UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Juhan-Rasmus Risti

Android application for organising meetings

Bachelor's Thesis (9 ECTP)

Supervisors: Satish Narayana Srirama
Liyanage Don Mohan Devapriya Liyanage

Tartu 2016

Android application for organising meetings

Abstract:

This bachelor’s thesis describes the creation of an Android application which is used to create and organise meetings in a fast and easy manner.

The resulting application gives a full overview of a user’s meetings, lets effortlessly create new meetings and to invite contacts to join. The main uniqueness of the application lies in the feature that the user has multiple options when choosing a location for a meeting. For example, it is possible to get automatic recommendations from the application based on the positions of the participants. In addition, the application enables users to track the positions of other participants in real-time, if the participants choose to share it.

The application is also compared against other similar and popular applications. It is found that the other applications do not offer similar location recommendation or participant position tracking features. On the other hand, the other applications have a wide user base and some functionalities the new application does not offer yet.

Keywords:

Android, meetings organiser, automatic location recommendation, real-time GPS tracking

CERCS:

P175 (Informatics, systems theory)

Androidi rakendus kohtumiste korraldamiseks

Lühikokkuvõte:

Käesolevas bakalaureusetöös kirjeldatakse Androidi rakenduse loomist, millega saab kiiresti ja lihtsalt teha ning planeerida kohtumisi.

Valminud rakendus annab täieliku ülevaate kasutaja kõikidest kohtumistest, lubab ilma vaevata luua uusi kohtumisi ning kutsuda osalema enda kontakte. Rakenduse põhiline unikaalsus seisneb võimaluses omada mitut valikut kohtumise asukoha valimisel. Näiteks on võimalik rakenduselt asukoha jaoks saada automaatse soovituse, mis arvutatakse kõikide osalejate positsioonidest. Lisaks võimaldab rakendus vaadata teiste osalejate antud hetke asukohti, kui nad on nõus seda jagama.

Uut rakendust võrreldakse ka teiste sarnaste ja tuntud rakendustega. Võrdluse tulemusel leitakse, et teised rakendused ei paku samalaadset asukoha soovitamist ega osalejate positsioonide jälgimise võimalust. Samas on teistel rakendustel eeliseks suur kasutajabaas ning mõned funktsionaalsused, mida loodud rakendus veel ei paku.

Võtmesõnad:

Android, kohtumiste korraldaja, automaatne asukoha soovitamine, reaalajas GPS jälitamine

CERCS:

P175 (Informaatika, süsteemiteooria)

Table of Contents

1 Introduction ... 5
1.1 The problem ... 5
1.2 The solution ... 5
1.3 Outline ... 5
2 State of the art ... 6
2.1 The client ... 6
2.1.1 Android ... 6
2.1.2 Android Studio ... 6
2.1.3 Gradle ... 6
2.1.4 Retrofit ... 6
2.1.5 Google Maps Android API ... 7
2.1.6 Google Places API Web Service ... 7
2.1.7 Rey5137's Material Design library ... 7
2.2 The server ... 8
2.2.1 Heroku ... 8
2.2.2 Spring Framework ... 8
2.2.3 Hibernate ... 8
2.3 Previous and similar work ... 9
2.3.1 Facebook Events ... 9
2.3.2 Google+ Events ... 10
3 The result ... 12
3.1 The application ... 12
3.1.1 Prerequisites ... 12
3.1.2 Login and registration ... 12
3.1.3 The meetings list ... 13
3.1.4 Creating a meeting ... 14
3.1.5 The meeting info view ... 17
3.1.6 The location recommendation algorithm ... 20
3.1.7 The participant tracker algorithm ... 21
4 Usability testing ... 22
4.1 Results ... 22
5 Future development ... 24
5.1 More efficient algorithm for recommending a meeting location ... 24
5.2 Push notifications ... 24

5.3 Allow a meeting leader to change an existing meeting ... 24
5.4 Allow users to change details about their account ... 24
5.5 An account-related in-app contacts list ... 24
5.6 E-mail and phone number authentication ... 24
5.7 Possibility to log in with a Google or social media account. ... 24
5.8 An in-app messaging or commenting system ... 25
5.9 Improve the markers in map layout ... 25
6 Summary ... 26
7 References ... 27
8 Appendix ... 30
8.1 The source code ... 30
8.2 Installation instructions ... 30
8.3 Database model ... 30
8.4 License ... 31

1 Introduction

1.1 The problem

In several ways, today's technology has changed the way humans communicate. Physical communication has been largely substituted by non-physical communication methods, e.g. instant messaging, voice calls and video calls. Nevertheless, in certain situations physical communication is still irreplaceable as it often helps humans to better connect with others and to satisfy their social needs. However, it is still possible to use modern technology to make physical communication also more comfortable.

One approach to make physical communication better would be to make it easier and more comfortable to organise and plan meetings. Without the help of technology planning meetings can be a big hassle and because of this people might prefer non-physical communication methods instead.

Although a few current solutions which tackle this problem are relatively good and largely used, they still lack some key features, e.g. being fast to use and helping to decide the location for a meeting.

1.2 The solution

The goal of this thesis is to create an application which would allow users to

- create both quick and normal meetings in a simple manner without sacrificing functionality;
- get recommended places for meetings when the user is not certain which location to choose;
- easily track and contact other participants to see how far they are;
- get a detailed yet simple overview of all the user's meetings;

The application aims to be easy to use and to decrease the effort from the user's side by automatically setting the starting time and duration of a quick meeting, recommending the best places for the user to start a meeting and giving an overview of the other participants' positions.

1.3 Outline

Section 2 describes the approach to solving the problem, more precisely describing the technologies and methods that are used on both the client and server side. It also describes some similar and well-knows applications, their pros and cons, and how they compare to the author's solution.

Section 3 describes the final result i.e. the application.

Section 4 describes the usability test results.

Section 5 discusses the future development options for the application.

Section 6 summarizes the work in English.

Section 7 contains all the references.

Section 8 contains the appendix.

2 State of the art

2.1 The client

2.1.1 Android

Android[1] is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets [1].

The main mobile operating system alternatives to Android would be Apple's iOS and Windows Phone (WP). Compared to iOS and WP, Android has a hugely larger global market share [2]. Because of this and the fact the author was more familiar with Android programming, he chose Android as the target operating system for this application.

2.1.2 Android Studio

Android Studio[2] is the official IDE[3] for Android application development, based on IntelliJ IDEA [3].

Compared to its main alternative, the Android Development Tools (ADT) plugin for Eclipse IDE, Android Studio is actively being developed by Google, whereas the ADT plugin has been discontinued [4]. This was also the main reason why the author chose Android Studio.

2.1.3 Gradle

Gradle[4] is a project automation tool that builds upon the concepts of Apache Ant and Apache Maven and introduces a Groovy-based domain-specific language (DSL) instead of the more traditional XML form of declaring the project configuration [5]. Thanks to Google's support, Android Studio offers native support for Gradle.

The main alternative for Gradle is Maven[5], a build-automation tool developed by Apache Software Foundation. Compared to Maven, Gradle has a shorter learning curve [6], can be up to two times faster [7], is easier to maintain [8] and better documented [9]. Because of those, Gradle is also the default build system for Android Studio. On the flip side, Maven has a lot more existing plugins [10] and it integrates much better with application servers and continuous integration servers [11].

The author chose Gradle because of its native support in Android Studio and its ease of use.

2.1.4 Retrofit

Retrofit[6] is a type-safe HTTP[7] client for Android and Java.

Compared to its main alternative, Google's Volley[8], which is also a HTTP library, Retrofit is easier to use as it lets the user treat API[9] calls as simple Java method calls, automatically

[1] https://www.android.com/
[2] https://developer.android.com/studio/index.html
[3] Integrated Development Environment
[4] https://gradle.org/
[5] https://maven.apache.org/
[6] http://square.github.io/retrofit/
[7] Hypertext Transfer Protocol
[8] https://developer.android.com/training/volley/index.html
[9] Application Programming Interface

supports a greater variety of response types, including converting JSON[10] objects into Plain Old Java Objects, it has a much greater documentation and is more lightweight. On the other hand, Volley has basic support for image loading, better caching system and request prioritization. [12]

The author chose Retrofit because it was easier to use and he did not need the extra features of Volley for this project.

2.1.5 Google Maps Android API

The Google Maps Android API[11] is a service which is part of the Google Play services library. The API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures and it allows to add markers, polygons, and overlays to a basic map, and to change the user's view of a particular map area [13].

There are also some lesser-known alternatives. For example, Mapbox[12] offers better map customizability with different styles for the free version. However, Mapbox maps can be less accurate (especially in rural areas in developed countries), it has a limit on the usage in free version (50 000 requests per day vs Google's unlimited usage) and it doesn't include other useful services such as Google's Places or Streetview [14] [15].

The author chose Google Maps because of its familiarity, unlimited usage and better accuracy.

2.1.6 Google Places API Web Service

The Google Places API Web Service[13] is a service which returns information about places (establishments, geographic locations, and prominent points of interest) using HTTP requests [16].

The main alternative is Foursquare Venues. The usage limits are quite similar for the free version, as Google Places permits 150 000 queries per day [17] and Foursquare Venues permits 120 000 queries per day [18]. While there are not any notable differences between the two services, the author preferred Google Places API because he had previous experience using it.

2.1.7 Rey5137's Material Design library

Rey5137's Material Design[14] library is an open source Android library which enables applications to use Material Design components on pre-Lollipop Android devices [19]. Thanks to the library the user interface will be uniform on both older and newer devices and it decreases the need to make separate layouts for different Android versions.

The main alternative would be navasmdc's Material Design[15] library. Unlike rey5137's library, navasmdc's library is not actively developed anymore and therefore the author chose the former.

[10] JavaScript Object Notation

[11] https://developers.google.com/maps/documentation/android-api/

[12] https://www.mapbox.com/

[13] https://developers.google.com/places/web-service/

[14] https://github.com/rey5137/material

[15] https://github.com/navasmdc/MaterialDesignLibrary

2.2 The server

2.2.1 Heroku

Heroku[16] is a cloud platform as a service (PaaS) supporting several programming languages [20]. Its free service offers up to 1000 dyno[17] hours per month per account for verified accounts [21] and limits the database to 10 000 rows [22]. It also offers good integration with Git.

One of the main alternatives to Heroku is Google App Engine[18], a platform for building scalable web applications and mobile backends. App Engine provides developers with built-in services and APIs such as NoSQL datastores[19], memcache, and a user authentication API. It offers up to 28 instance hours per day per account, limits the database storage to 1 GB and limits read/write operations. It also forces users to use Cloud Datastore instead of traditional relational databases. This increases its learning curve and decreases the portability of the database. [23] [24]

Another alternative is OpenShift[20], a PaaS provided by Red Hat. The free version limits the database storage to 1 GB per application and three applications per account [25]. Otherwise there are no major differences between OpenShift and Heroku, as both support relational databases.

Ultimately the author chose Heroku over the other alternatives as Google App Engine had too many restrictions and he was more familiar with Heroku than Openshift.

2.2.2 Spring Framework

The Spring Framework[21] provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform. A key element of Spring is infrastructural support at the application level: Spring focuses on the "plumbing" of enterprise applications so that teams can focus on application-level business logic, without unnecessary ties to specific deployment environments. It features dependency injection, Spring MVC[22] web application, RESTful[23] web service, etc. [26]

The author chose Spring Framework because he had previous experience using it.

2.2.3 Hibernate

Hibernate[24] is an object-relational mapping framework for Java. It provides a framework for mapping an object-oriented domain model to a relational database. Hibernate solves object-relational impedance mismatch problems by replacing direct, persistent database accesses with high-level object handling functions. [27]

The author chose Hibernate because he had previous experience using it.

[16] https://dashboard.heroku.com/

[17] A lightweight Linux container that runs a single user-specified command.

[18] https://appengine.google.com

[19] A highly-scalable non-relational database.

[20] https://www.openshift.com/

[21] https://projects.spring.io/spring-framework/

[22] Model–view–controller

[23] Representational State Transfer

[24] http://hibernate.org/

2.3 Previous and similar work

There are several applications which offer the functionality to easily create and organise physical meetings. This section describes a few free applications, their pros and cons, and how they compare to the solution proposed in this thesis. The comparisons were made by testing them side-by-side. The applications described here will be Android applications. The included screenshots were taken with a LG G2, Android version 4.4.2.

2.3.1 Facebook Events

Facebook Events[25] is not a standalone application but rather a feature of Facebook. It offers similar functionalities to the author's solution, e.g. being easy and fast to use. However, its location recommendation system is lacking and it doesn't allow to track other participants.

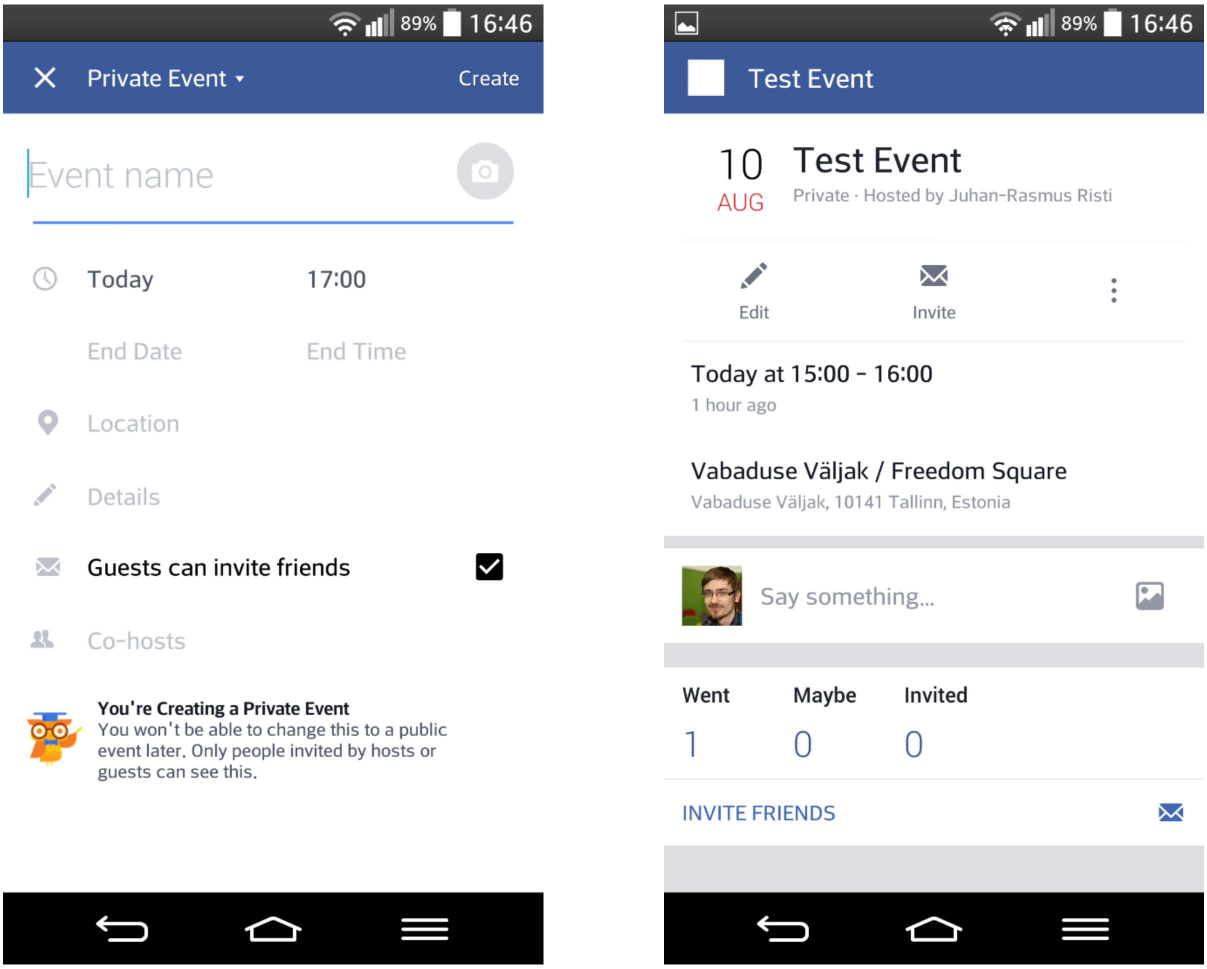
Pros:

- Easy to use
- Vast user base
- Allows to invite Facebook contacts
- Several useful functionalities, including allowing users to distinguish private and public events, to add co-hosts to events and to add comments to events

Cons:

- Not a standalone application; can be bothersome to use because you have to use it through Facebook's application
- Location recommendation system consists only of recommending nearby places to the user
- Doesn't allow to track the locations of other participants

[25] https://events.fb.com/



89% 16:46

Private Event ▾ Create

Event name

Today 17:00

End Date End Time

Location

Details

Guests can invite friends ☑

Co-hosts

You're Creating a Private Event
You won't be able to change this to a public event later. Only people invited by hosts or guests can see this.

89% 16:46

Test Event

10 AUG **Test Event**
Private · Hosted by Juhan-Rasmus Risti

Edit Invite

Today at 15:00 - 16:00
1 hour ago

Vabaduse Väljak / Freedom Square
Vabaduse Väljak, 10141 Tallinn, Estonia

Say something...

Went	Maybe	Invited
1	0	0

INVITE FRIENDS

Figure 1. Facebook Event creation view on the left
and the main event detailed view on the right.

2.3.2 Google+ Events

Google+ Events[26] is not as popular as Facebook Events, but it also offers a very similar functionality. However, it doesn't have any location recommendation system and it also doesn't allow to track other participants.

Pros:

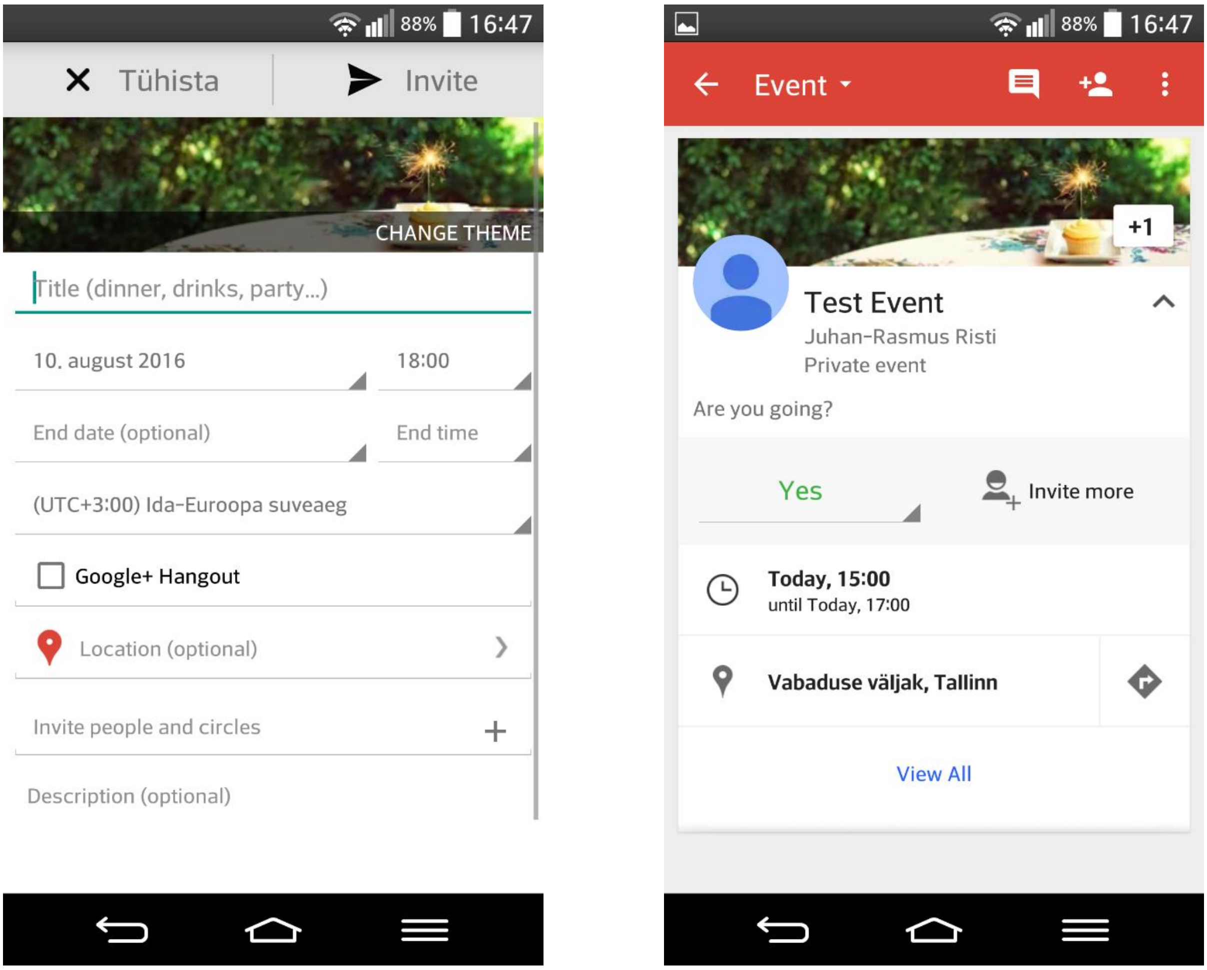
- Easy to use
- Quite large user base
- Allows to invite Google+ users
- Several useful functionalities, including allowing users to distinguish private and public events, and to add comments to events

Cons:

- Not a standalone application; can be bothersome to use because you have to use it through the Google+ application
- Features no location recommendation
- Application is unstable; crashed once when testing it

[26] https://plus.google.com/

- Doesn’t allow to track the locations of other participants



Title (dinner, drinks, party...)

10. august 2016 18:00

End date (optional) End time

(UTC+3:00) Ida-Euroopa suveaeg

□ Google+ Hangout

Location (optional)

Invite people and circles

Description (optional)

Test Event
Juhan-Rasmus Risti
Private event

Are you going?

Yes

Invite more

Today, 15:00
until Today, 17:00

Vabaduse väljak, Tallinn

View All

Figure 2. Google+ Event creation view on the left
and the main event detailed view on the right.

3 The result

3.1 The application

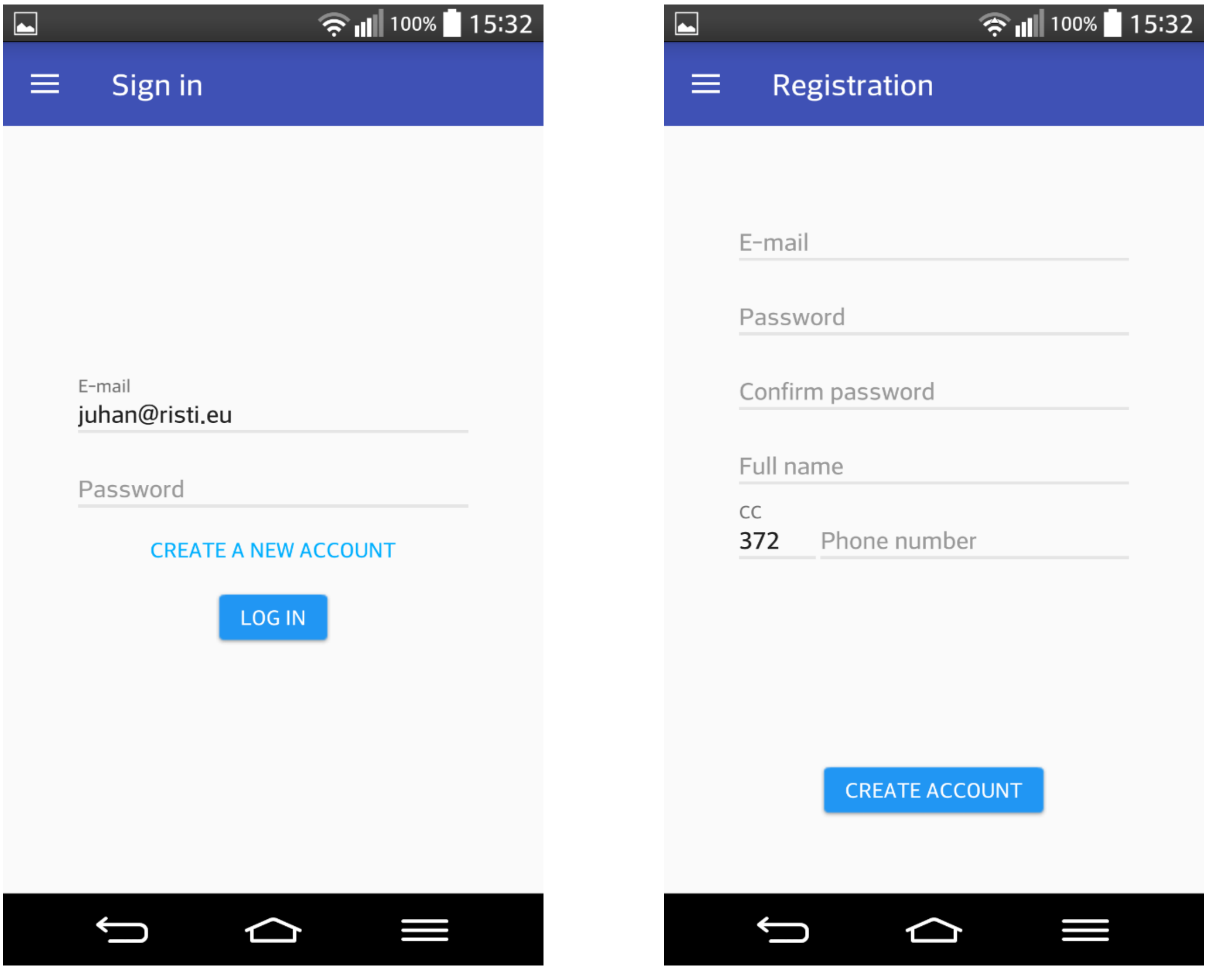
This section describes the views and all related functionalities of the developed mobile application. The paragraphs also describe the application's flow and are ordered according to it, starting from the very first procedures i.e. signing in and registering.

3.1.1 Prerequisites

The application requires an Android phone or tablet with Android version 4.0 (Ice Cream Sandwich) or higher. The download link for the application's apk file is included in the appendix section.

3.1.2 Login and registration

When first starting the application, the user is shown the login screen (Figure 3). They can log in using their e-mail and password or continue to the registration view by clicking on the "Create a new account" button. The application doesn't presently allow logging in using a social media account.



E-mail
juhan@risti.eu

CC
372

Figure 3. Login view on the left and registration view on the right.

In the registration view (Figure 3) the user must fill the following information:

1. E-mail
2. Password and password confirmation
3. Full name
4. Phone number with his country code

It is important that the phone number and e-mail are correct as those values are used to help other people send them invitations. When the user has inserted all required information, they can click on the “Create account” button to continue.

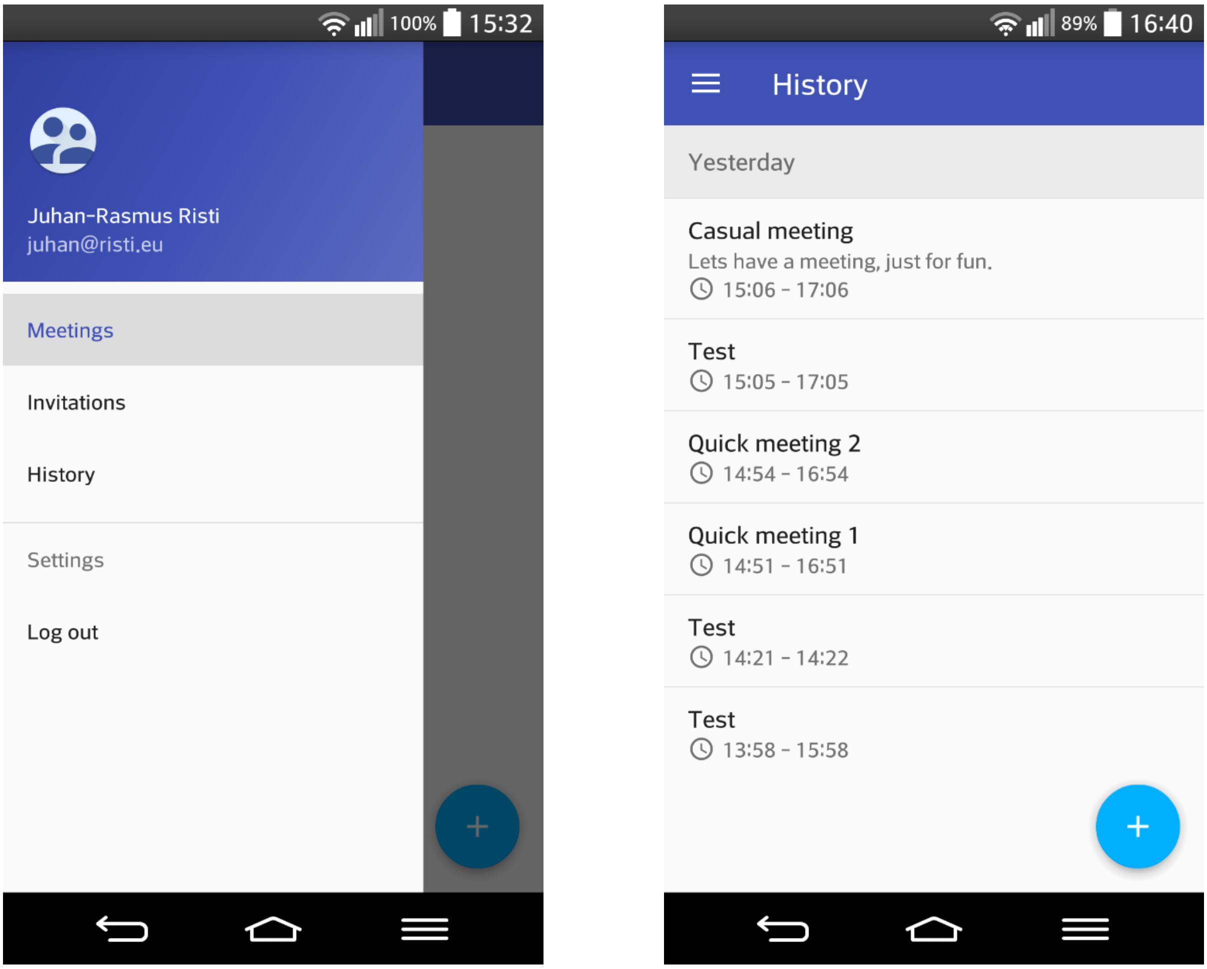
3.1.3 The meetings list

After logging in or registering successfully, the user is directed to the meetings list view (Figure 4). There they can open the drawer menu and see basic information about their account (name and e-mail). The user has four options to choose from:

1. “Meetings” – shows the user a list of currently ongoing and upcoming meetings
2. “Invitations” – shows the user a list of meetings to which they have been invited but have not given an answer yet
3. “History” – shows a list of all meetings where the user participated and which have already ended
4. “Log out” – logs the user out from his account

The meetings in each list are grouped by their starting date and sorted by their starting time. Each list item shows an overview of the related meeting: its title, description and duration time. The user is brought to the meeting’s detailed view when choosing it from the list. Additionally, the user can click on the floating action button[27] (FAB) with the “+” sign, which is situated in the bottom right corner of the screen. By doing so, they are presented with the meeting creation view.

[27] A specialised action button which ‘floats’ on top of the UI in a circular shape.



Yesterday

Casual meeting
Lets have a meeting, just for fun.
15:06 - 17:06

Test
15:05 - 17:05

Quick meeting 2
14:54 - 16:54

Quick meeting 1
14:51 - 16:51

Test
14:21 - 14:22

Test
13:58 - 15:58

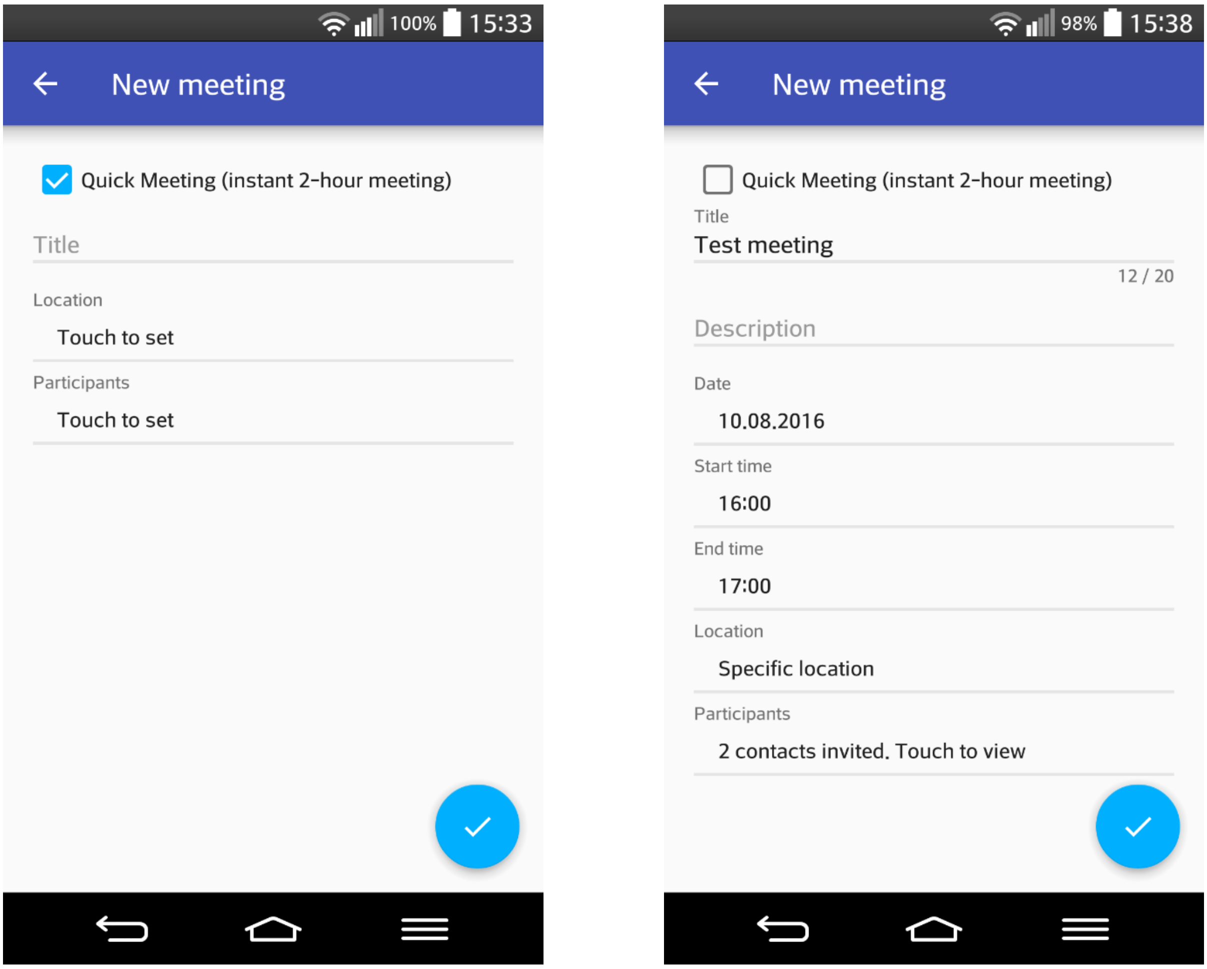
Figure 4. Drawer menu on the left
and a non-empty meetings list on the right.

3.1.4 Creating a meeting

The “New meeting” view consists of several components (Figure 5). The first component is a checkbox to toggle the Quick Meeting functionality on and off. The other components are field to set the title, description, date, starting time, ending time, location and participants of the meeting.

3.1.4.1 Quick Meeting

A Quick Meeting is a special meeting type. Choosing it creates a descriptionless 2-hour meeting which starts instantly after creating it. With Quick Meeting the user only has to set the title, location and invite participants. In addition to that, Quick Meeting unlocks the location recommendation system. The recommendation system is not available with a normal meeting because it recommends a location to the user based on the current positions of the participants and therefore it is useful only when the meeting starts as soon as possible. If the meeting would start at a later time, the recommended location would not be valid anymore because most likely the positions of the participants have changed by that time.



100% 15:33

New meeting

☑ Quick Meeting (instant 2-hour meeting)

Title

Location

Touch to set

Participants

Touch to set

98% 15:38

New meeting

□ Quick Meeting (instant 2-hour meeting)

Title

Test meeting

12 / 20

Description

Date

10.08.2016

Start time

16:00

End time

17:00

Location

Specific location

Participants

2 contacts invited. Touch to view

Figure 5. A quick meeting form on the left
and a filled normal meeting form on the right.

3.1.4.2 Choosing the location

After tapping on the "Location" field, the user is sent to a map view (Figure 6). The view consists of:

1. the map itself;
2. a search bar to search a location;
3. several FABs to help the user interact with the map;
4. a location choice spinner[28];
5. a text view which describes the chosen option;

The user must choose one of the three available location options. As mentioned before, the last two choices are available only with a Quick Meeting.

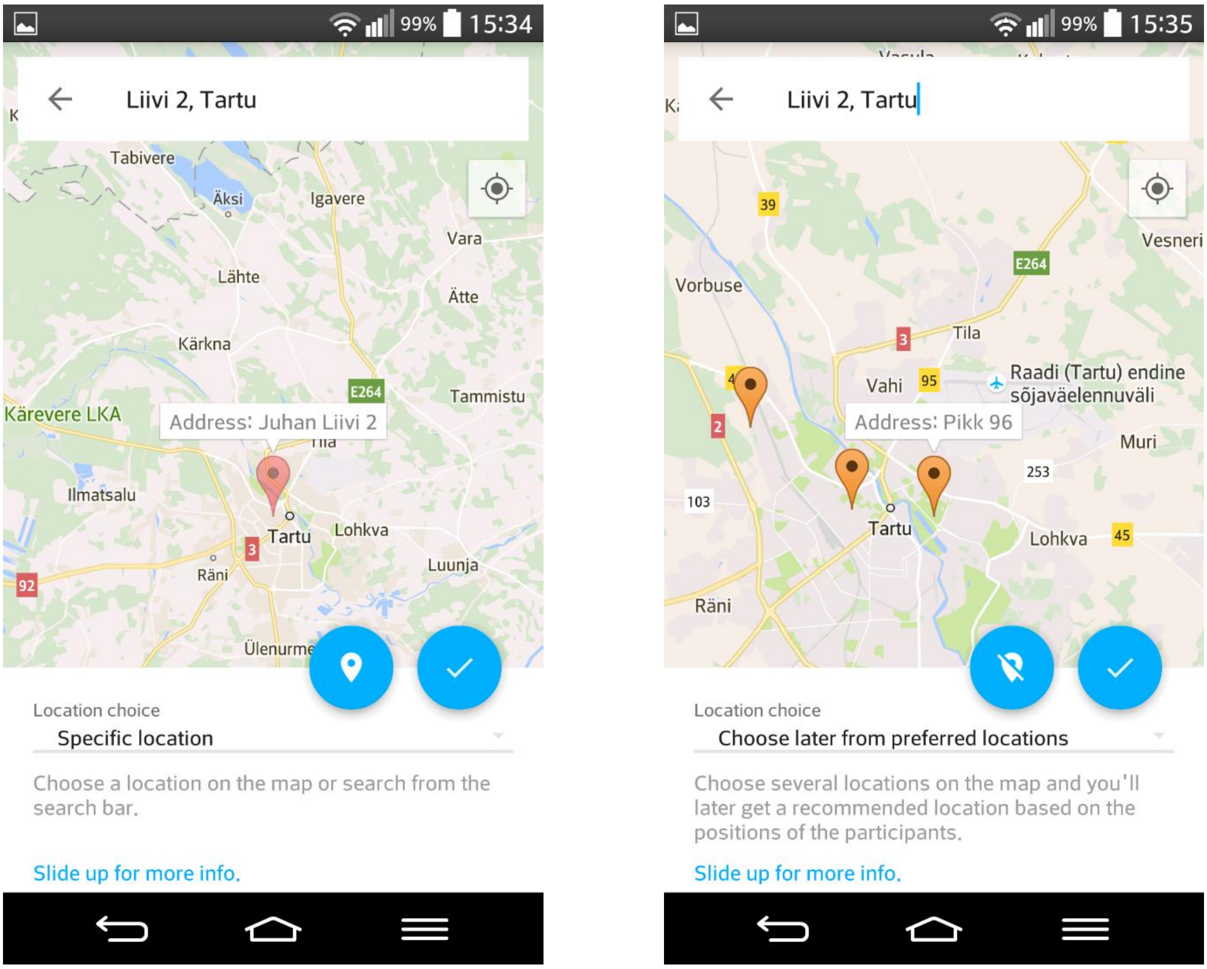
The first location option is "Specific location". With it the user must pinpoint the desired location on the map or search for it using the search bar. They can place a temporary marker freely anywhere and after placing the temporary marker on the suitable location, the user must confirm the location by clicking on FAB with the marker icon. They can delete the confirmed marker by clicking on the FAB with the crossed out marker icon.

[28] A dropdown list in Android.

The second location option is “Choose later from preferred locations”. The user can put up to five location markers on the map and later, after the meeting is created and the participants have sent their locations, the application calculates which location of those should be the optimal choice and recommends it to the user.

The third option is “Choose later from recommended places”. The user won’t have to choose any locations instantly, but later, after the participants have sent their locations, the user is asked to choose suitable location types (e.g. a café or restaurant) and they are given a list of recommended places to meet.

After choosing a suitable option, the user can exit the view by clicking on the FAB with the check mark or by hitting the back button.



Liivi 2, Tartu

Location choice
Specific location

Choose a location on the map or search from the search bar.

Slide up for more info.

Liivi 2, Tartu

Location choice
Choose later from preferred locations

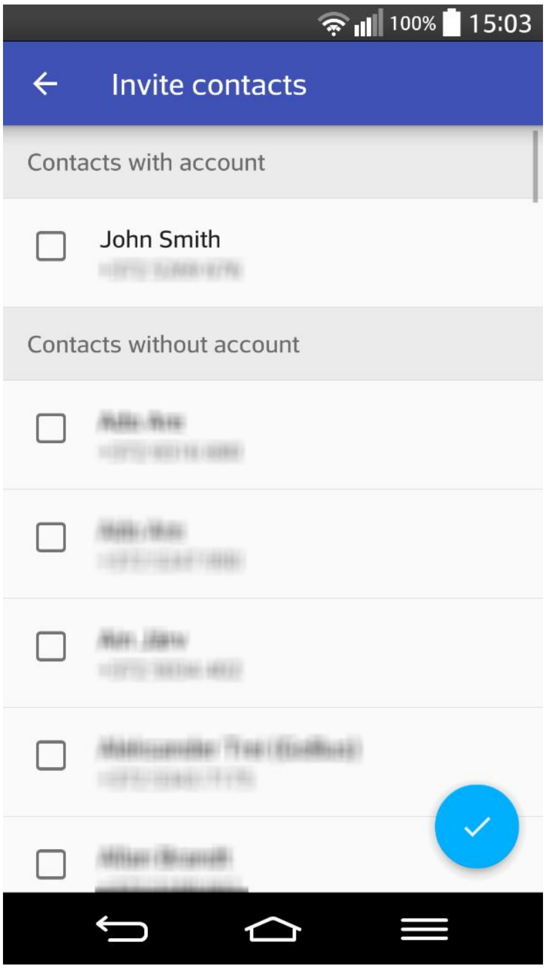
Choose several locations on the map and you'll later get a recommended location based on the positions of the participants.

Slide up for more info.

Figure 6. “Specific location” choice on the left
and “Choose later from preferred locations” choice on the right.

3.1.4.3 Inviting contacts

After tapping on the “Participants” field in the “New meeting” view, the user is shown the “Invite contacts” view (Figure 7). The view consists of a list containing all the Android contacts of the user. The list items are divided into two groups: “Contacts with account” and “Contacts without account”. Each list item displays the name and phone number of the contact. After choosing the contacts the user wishes to invite, they can either click on the FAB with the check mark or hit the back button to confirm selection.



Invite contacts

Contacts with account

□ John Smith

Contacts without account

□

□

□

□

□

Figure 7. The “Invite contacts” view. Personal information has been blurred for the sake of privacy.

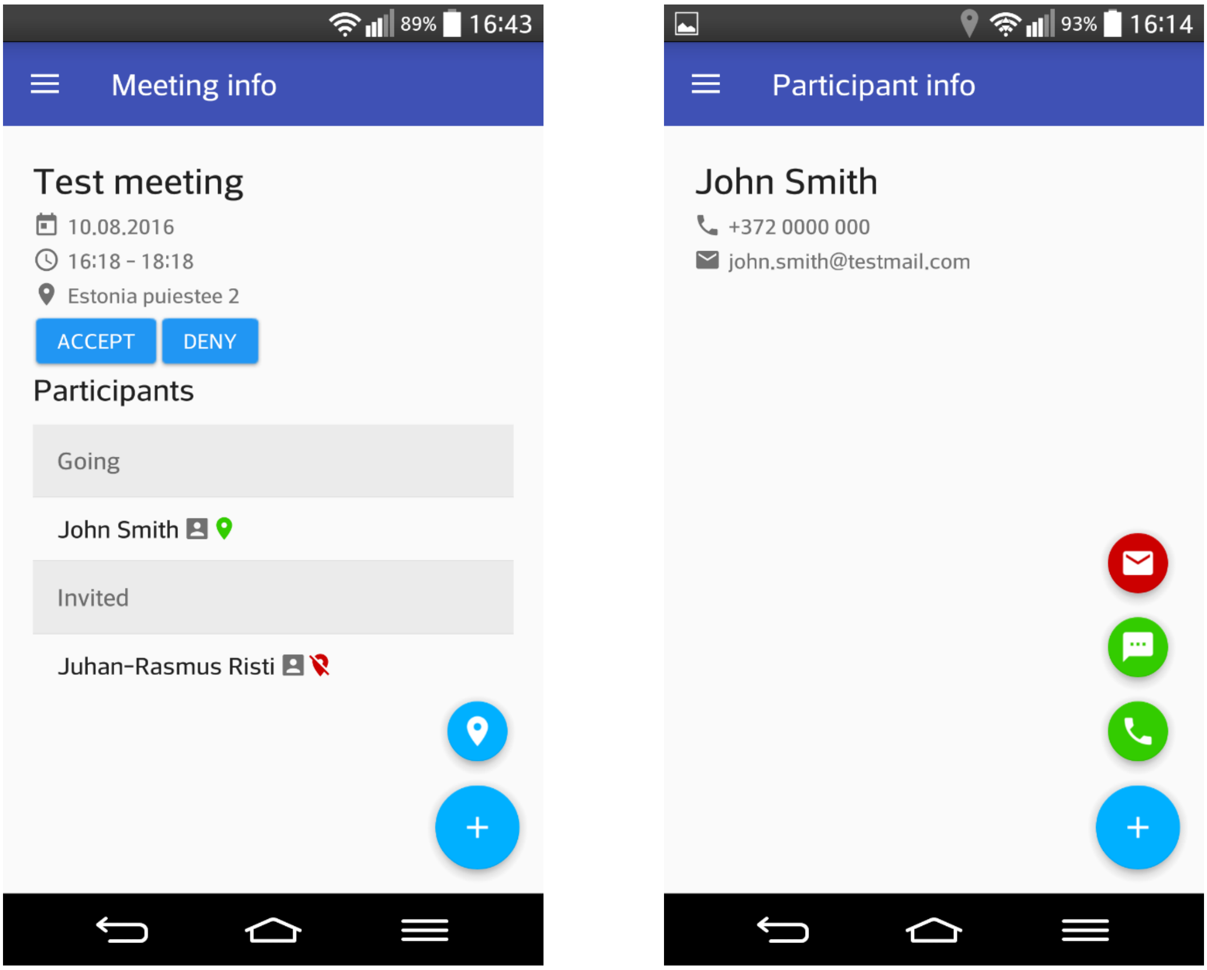
After the user has filled all required fields in the “New meeting” view, they can confirm the creation by hitting the confirm FAB.

3.1.5 The meeting info view

When the user creates a meeting or chooses a meeting from the meetings list, they are brought to the “Meeting info” view (Figure 8). The view consists of:

1. several text views which show the title, description, date, time and location of the meeting;
2. a dynamic layout section which can show one or two of the following:
   a. a check box which toggles whether or not the GPS location of the user is shared with other participants;
   b. “accept” and “deny” buttons which are shown when the user has yet not chosen whether they participate or not;
   c. a “choose location” button which is shown when the user chose a “choose later” location option;
3. a list of participants;
4. a small FAB with a marker icon, which opens a map view to show the meeting location;

The participants are grouped into “Going”, “Not going” and “Invited”. Each list item shows the name of the participant, an account icon if they have an account in the application and a green or red marker symbol. A green marker icon is shown when the user is currently sharing his GPS position and a red marker icon is shown when they are not sharing it. By clicking on a participant in the list, the user is shown a detailed view of the participant’s information.



Test meeting

10.08.2016

16:18 - 18:18

Estonia puiestee 2

Participants

Going

John Smith

Invited

Juhan-Rasmus Risti

John Smith

+372 0000 000

john.smith@testmail.com

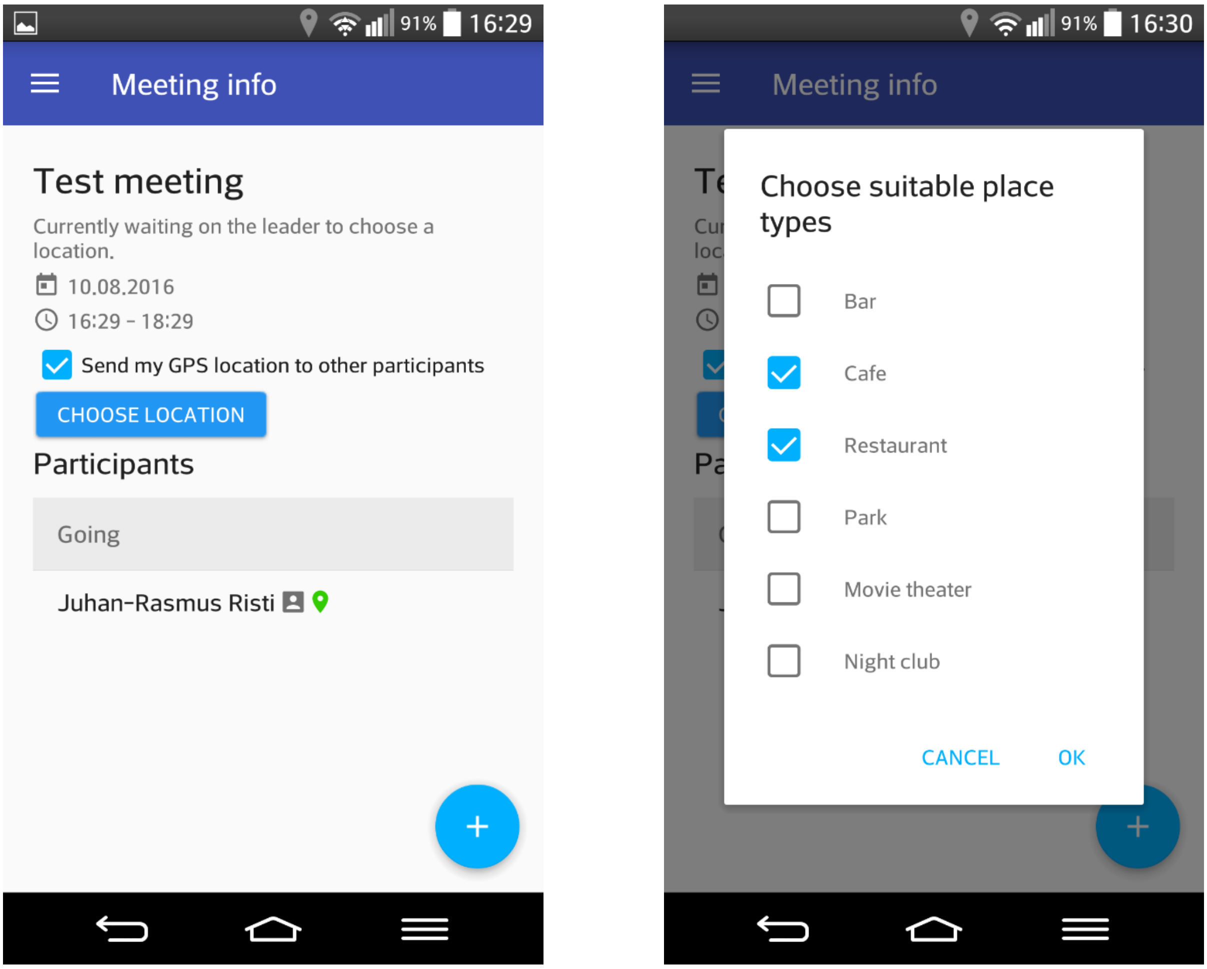
Figure 8. On the left is the view for a meeting where the user has yet not chosen whether they participate or not. On the right is the “Participant info” view.

3.1.5.1 The participant info view

The participant info view (Figure 8) displays the name, phone number and e-mail of the participant. In addition, it features three FABs: one to send an e-mail through an e-mail application; one to send a SMS through a messaging application; and one to call the person directly.

3.1.5.2 Choosing a recommended location

If the user chose a “choose later” location option earlier, they will have to choose a location after the participants have shared their location. The user will see a “Choose location” button in the “Meeting info” view (Figure 9). The recommended location will be calculated on the GPS locations of all the participants who shared theirs. It is necessary that at least one participant has shared their location, otherwise the user cannot continue.



Test meeting

Currently waiting on the leader to choose a location.

10.08.2016

16:29 - 18:29

☑ Send my GPS location to other participants

Participants

Going

Juhan-Rasmus Risti

Choose suitable place types

□ Bar
☑ Cafe
☑ Restaurant
□ Park
□ Movie theater
□ Night club

Figure 9. On the left is a meeting where the user can choose a location. On the right is the list of place types the user can choose when they earlier chose the “Choose later from recommended places” location option.

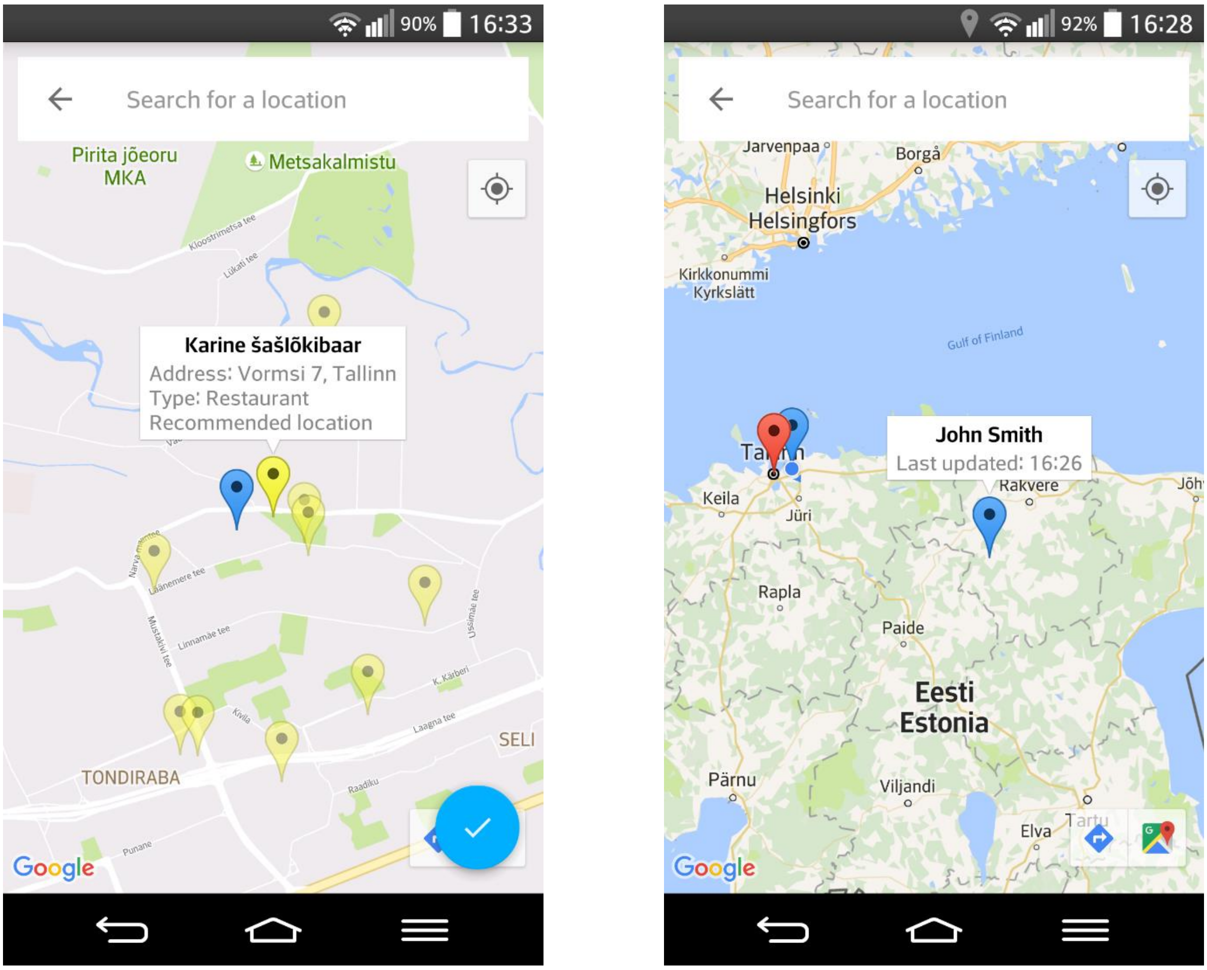
If the user clicks on “Choose location” and they earlier chose the “Choose later from recommended places” option, they will first be presented with a list of place types to choose from (Figure 9). After that the user will be sent to a map view where they can see up to 10 recommended places for the meeting (Figure 10). The most recommended place will be displayed as a non-opaque yellow marker, while the less recommended places will be opaque yellow markers. After choosing the preferred place, the user must click on the FAB with the check mark icon to confirm the selection.

If the user earlier chose “Choose later from preferred locations”, they will instantly be sent to the map view, where they can see the locations they earlier picked as the preferred locations. The recommended location will be a non-opaque orange marker and the less recommended locations will be marked as opaque orange markers.

3.1.5.3 Tracking other participants and showing the meeting location

When a meeting is ongoing, users can share their location with other participants by checking “Send my GPS location to other participants”. To track other participants, they must simply open the map view by clicking on the small FAB with the marker icon.

In the map view participants are displayed with blue markers and the meeting location is displayed with a red marker (Figure 10). By clicking on a blue marker, users can see the name of the participant and the time the participant's location was last updated. By clicking on a red marker, users can see the address of the location. If the meeting location is a specific place, they can also see the place name and type (e.g. restaurant).



Search for a location

Karine šašlõkibaar
Address: Vormsi 7, Tallinn
Type: Restaurant
Recommended location

Search for a location

John Smith
Last updated: 16:26

Figure 10. On the left the blue marker is the user and the yellow markers are the recommended places for the meeting. On the right the blue markers are the participants and the red marker is the confirmed meeting location.

3.1.6 The location recommendation algorithm

The location recommendation algorithm works in several steps:

1. Takes a list of participant objects as the input.
2. Calculates the centre geographical coordinate between the locations of the participants
3. Finds the nearest suitable locations to the centre coordinate.
    a. In case of the "Choose later from preferred locations" option, the algorithm
        i. calculates the distances between the centre coordinate and the locations the user earlier picked,
        ii. recommends the location with the shortest distance.
    b. In case of the "Choose later from recommended places" option:
        i. the user will be asked to choose suitable place types,

ii. the application will send queries to the Google Places API Web Service to get all the places which are in a 5 km radius of the centre coordinate and which are of the same type as the user chose,
iii. the algorithm measures the distances between the centre coordinate and every suitable place,
iv. sorts the places list by their relative distance to the centre coordinate;
v. returns the top 10 places nearest to the centre coordinate.

3.1.7 The participant tracker algorithm

When the user opens the "Meeting info" view and the meeting is currently ongoing, two services are started.

The first service will get the user's GPS location every 10 seconds and send it to the server. The server will update the location value for all ongoing meetings the user has and it will always send an answer back to the application whether or not it is necessary for the service to continue. The service will stop when it receives the answer to not continue anymore or when the user exits the application.

The second service will send a query to the server every 5 seconds asking for any updates about the currently viewed meeting. The service will also send a signal to the user interface to update the meeting information and participant markers in the map view. The service will stop when the user leaves the "Meeting info" view and map view.

It is possible for only one instance of each service to run at a time, so there is no risk of several instances of the same service accumulating.

4 Usability testing

The quality of the application was measured by usability testing. Usability testing refers to evaluating a product or service by testing it with representative users, who will try to complete typical tasks while observers watch, listen and takes notes. The goal is to identify any usability problems, collect qualitative and quantitative data and determine the participant's satisfaction with the product. [28]

A group of five participants (aged 20-23) were chosen for this test. They were all acquaintances, which helped against the problem that they would not have any contacts who have an account. Two of the participants had a technical background and the other three were everyday users of smartphones. Each participant was permitted to use their own phone for this test. They were given the following tasks:

1. create an account;
2. create a normal meeting with a specific location;
3. create a quick meeting with a recommended location option;
4. invite at least one friend and try to track their location;

In addition, they were permitted to do any extra tasks they would wish. At the end of the testing, they were asked

- whether they had encountered any problems;
- how would they rate the ease of use on a 5-point scale;
- would they use the application again;
- if they had any further suggestions;

4.1 Results

The following problems were found:

- On the newest Android versions there is a problem with the dynamic layout in “Meeting info” view: after accepting an invitation, the “Accept” and “Deny” buttons don’t disappear and the check box component is drawn on top of them.
- Three participants found it confusing that you can invite contacts who don’t have an account, but those contacts don’t receive any kind of invitation to the application. (Note: The solution to this problem is described in the “Future development” section.)
- When searching for a place on the map by its name, it usually finds the correct location but the application only saves the address. It should also save the place name.
- The Quick Meeting functionality should be better explained in the application, as currently it was a bit confusing.

The suggestions were as following:

- When the user has entered his e-mail and password in the login view and then moves to the registration view, the e-mail and password fields in the registration view should already be filled with the values entered in the login page.
- The phone number field in the registration view should already be filled with the number taken from the phone. (Note: The author researched this possibility earlier already, however most Android phones don’t store the phone number.)

- Currently the Meetings, Invitations and History views all display “No meetings found” when there are no available meetings. The views should each have their own message (e.g. “No invitations found”) to better distinguish them.
- Addresses should also show the city they are in.
- It would be nice to be able to send a message through the application. While it is possible to send a SMS quickly, it is not a free service.

Overall, the ratings for the ease of use were quite good with an average score of 4/5, as visualised on Figure 11. As visualised on Figure 12, two participants would not use the application again, two are not sure and one would use the application again, on the presumption that more people would start using it.

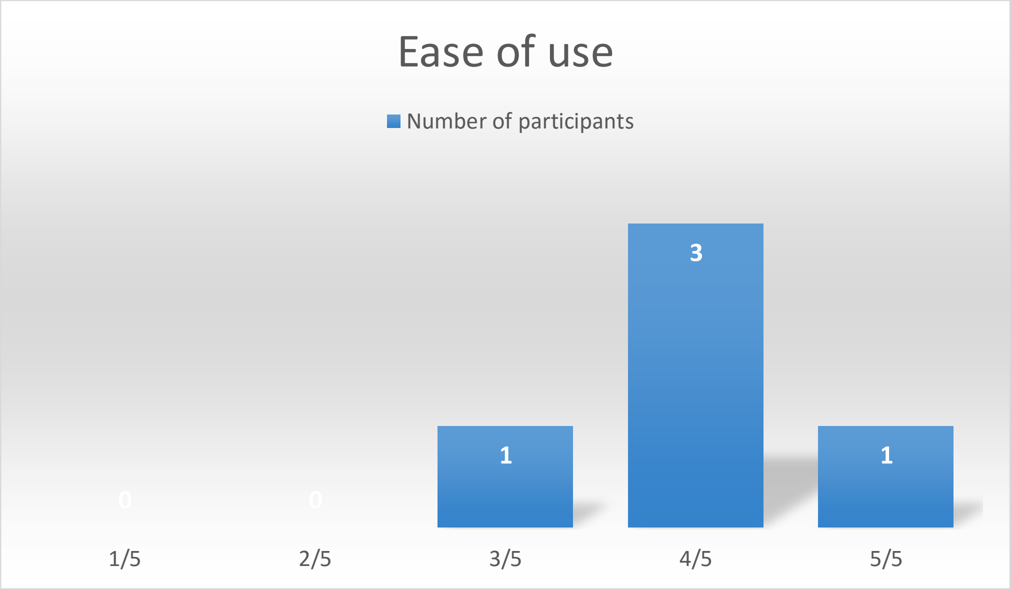


Figure 11. Ratings for the ease of use.

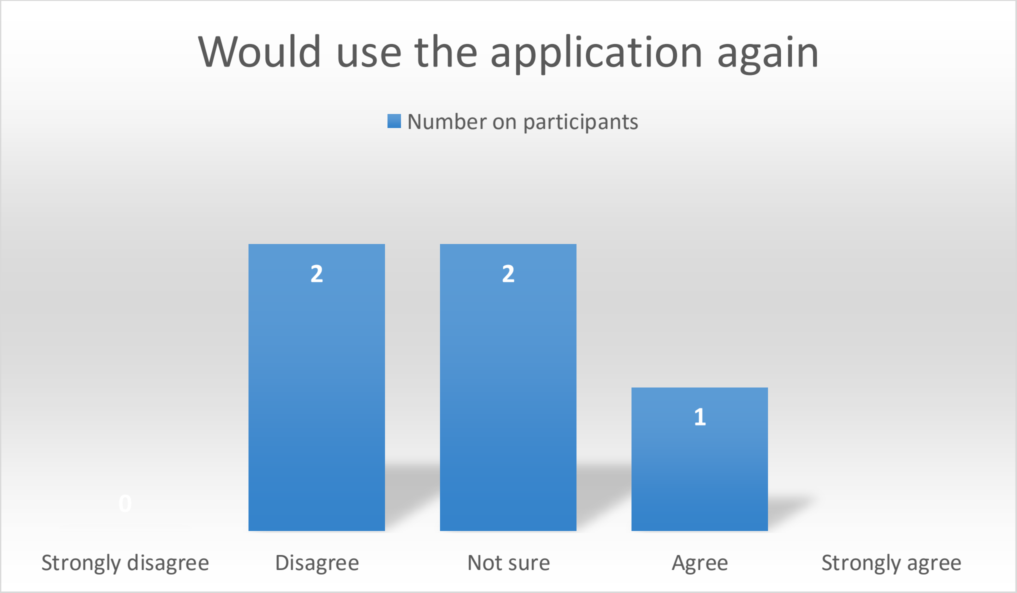


Figure 12. “Would use the application again” ratings.

5 Future development

In addition to the problems found and suggestions received in the usability testing, there are several ways to improve the effectiveness, secureness, ease of use and appeal of the application.

5.1 More efficient algorithm for recommending a meeting location

Currently the recommended location algorithm only considers the distance of each participant from each other. A more efficient algorithm should also consider the travelling speed of each participant and the roads that can be used, to get a more accurate estimation for the recommended location.

5.2 Push notifications

The application should be able to push notifications to the user using Google Cloud Messaging[29]. Notifications could be used to alert the user of a new invitation, of the beginning of a meeting or of any other important event.

5.3 Allow a meeting leader to change an existing meeting

The meeting leader should be able to change the details of an existing meeting, including changing the title, time, location and status of the meeting, and expelling participants.

5.4 Allow users to change details about their account

Users should be able to change various details about their account, including their name, phone number, email, password and participation in a meeting.

5.5 An account-related in-app contacts list

It should be possible for users to add other people to their contacts list from the application. This would be especially important in situations where the user logs in from another device which doesn't have his contacts list saved in it.

5.6 E-mail and phone number authentication

The application should require the user to authenticate his e-mail and phone number after registration to prevent identity theft. Before authenticating it should not be possible for the user to receive any invitations from other users, in case they entered someone else's credentials.

5.7 Possibility to log in with a Google or social media account.

To make the login and registration process simpler, it should be possible for users to log in with a Google or social media (e.g. Facebook) account. This feature could be used to allow the user also invite his Facebook contacts.

[29] https://developers.google.com/cloud-messaging/

5.8 An in-app messaging or commenting system

Although this doesn’t currently appear as a necessary functionality, allowing users to communicate between each other through the application would increase the application’s appeal and make it easier to use, as otherwise users would need to use a third-party application for communicating with each other.

5.9 Improve the markers in map layout

Currently the application uses standard markers in the map layout, which are coloured by their type. However, if the markers would visually give more information about the data they’re storing, it would make the application easier to use. For example, participant markers could show a picture of the person they’re representing, so users wouldn’t have to click on the marker to see whom it represents.

6 Summary

As a result of the thesis, an Android application was developed which allows users to create both quick and normal meetings in an easy way, get recommendations for choosing a place for their meeting, easily track other participants through the application, have several options to contact a participant quickly (send an e-mail, SMS or call from the application) and get a simple overview of all the user’s meetings.

The usability tests revealed some problems the author was not aware of, for example a layout bug in newer Android devices and the fact when a place is searched by its name, only its address is saved as the location. In addition, the five test participants had several suggestions to the author which mainly were related to the usability and visual side of the application. Overall, the test participants were quite satisfied as they gave an average of 4/5 rating for the ease of use and only two participants disagreed with the statement that they would use the application again.

7 References

[1] "Android (operating system)," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Android_(operating_system). [Accessed 11th August 2016].

[2] "Global mobile OS market share in sales to end users from 1st quarter 2009 to 1st quarter 2016," Statista, [Online]. Available: http://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/. [Accessed 11th August 2016].

[3] "Android Studio," Android Developers, [Online]. Available: http://developer.android.com/tools/studio/index.html. [Accessed 11th August 2016].

[4] "An update on Eclipse Android Developer Tools: Time to Migrate," Android Developers, 26 06 2015. [Online]. Available: http://android-developers.blogspot.com.ee/2015/06/an-update-on-eclipse-android-developer.html. [Accessed 11th August 2016].

[5] "Gradle," Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Gradle. [Accessed 11th August 2016].

[6] O. White, "How easy is the initial learning curve?," ZeroTurnaround, [Online]. Available: http://zeroturnaround.com/rebellabs/java-build-tools-part-2-a-decision-makers-comparison-of-maven-gradle-and-ant-ivy/2/. [Accessed 11th August 2016].

[7] O. White, "How fast are different builds with each tool?," ZeroTurnaround, [Online]. Available: http://zeroturnaround.com/rebellabs/java-build-tools-part-2-a-decision-makers-comparison-of-maven-gradle-and-ant-ivy/3/. [Accessed 11th August 2016].

[8] O. White, "How complex is it to create and maintain the build script?," ZeroTurnaround, [Online]. Available: http://zeroturnaround.com/rebellabs/java-build-tools-part-2-a-decision-makers-comparison-of-maven-gradle-and-ant-ivy/5/. [Accessed 11th August 2016].

[9] O. White, "How good is the community and documentation for each tool?," ZeroTurnaround, [Online]. Available: http://zeroturnaround.com/rebellabs/java-build-tools-part-2-a-decision-makers-comparison-of-maven-gradle-and-ant-ivy/7/. [Accessed 11th August 2016].

[10] O. White, "How many plugins exist and how simple is it to customize your own plugins?," ZeroTurnaround, [Online]. Available: http://zeroturnaround.com/rebellabs/java-build-tools-part-2-a-decision-makers-comparison-of-maven-gradle-and-ant-ivy/6/. [Accessed 11th August 2016].

[11] O. White, "How well does each tool integrate with developer tools?," ZeroTurnaround, [Online]. Available: http://zeroturnaround.com/rebellabs/java-build-tools-part-2-a-decision-makers-comparison-of-maven-gradle-and-ant-ivy/8/. [Accessed 11th August 2016].

[12] V. Chijwani, “Comparing Retrofit 2.0 vs. Volley,” 07 12 2015. [Online]. Available: http://vickychijwani.me/retrofit-vs-volley/. [Accessed 11th August 2016].

[13] “Introduction to the Google Maps Android API,” Google Developers, [Online]. Available: https://developers.google.com/maps/documentation/android-api/intro. [Accessed 11th August 2016].

[14] “Plans & pricing,” Mapbox, [Online]. Available: https://www.mapbox.com/pricing/. [Accessed 11th August 2016].

[15] “Google Maps APIs: Pricing details,” Google Developers, [Online]. Available: https://developers.google.com/maps/pricing-and-plans/#details. [Accessed 11th August 2016].

[16] “The Google Places API Web Service,” Google Developers, [Online]. Available: https://developers.google.com/places/web-service/intro. [Accessed 11th August 2016].

[17] “Places API Web Service: Usage Limits and Billing,” Google Developers, [Online]. Available: https://developers.google.com/places/web-service/usage. [Accessed 11th August 2016].

[18] “Rate Limits,” Foursquare Developers, [Online]. Available: https://developer.foursquare.com/overview/ratelimits. [Accessed 11th August 2016].

[19] R. Pham, “Material,” [Online]. Available: https://github.com/rey5137/material. [Accessed 11th August 2016].

[20] “Heroku,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Heroku. [Accessed 11th August 2016].

[21] “Free Dyno Hour FAQ: How many Free dyno hours will I receive?,” Heroku, [Online]. Available: https://devcenter.heroku.com/articles/free-dyno-hour-faq. [Accessed 11th August 2016].

[22] “Simple, Flexible Pricing,” Heroku, [Online]. Available: https://www.heroku.com/pricing. [Accessed 11th August 2016].

[23] “App Engine pricing,” Google, [Online]. Available: https://cloud.google.com/appengine/. [Accessed 11th August 2016].

[24] “Cloud Datastore: Highly Scalable NoSQL Database,” Google, [Online]. Available: https://cloud.google.com/datastore/. [Accessed 11th August 2016].

[25] “Openshift Monthly Plan Comparison,” Red Hat, [Online]. Available: https://www.openshift.com/pricing/plan-comparison.html. [Accessed 11th August 2016].

[26] “Spring Framework,” Pivotal, [Online]. Available: http://projects.spring.io/spring-framework/. [Accessed 11th August 2016].

[27] “Hibernate (framework),” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Hibernate_(framework). [Accessed 11th August 2016].

[28] “Usability Testing,” U.S. Department of Health & Human Services, [Online]. Available: https://www.usability.gov/how-to-and-tools/methods/usability-testing.html. [Accessed 11th August 2016].

8 Appendix

8.1 The source code

Client: https://github.com/juhanr/MeetingOrganizer

Server: https://github.com/juhanr/MeetingOrganizerServer

8.2 Installation instructions

1. Download the apk file from https://github.com/juhanr/MeetingOrganizer/raw/master/app/meeting-organiser.apk
2. Move the apk file to the phone.
3. Use a file manager application to browse the file on the phone and extract it.

8.3 Database model

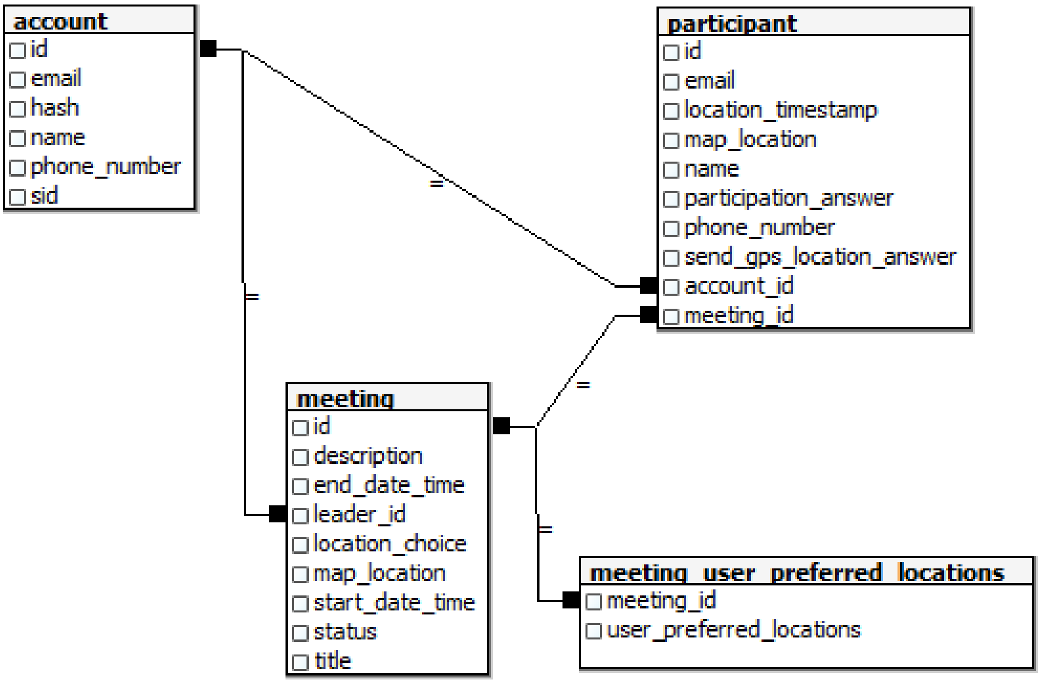


Figure 13. The entity-relationship database model.

8.4 License

Non-exclusive licence to reproduce thesis and make thesis public

I, Juhan-Rasmus Risti (date of birth: 16.03.1993),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

“Android application for organising meetings”

supervised by Satish Narayana Srirama and Liyanage Don Mohan Devapriya Liyanage

2. I am aware of the fact that the author retains these rights.

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

Tartu, **11.08.2016**