with override “onCreateView” method program change list to array list and next put list with Context in classes with names “ToursAdapter1.class” and “ToursAdapter2.class”. Context is interface which contain information about application. Context is needed for another class understanding in which class need to change elements. Now context have information about “Tours.java” class.

“ToursAdapter1.class” is needed to place all data from list into fragment where user can see all tour in application. About “ToursAdapter2.class” we talk later in Favorite screen part. In “ToursAdapter1.class” first adapter create view which contain information about layout where need to put information about tour and context which give program understanding where need to set this layout with already settled elements. Then start to work override method “onBindView()”, this method is needed to set information from list in layout. For setting picture from list author use Glide plugin. First author used Picasso plugin but because Picasso cannot scale picture to layout size and code to set picture was much bigger than Glide author choose to use this plugin. Application starts with taking

information about tour from database and if user is liked now parsed from database tour before then like on this tour change appearance if tour was not liked before then like not change its appearance. For like realization was choose another plugin SparkButton. Also, when user click on tour image or name application first declare tour name and send it to another class and change activity to “Gps.java” where user start tour. Also, if user click on like button, then like button change its appearance and send information about this to database, if user open application next time, then button appearance stays same next time. On class bottom is declared elements which class change. On this screen user can also see points which he/she receive by activities in tours and search tour by town name in which he/she is. When user write something in place where need to write town name then tours fragment change to search fragment which update each time when user write a new letter or do changes in town name. Representation of screen can be found on **Figure 14**.

Figure 14 Tour view.

2.3.4 Navigation panel

Navigation panel is needed for changing different fragments in application. In application planned different features and for that need different fragments with different information. For now, in application work fragment where user can see tours which is liked by user.

Navigation panel work by navigationView element in “nav.xml”, all elements which is set in menu is in nav_menu.xml, and all elements work by code which is written in “nav.java”. When user click on

specific item in navigation menu it changes fragment on screen but do not change screen itself it only changes interface. Also, when user is anonymous then username in navigation menu set to “anonymous”, when user is registered in system and login with account then it can see his username in tour. Representation of screen can be found on **Figure 15**.

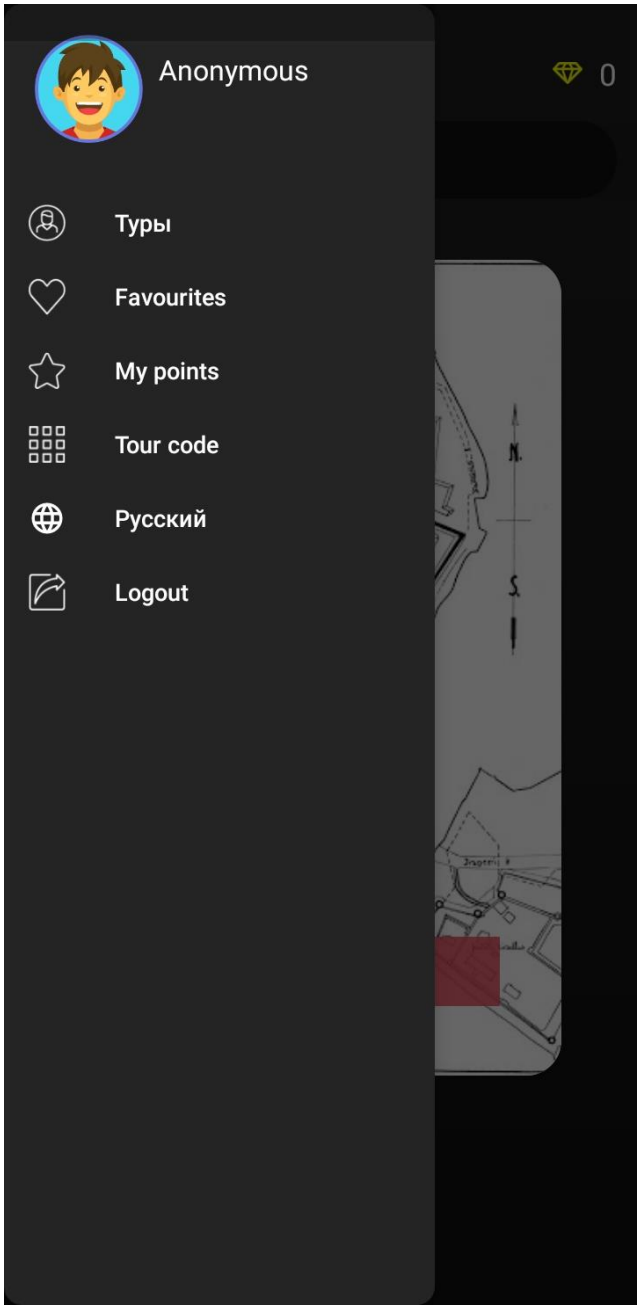

Figure 15 Navigation.

2.3.5 Favorite Screen

Favorite screen gives user information about which tour is liked and user can choose which tour user like to go through and start it. This screen shows only tours which user is liked by clicking on like button.

When user click on heart button first time it means that he like this tour and maybe this user like to go through this tour later or second time. When user click on that button application check in “TourAdapter1.java” or “TourAdapter2.java” class and if button state is false then tour title appends to user account in database and button change own appearance. If user like to see favorite tours, then in navigation view user can choose “Favorites” and fragment with tours change to favorite tours fragment. As author said before when user change screens by navigation view screen itself does not change only change fragment with information which application give to user. In favorite tours user can choose tour and go through it. In “Favourite.java” application take a tour title from user account and check if in this tour like status is true then it takes tour information from Tours table and set it base on same principal as fragment with tours. User can also delete tours from favorite, when user click on active like button then application change button status from true to false and write it to user database and remove it from fragment, fragment itself reload and change data. Representation of screen can be found on **Figure 16**.

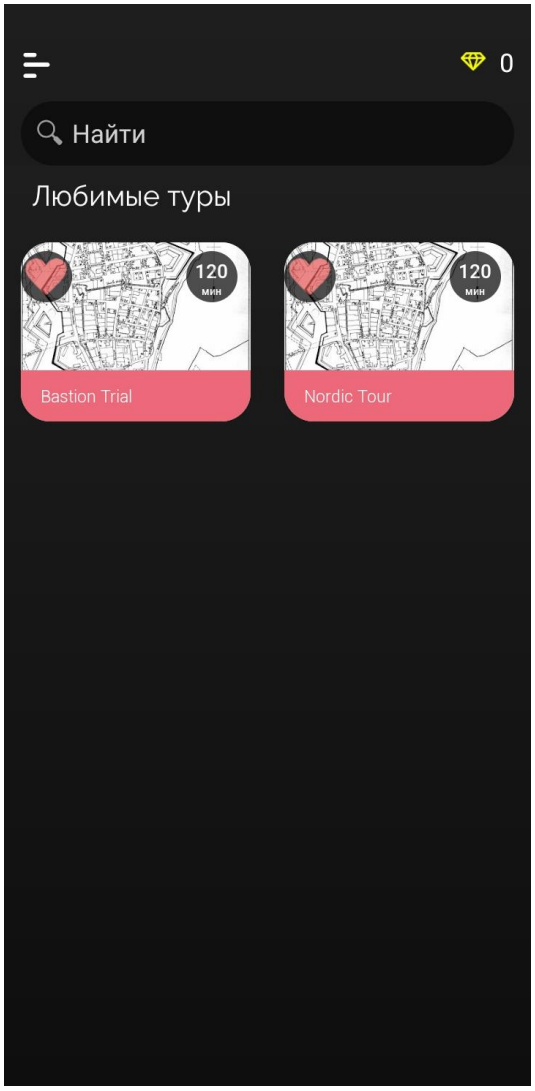

Figure 16 Favorite tours screen.

2.3.6 Map Screen

Map screen show user whole tour which user is chosen to go through: path between points, user location and path until first tour location. On bottom of screen user can found information about next point of tour and when user is near with location on map by clicking on point picture user can see information about tour, picture, text and audio which describe place or building. User also can answer questions which show him application when user is clicking on button. For that user receive points which later can be used by user in future. Also, if tour point has eye icon on it then user can click on that and see 3D model of building in AR.

When user open tour from favorite tour screen or tour screen, application open “Gps.java” class which create layout from “map.xml” file. App check if user is not anonymous if it is true then application connect with database and check if on user account already been on this tour and have saves from previous walking this tour then it receive parameters which there is, if there is not such parameters then application add this parameters to database and set them to default value and all information about tour points and questions set to arraylist and send it to “PointAdapter.java” class by constructor where program show user information about points of tour. Next application retrieves coordinates which is needed for creation of path between points and set all lines by addPolilinemethod() which is android studio method for creation of custom lines on map by coordinates. Next application set place for information which user receive about points by set adapter to place where this information show. Next application set points which user have in database and if user is anonymous then points are equal to 0 because anonymous user data do not save in database. Program have button which show user his/her location, if user click on that button then user receive location of his phone by method getMyLocation() which receive coordinates of user phone and change map center point to user current location, and button which user can go back to tour screen, when user click on that button application ask user if he like to go back to tour screen if user click second time then application send user screen with tour or another words application open “nav.java” class.

For right work of google map android application need override onMapReady() method which work directly with google map. Here application set all main coordinates on map. User location check onLocationChange () method which take current coordinates of user phone and coordinate of next point of tour and then calculate distance between two points. For reading user calculation user distance between points was written method which was not written with author and reference for that code can be found in references. Next application create line between user location and first point of tour, for that application take from database coordinate of first point and user location point and create automatic path between them.

When user came to point on map then “PointAdapter.java” class start to work by showing all information about point to user. PointAdapter set icons with tour point on map bottom and user can scroll them and click on them and receive different information. When user click on point of tour which is not next destination of user application show message and number of points which is user current location. When user came to tour point application control if user is 50 meters near with point and point of tour is current user destination, then it set all information about tour from list and set it in custom

layout. Program check three parameters: if in information about tour is AR component then in information about point application set button visible and clickable and when user click on that user can see 3D model of building in AR; if user reach final point of tour then application change button name to “Finnish” and when user click on that then it show user “thanks for going through this tour and send user to tour screen” , in normal condition button name is “Next” and when user click on that application set next point of tour as main point and close layout with information about previous point and if user is not anonymous user receive points for point and set last user point into database, if user already gone through this point then user not receive points; application check which language is now default and translate information about point (audio and text) to native language. If user have questions in this location, then program take information from list with questions and show that question which is linked to point by number. When user receive question, it has 2-4 answers for question, if user choose right question, then user receive points, if chosen answer was wrong then it has one more time to choose answer for question, if it same wrong then question close and user not receive points. Application check number by searching it in database in information about point of tour. When application set question and answers on custom created layout application check how many questions is in database and based on that application can set invisible and not clickable 3rd or 4th answers if database not found them. All screens can be found on **Figure 17**.

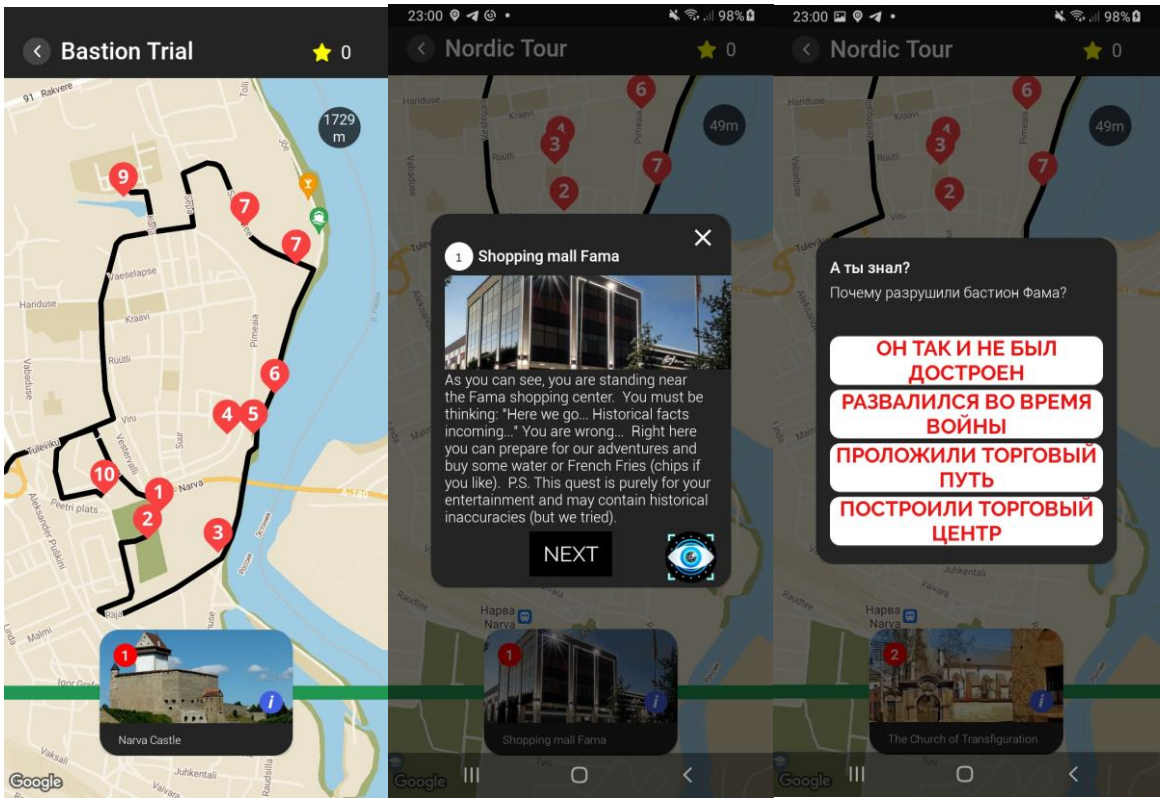

Figure 17 Map view.

2.5 AR screen

AR screen show user 3D model of building in augmented reality by ARKit technology which Google use for devices with Android OS. When user need to set 3D model in augmented reality user clicks on button “Download” and when downloading and rendering 3D model is ready user can set it in place where user like.

When user click on eye button application send user to “ARModue.class” which take current tour name and by that see which 3d model is linked to this tour. When application find link for that model it takes it from firebase storage where is all models and images store. For that application take a file name and file extension which describe object format. Base on that format application render that object and if object is completely rendered then user can place it by clicking on place where user like to see it. Model stays on place where user is set it and user can go around that model and see it from different point of view.

When user is finished to see that model, it can click on back button and it send user back in tour. Screen representation can see in **Figure 18Figure 18** AR screen.

Figure 18 AR screen.

2.4 Database architecture

Application consists of two parts, first is application created for phones on Android OS and second is Firebase platform where is database for information about tours and users and storage for media files.

Database is split to two parts. One is information about tours which contain different information about tour such as: name of tour, town for searching in Search class, color for header where name is hold in Tour class, likes from users, how much need to go through tour, question indicator which indicate question number, questions which have tour, text and audio translation on different languages, link for image and 3D model in firebase storage. MainCoordinates table hold information about main points of tour. SubCoordinates table hold information about points which help to create path between main points on map. Realization of database can be found on **Figure 19**.

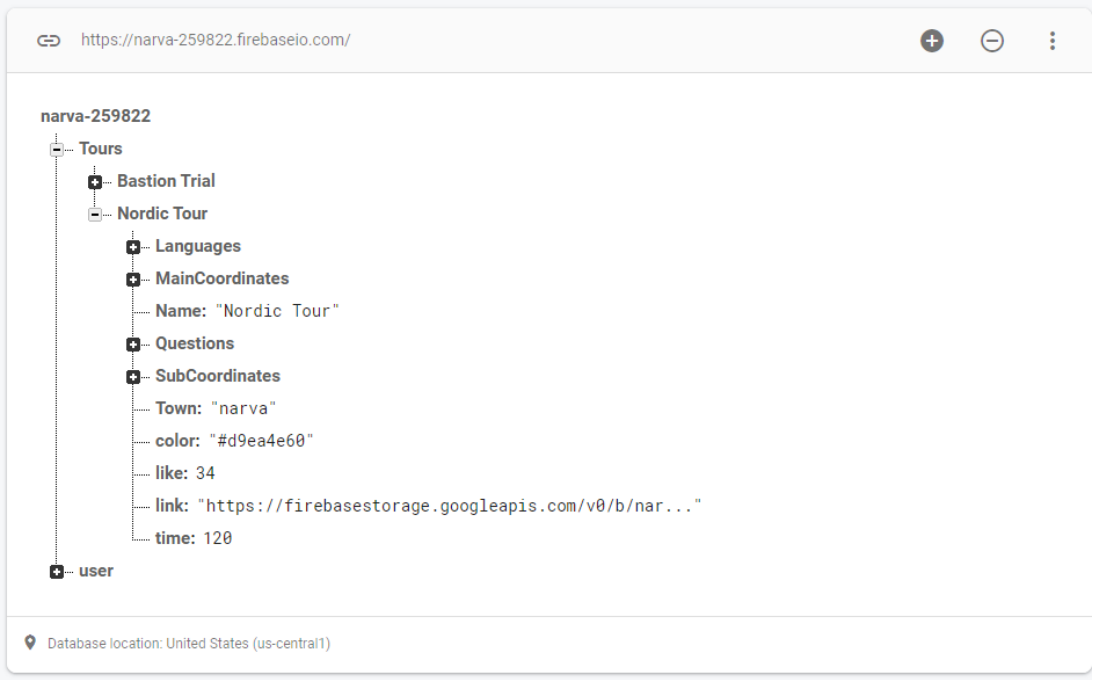

Figure 19 Database for tours.

Second part of database is information about user. When user create account, application give to that account random id. Application hold information about user: username and points which user receive. User e-mail and password hold in authentication program, this is database for user data to login into system. This data is crypted by algorithm which Google is created and information about account is not possible to see. Also, it holds information about which tours user went through or stay on the middle of tour or started it but not go through all points, also tour have information about if user like that tour or not. Realization of database can be found on **Figure 20** Database for users.

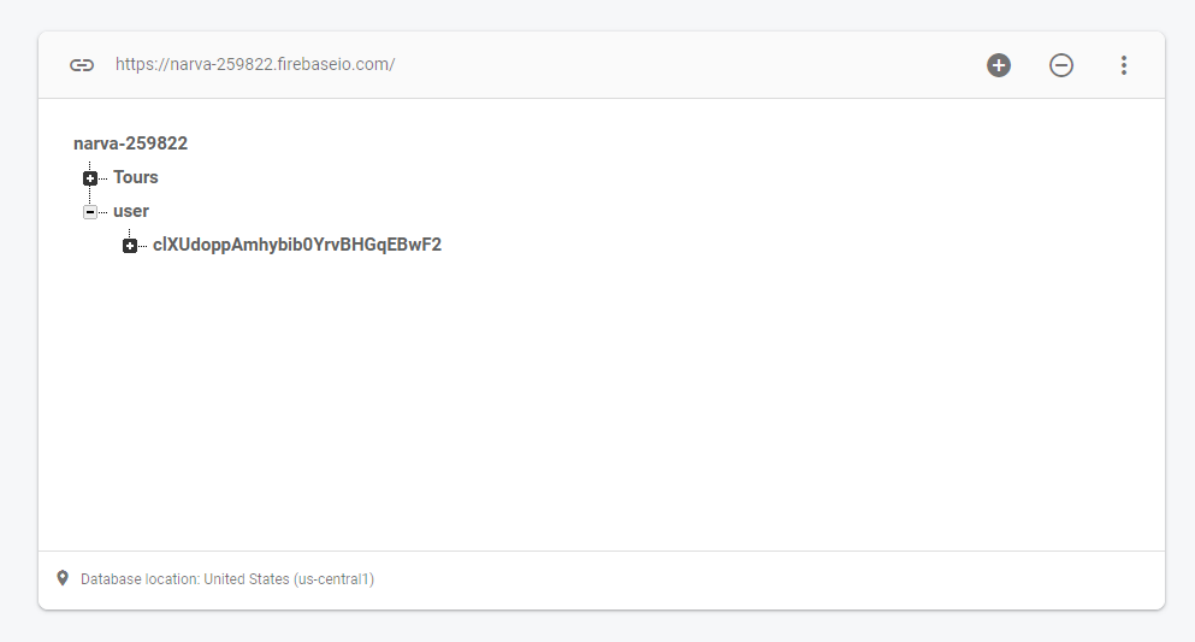

Figure 20 Database for users.

2.5 Quest creation

Tours in application was created by Alex Bliskov. Process of tour creation start with location choosing and collection materials about different historical buildings and places. Good example of high-quality tour can be Nordic Tour which was created with cooperation with ARN team and Nordic Council. When materials about different important history places is collected team start to choose which places can be linked by story in tour and points with different places can be connected by path on map. Story in tour need to be interesting for user and path between different points need to be not long from one point to another. For creation tour map prototype was chosen web application Google My Maps. This web application gives user instruments for creating own map and importing it in kml or kmz file formats which later user can use on phone and this custom map can be places on Google maps on user phone. My maps application was used for testing and development of map. When all main points, questions and path between points was set on My Maps application map ARN team go through that tour and search strong and weak points. When tour story and map weak points was changed then all information about as given to author and all this information was paced in database. Author used My Maps application kml file to extract from it coordinates of main points, questions, and coordinates for creating path between points in tour.

2.6 Creation of 3D Models

3D models were created by Yuri. 3D models were created Blender programs. Blende is application which help to create different animations and work with 3D objects. Blender was used to design and create buildings.

2.7 Source code

Application files can be found by GitHub link: https://github.com/santeri13/Diploma_work

Application supports all Android phones from 24 SDK version.

3.Future plans on application

Application has base functions which need for test. App can grow in future when would be tested on user and receive reviews, but author already have plans for future for application.

In future author plan to integrate achievements which user can receive for different activity in application. Next idea is to create web application which give user options to create own tours and send them for moderation and for next implementation them into application. Author already made mockup for that idea but first like to test application on user to understand if this idea is good or not. Next idea for now is creating artifact or another words collectables items which user can collect and receive for that points or thing which linked with next idea, this collectable item can help to hold users in tours and give them feeling of not completed adventure. Final idea is to somehow work with local business and send user for them. As example when user receive some prize in tour it can receive coupon in café, or this business can be integrated in tour as shops which sell souvenirs.

Conclusion

As a result of the thesis, an Android application has base functions. In application was implemented login and register screen. To login, the user needs to write his/her e-mail and password correctly if the user writes wrong e-mail or password then he/she receive a message about the error. Register users can choose tours and go through them, receive points for going through tours, answer questions and receive new knowledge by different possibilities. There is the possibility of anonymous login for users who do not want to spend time on registration. Anonymous users have the same rights as registered ones, to see tours, and go through them but cannot receive points for them and cannot save them into account.

All the basic functions needed in this application are present and ready to go. The application works good but has not been tested on other people, except the author and his computer, as well as the phone.

Resümee

Lõputöö tulemusel on Androidi rakendusel baasfunktsioonid, kuid rakenduse täiendamiseks tuleb töötada täiendavate funktsioonidega. Rakenduses rakendati sisselogimise ja registreerimise ekraan. Sisselogimiseks peab kasutaja õigesti kirjutama oma e-posti ja parooli, kui kasutaja kirjutab vale e-kirja või parooli, siis ta saab teade vea kohta. Registreeritud kasutajad saavad valida ekskursioonid ja neid läbida ning näha erinevaid 3D-mudeleid koos nende kirjeldusega. Kasutajatel, kes ei soovi registreerimiseks aega kulutada, on anonüümne sisselogimise võimalus. Anonüümsetel kasutajatel on ekskursioonide vaatamiseks ja nende läbimiseks samad õigused kui registreeritud kasutajatele aga anonüümne kasutaja ei saa avada . jaotise lemmikute tuuridega ja ei saa punktid tuuride läbikäimisel.

Kõik selles rakenduses vajalikud põhifunktsioonid on olemas ja kasutamiseks valmis. Rakendus töötab stabiilselt, kuid seda pole teiste inimeste peal testitud, välja arvatud autor ja tema arvuti, samuti telefon. Selle lõputöö lõpus kirjeldab autor edaspidiseid töid, mis tuleb ära teha rakenduse rakendamiseks tootmises.

REFERENCES

Wikipedia. Android (Operating System). Available at https://en.wikipedia.org/wiki/Android_(operating_system), accessed April 15 2020.

History of Java programming language. Available at https://www.freejavaguide.com/history.html ,accessed April 15 2020.

Unity (Game Engine), Available at https://en.wikipedia.org/wiki/Unity_(game_engine), accessed April 15 2020.

Extensible Markup Language (XML) 1.0, Available at https://www.xml.com/axml/axml.html, accessed April 16 2020.

Gradle, Available at https://ru.wikipedia.org/wiki/Gradle, accessed April 18 2020.

Gradle User Manual, Available at https://docs.gradle.org/current/userguide/userguide.html, accessed April 18 2020.

Android Studio, Available at https://en.wikipedia.org/wiki/Android_Studio, accessed April 18 2020.

View, Available at https://developer.android.com/reference/android/view/View, accessed April 21 2020.

Activity, Available at https://developer.android.com/reference/android/app/Activity, accessed April 21 2020.

Firebase, Available at https://en.wikipedia.org/wiki/Firebase, accessed April 22 2020.

Firebase Realtime Database, Available at https://firebase.google.com/docs/database, accessed April 15 2020.

Questcity, Available at https://questcity.ee/en/, accessed May 18, 2020.

Lokimo, Available at https://lokimo.ee/, accessed May 18, 2020.

Code for creating route until point, Available at https://github.com/Vysh01/android-maps-directions/tree/master/app/src/main/java/com/thecodecity/mapsdirection/directionhelpers, accessed March 20 2020.

Tutorial with navigation bar, Available at https://www.youtube.com/playlist?list=PLrnPJCHvNZuD-Q-jWPw13-wY2J57Z6ep6G, accessed May 19 2020.

Firebase query tutorial, Available at https://www.youtube.com/watch?v=WeoryL3XyA4, accessed May 19 2020.

Code for Slider creation, Available at https://www.youtube.com/watch?v=byLKoPgB7yA, accessed February 11 2020.

Data, Available at https://en.wikipedia.org/wiki/Data , accessed April 20 2021.

Object oriented programming, Available at https://en.wikipedia.org/wiki/Object-oriented_programming , accessed April 20 2021.

Application source code, Available at https://github.com/santeri13/Diploma_work