UNIVERSITY OF TARTU

Faculty of Science and Technology

Institute of Computer Science

Computer Science Curriculum

Kristjan Puusepp

Multipurpose Android application for reminders and expenses management

Bachelor's Thesis (9 ECTS)

Supervisor: Satish Narayana Srirama

Tartu 2018

Multipurpose Android application for reminders and expenses management

Abstract:

The aim of this bachelor thesis is to create an open source Android application which serves multiple purposes. Without registering an account in the application, the user can write up reminders and order them by title, creation date or end date. Additionally, the user can write up their expenses and have an overview of them in form of a pie chart. Registering an account allows users to sync their data with multiple devices, the main uniqueness of the application is the functionality to create groups and group reminders.

Furthermore, the thesis also includes comparison between other similar applications. As it is, other similar applications do not offer such functionality as to create groups and group reminders, those which do are usually very specific (meant for certain type, e.g. school groups).

Additionally, the thesis includes the tools used to create the application, the languages used and why such application is necessary.

Keywords:
Reminders, tasks, groups, expenses

CERCS:
P175 (Informatics, systems theory)

Mitmeotstarbeline Androidi rakendus meeldetuletuste ja kulutuste haldamiseks

Lühikokkuvõte:

Bakalaureuse töö raames valmis avatud lähtekoodiga Androidi rakendus, millel oli mitu eesmärki. Rakenduse kasutamine kasutajat loomata võimaldab üles kirjutada meeletuletusi ja sorteerida neid pealkirja, loomiskuupäeva või lõppkuupäeva järgi. Lisaks on võimalus märkida kulutusi, millest tekitatakse ülevaatlik sektordiagramm vastavalt kulutuse tüübile (toit, meelelahutus, transport, riideesemed, muu). Kasutaja loomine rakenduses võimaldab meeldetuletusi sünkroniseerida mitme seadme vahel. Rakenduse põhiliseks iseärasuseks on gruppide ja gruppide meeldetuletuste loomine.

Töö sisaldab ka võrdlust teiste sarnaste rakendustega. Praeguse seisuga ei võimalda teised sarnased rakendused luua gruppe ja gruppides meeldetuletusi, rakendused, mis seda võimaldavad, on sageli spetsiifilised (mõeldud kindlale sihtgrupile, näiteks õpilastele).

Lisaks sisaldab töö ülevaadet tööriistadest, mida kasutati rakenduse loomisel, ja põhjust, miks selline rakendus võib olla vajalik.

Võtmesõnad:

Meelespead, ülesanded, grupid, kulud

CERCS:

P175 (Informaatika, süsteemiteooria)

Table of Contents

List of Abbreviations

JSON	JavaScript Object Notation
IDE	Integrated Development Environment
APK	Android application package file
IO	Input/ Output
CEO	Chief Executive Officer
HTTPS	Hypertext Tranfer Protocol Secure
TLS/SSL	Transport Layer Security/ Secure Sockets Layer
TLD	Top Level Domain
BSD	Berkeley Software Distribution
ACID	Atomicity, Consistency, Isolation, Durability
CPU	Central Processing Unit
RAM	Random Access Memory

List of Figures

List of Tables

1 Introduction

1.1 Introduction

Owning a smartphone nowadays has become a norm and to many people even a real necessity. Life has become more fast paced and more people need to keep track of more things. That is why smartphones have become an important extension to some people's lives which help them to stay within their schedule, check their budget and much more[1].

It has become quite common to monetize applications, have restricting licenses on the source code and constantly have ads showing while using the application[2]. Even though this is quite commonplace, some users still prefer free application which are also ad-free[3].

Almost all popular Android phones have atleast some uninstallable applications which cluster the phone's screen[4]. This is why combining similar or daily-used applications to one application may become more necessary and may speed up daily affairs. Since smartphone's are getting faster and more powerful, combining rather simple and straight-forward applications to one does not affect the performance[5].

1.2 The solution

The aim is to create an ad-free Android application which serves multiple purposes. Without registering an account in the application, the user can write up reminders, order them by creation, end date or title. Additionally, the user can write up their expenses and have a fine overview of them in form of a pie chart. Registering an account allows users to sync their data with multiple devices, create groups and group reminders.

The end product shall be an Android application which is open-source, meaning everyone can see and obtain a copy of the code for themselves. This could be helpful towards existing Android developers or paving a road to new ones.

1.3 Outline

Section 2 – State of the art – describes the tools used and comparison between similar applications

Section 3 – Overview of the application – describes the end product, associated functional requirements and the usage and flow of the application. The section also includes information about server – describes the backend of the application. Additionally there is a subsection about datamining – describes the user permitted data usage. Last part describes the testing and evaluation of the application in the hardware level.

Section 4 – Conclusions and future work – summarizes the thesis and discusses the future development for the application

2 State of the art

2.1 Android

Android is an operating system for phones, tablets, smartwatches or other mobile devices and is based on a modified version of the Linux kernel and other open source software. It is being developed by Google, written in Java (UI), C (Core), C++ and other languages. The initial release of Android was in September 23, 2009. Compared to other main alternatives for mobile operating systems such as iOS and Windows Phone, Android has the biggest market share at 85% in the first quarter of 2017. Google also keeps the mobile operating system as a free-for-everyone and open-source[6,7,8].

As a result of these and the author's familiarity with Android programming the target operating system was chosen to be Android.

2.2 Android Studio 3.0

Android Studio[1] is an official IDE for Android and is based on the IntelliJ IDEA[2]. It is purpose built for Android and is intended to accelerate the development. It provides specific tools for code editing, debugging, testing and profiling[9].

2.3 Common IO 2.4 library

Common IO 2.4[3] library is an open-source Android library under the Apache Licence (Version 2.0)[4] which assist with developing useful IO functionality. The library is used in the application to parse and create JSON data for the communication between the application and server[10, 11].

2.4 MPAndroidChart library

MPAndroidChart[5] library is a free software under the Apache Licence (Version 2.0) which allows to create different types of charts (including line charts, bar charts, pie charts etc) with ease. The library is used in the application to create a pie chart of the expenses[12].

[1] https://developer.android.com/studio/features.html
[2] https://www.jetbrains.com/idea/
[3] https://commons.apache.org/proper/commons-io/
[4] https://www.apache.org/licenses/LICENSE-2.0
[5] https://github.com/PhilJay/MPAndroidChart

2.5 Similar solutions

There are several applications which offer the functionality to create reminders/ tasks but almost none of the most popular applications have the possibility to create groups and therefore create reminders/tasks for the aforementioned group. Most commonly other applications have implemented the functionality to send a reminder via Messenger, SMS or other way but almost none have the ability to just create groups and add reminders to that group and therefore eliminating the extra work to send reminders to other people. On top of that, the application allows users to add expenses in a separate view which gives the product a more multifunctional feeling.

Table 1 displays a comparison between the developed application and other popular and similar applications.

Only the Google App and Google Keep from the top ten most popular reminder applications have Estonian translation and are only translated if the device is set to be Estonian[13].

The goal of the author is to create an application for Android which can be set to Estonian without changing the whole device's default language and create reminders, create groups and group reminders and also add expenses and view these in a piechart and a list. Additionally, the whole Android application will be an open-source project and anyone can obtain the copy of the whole code and deal with it without restrictions. This may give start to new Android developers.

Table 1. Comparison table

Application / Functionality	Reminders and finance management	Any.do	GTasks	BZ Reminders
Necessary to create account	No	Yes	No	No
Premium for payment	No	Yes	Yes	Yes
Completed tasks/reminders are separately	Can be hidden	No	Can be hidden	In a separate view
Sorting reminders	Title, creation date, end date	Time, List	Date, Time, Priority	None
Setting alarm for reminder	No	Yes	Yes	Yes
Changing theme	None	5 different colours	7 different colours	Dark and light theme
Writing down expenses	Yes	No	No	No
Creating groups	Yes	No	No	No
Creating group tasks/ reminders	Yes	Functionality to send individual reminders via Messenger, WhatsApp, SMS		
Floating action button (this type of button is widely used)	Yes	Yes	No	Yes
Translations	Estonian, English	30+ languages	20+ languages	17 languages
Ads	No	No	No	Yes
Sync	Yes	Yes	Yes	Yes

3 Overview of the application

3.1 The application

This part of the thesis gives an overview of the architecture of the application, the functional requirements, describes the flow of the application and the views with their corresponding functinalities.

3.2 Prerequisisties

The application runs on Android 5.0 (Lollipop) and higher. The download link for the APK file can be found in the Appendix E.

3.3 Architecture of the application

The graphical overview/ flow of the application are displayed in Appendix A.

3.4 Use cases

Two use case diagrams are displayed in Appendix B. The use cases for the application are under Appendix C.

3.5 Functional requirements

MainActivity – view where user can switch between Reminders view, Groups view and Expenses view

GroupRemindersActivity – view where user can see group reminders and add new ones

Group – Set of members who can create group reminders

Group reminder – A reminder belonging to a group, all group members see it and admins can edit, delete it

(Group) Admin – Member of a group who can (besides creating new group reminders) delete existing group reminders, delete them, edit members (kick, make them admin), edit group name

CreateReminder – View where user can create a reminder

1. User has the possibility to create an account (register)
2. User can log in with their email and password combination
3. User can use the application without logging in (lacking some functionality)

   a. They cannot synchronize their personal reminders with other devices
      i. Uploads all personal reminders to server
      ii. Downloads all reminders not present in the current device (for example user has multiple devices, uploads different reminders from each device), synchronizing personal reminders between all devices
   b. They cannot create groups, edit groups and it’s members, see groups, see group reminders
4. User can see their reminders in a scrollable list
   a. Clicking on an existing reminder opens CreateReminder
      i. The view is filled with already existing data (title, content, date and time)
      ii. User can change the title, content, date or time
      iii. User can save changes or disregard them
5. User can sort reminders
   a. By creation date (newer or older first)
   b. By end date (newer or older first)
   c. By title (A-Z or Z-A)
6. User can create reminders (On either Reminders view or GroupReminders view)
   a. There should be a button in the bottom right corner of the screen
   b. Add title to reminder
      i. If no title is defined, it will automatically be „*no title*“
   c. Add content to reminder (Can be empty)
   d. Set date and time for the reminder
      i. If only date is chosen, time will be set as 00:00
      ii. If only time is chosen, date will not be set and user will see it as „Set date“
      iii. If none is chosen, user will see it as „Set date“
   e. Group reminders will be automatically uploaded to the server
7. User can mark personal reminder as completed
   a. There should be a checkbox in the list item (right side)
8. User can delete personal reminders
   a. Clicking on the reminder opens CreateReminder
   b. There should be a button (trashcan icon) and upon clicking deletes current reminder
9. User can add expenses
   a. There should be a button near the bottom right corner of the screen
   b. Add content (Can be empty)
   c. Add a cost
      i. If not marked, a Toast will pop up mentioning that cost is necessary
   d. Choose a category for the expense
      i. Food, Transport, Entertainment, Clothes, Other
      ii. If category is not chosen, it will automatically be „Other“

        e. User can set the date
            i. If date in not set, it will automatically be current day’s date
10. User can see the piechart created of existing expenses costs
    a. There should be a legend (colours) for each expense category
    b. There should be category names and totals sums on piechart slices
11. There should be a text mentioning total sum under the piechart
12. There should be a text mentioning total sums of categories under abovementioned text
13. User can click button in Expenses view which opens a new view containing list of all separate expenses
    a. User can click on the expense and edit it (change cost, content, date, category)
    b. User can delete the expense
14. User can switch between three main view ((personal) Reminders, Groups and Expenses) by using the bottom navigation bar in the MainActivity
    a. Active view will be highlighted in the bar
15. User can click the top right settings button (three dots) to open drop down menu
    a. In the MainActivity it shows five items
        i. Settings
            1. Clicking it opens view with different setting options
                a. Language
        ii. Sync
            1. Synchronizes the personal reminders
            2. If not logged in, a Toast will pop up mentioning that the user has to be logged in
        iii. Hide completed/Show completed
            1. Either hides the completed (personal) reminders from the list or shows them
        iv. Groups
            1. Clicking open new view with list of groups where the user is currently in
            2. If not logged in, a Toast will pop up mentioning that the user has to be logged in
        v. Login/Logout
            1. If user chose to continue to the application without registering/logging in, user can click „Login“ to get to the Login view
                a. In the Login view user can enter their email and password to log in
            2. If the user is logged in, drop down menu will show „Log out“
16. User can add new groups (in GroupsActivity view)
    a. Clicking a button opens up a Dialog

            i. User can enter group name and click „Save“ which creates a new group

17. Clicking on a list item (group name) opens new view with all group reminders
    a. User can see the content or edit (if admin) the reminder. See more in (4.)
18. User can click settings button on a group list item which opens dialog
    a. User can click member in the Dialog to open another view with members emails
        i. Admins can
            1. Kick players
            2. Make others also admin
    b. Delete group (if admin)
    c. Leave group
    d. Change group name (if admin)

3.6 First view, login and registration

When launching the application for the first time, the user is shown the startup view (Figure 1 (left)). From that view the user can either continue to the application by clicking the button „Continue without logging in“ and therefore use all the functionalities which do not require an internet connection.

The user can log in by clicking the „Log in“ button and then log in using their email and password in the login view.

The user can register by clicking the „Register“ button and then create an account in the registration view.

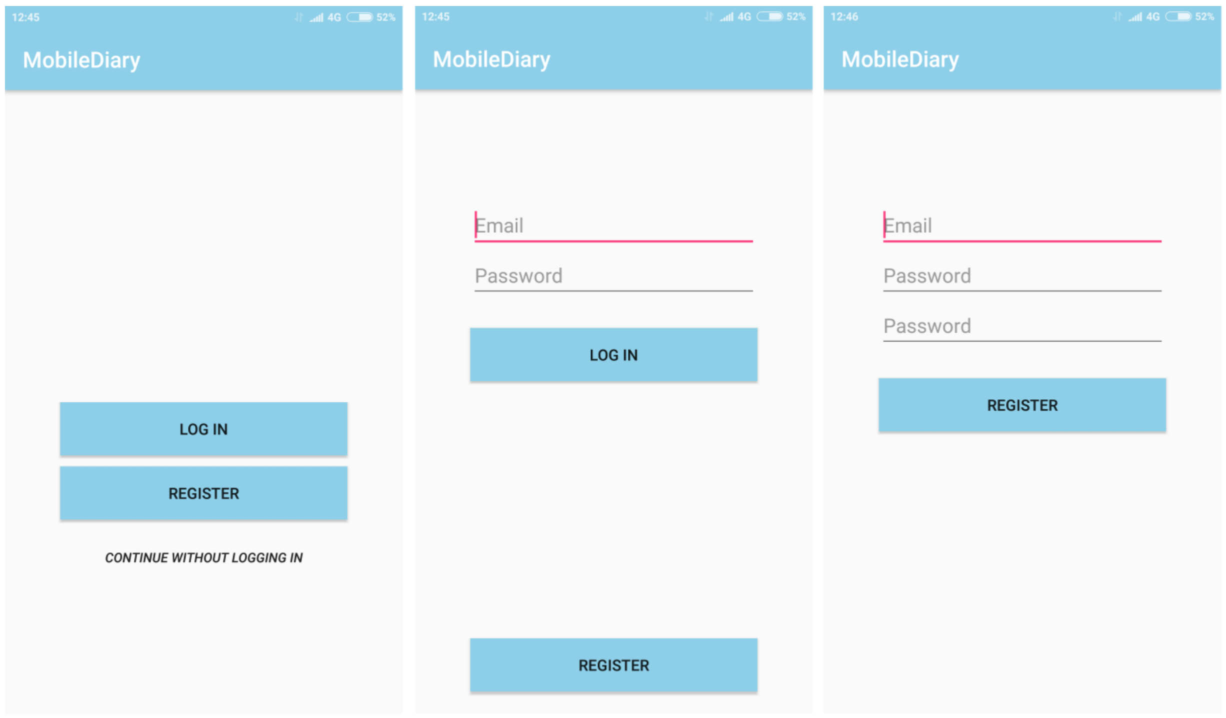


Figure 1. First screen (left), login view (middle), registration view (right)

3.7 Reminders view

After continuing without logging in, logging in or registering successfully, the user is brought to the reminders view (Figure 2 (left)). In the top right corner of the application the user can click on the kebab menu button to open the drop down menu (Figure 2 (middle)). If the user has logged in, the menu has four options:

1. „Settings“ – Change the language of the application (English/ Estonian). Check Appendix D for further information.
2. „Sync“ – Synchronize the data between the device and the server
3. „Hide completed“ – Hides the completed reminders from the reminders list
4. „Log out“ – Logs the current user out from the application

In the case that the user is using the application without having logged in, the drop down menu lacks the options for syncing and logging out. „Log out“ is replaced with „Log in“ option, which takes the user to the login view.

The reminders can be sorted by 3 different methods, each method with ascending or decending option. User can click on the „Sort“ button which opens up a drop down menu (Figure 2 (right)). From there the user can choose to sort their reminders by creation date (date when the reminders was created), end date (date which the user set for the reminder) and by the title.

All the main views (Reminders view, groups view, expenses view) can be accessed through the bottom navigation bar which is displayed at the bottom of all these three views (Figure 2 (left)).

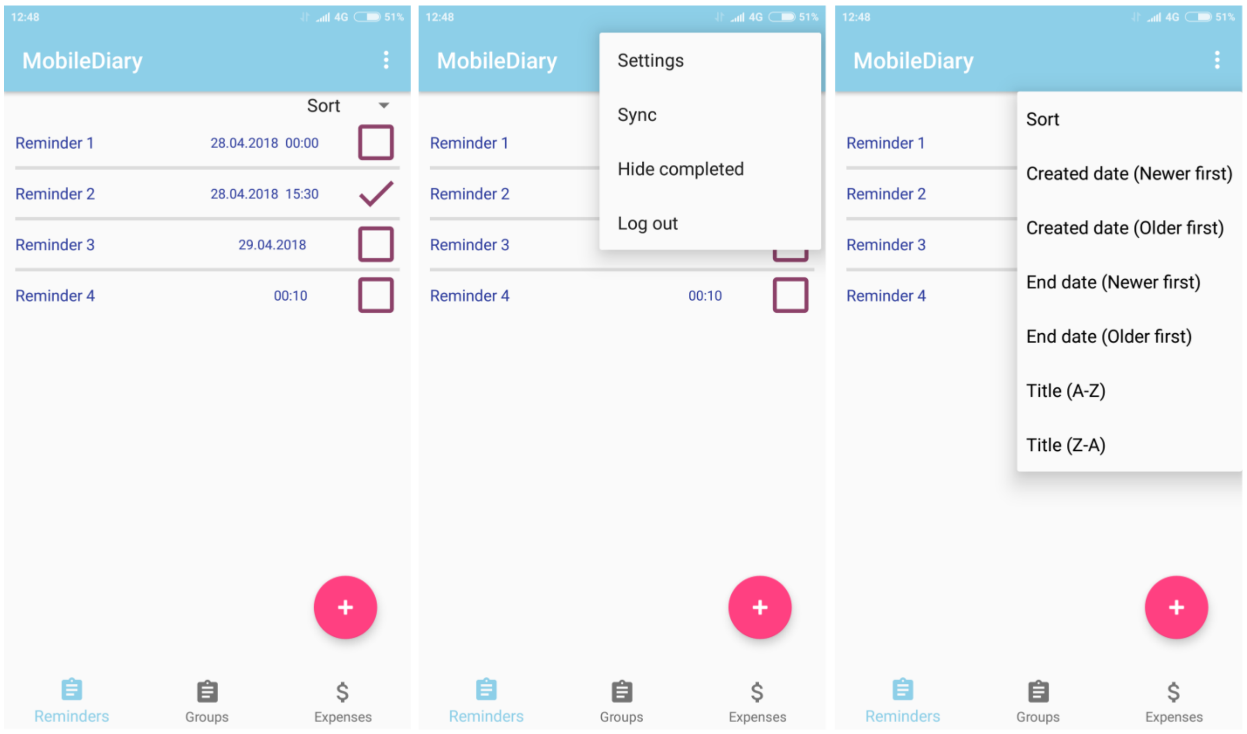


Figure 2. Reminders view (left), kebab button drop down menu (middle), sorting drop down menu (right)

3.8 Creating/ editing a reminder

In order to create a reminder the user has to click on the red floating action button in the bottom right corner of the reminders view (Figure 2 (left)). To edit an existing reminder the user can click on the desired reminder in the reminders view (Figure 2 (left)). This opens a new view where the user can add the information for the reminder or change the information for the existing reminder.

The view consists of the following (Figure 3):

1. Text field for the title of the reminder
2. Calendar button which opens the dialog window to choose the date
3. Clock button which opens the dialog window to choose the time
4. Text field for the content of the reminder
5. Button shaped like a „trash bin“ for deleting the reminder
6. „Cancel“ button to cancel the creation/ editing the reminder
7. „Save“ button to save the reminder

Filling in the title, the content or choosing date or time are all optional and user can leave the empty.

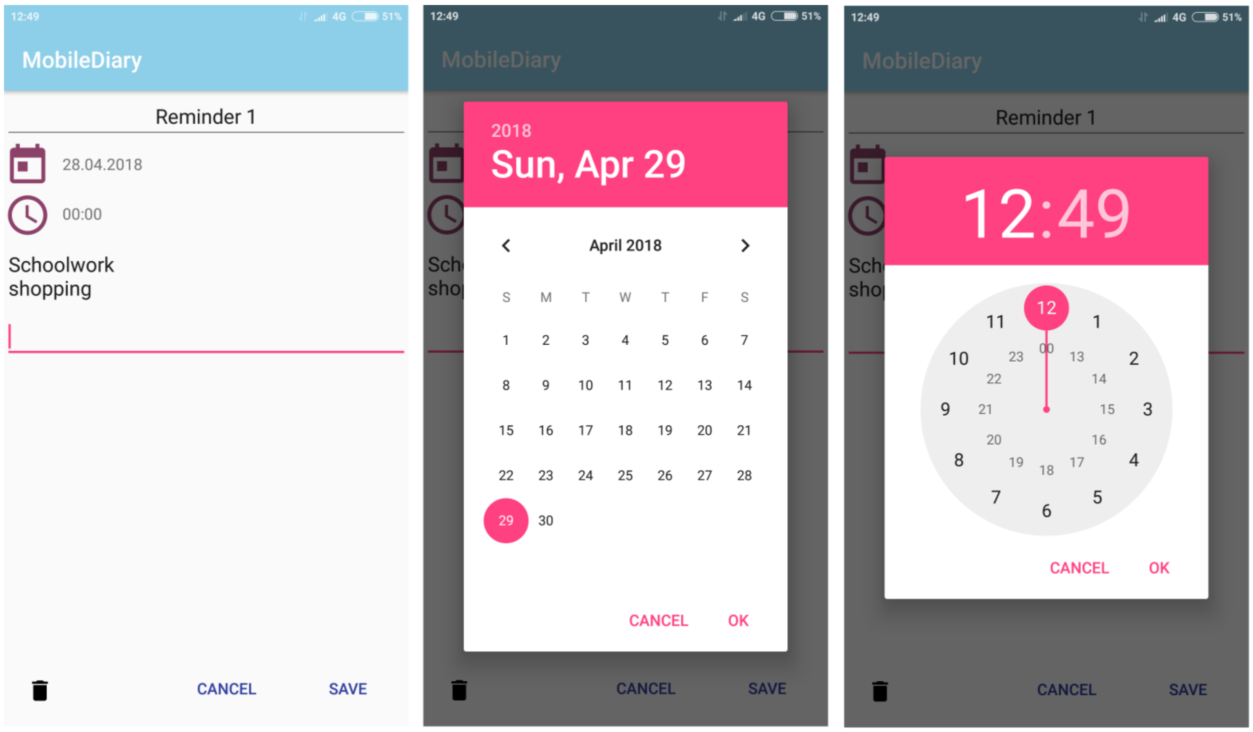


Figure 3. Reminder information (left), calendar window (middle), clock window (right)

3.9 Groups view

The view consists only of the list of groups and the red floating action button to add new groups (Figure 4 (left)). Clicking on the button opens a dialog window where the user can fill in the name for the new group and click „Save“ in order to create that group (Figure 4 (right)).

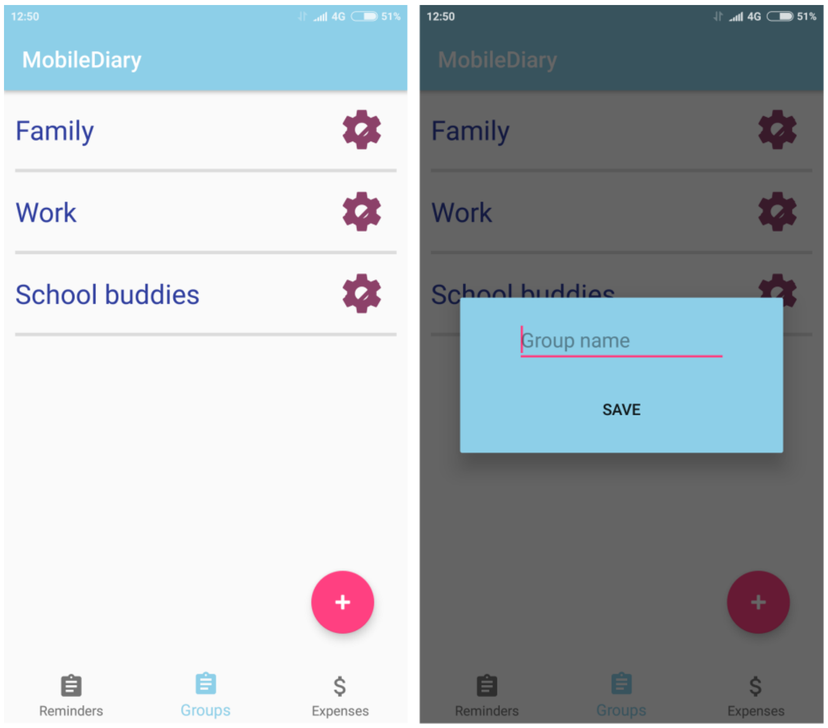


Figure 4. Groups view (left), creating new group (right)

Clicking on the gear-shaped button next to the group’s name opens a dialog window with following options (see Figure 5 (left)):

1. „Members“ (see Figure 7 (left)) – opens a view with the list of members
2. „Delete group“ – allows the admin to delete the chosen group (if the user is not admin, a message will be promted „You are not admin“).
3. „Leave group“ – removes the user from the group
4. „Change name“ (see Figure 5 (right)) – allows the admin to change the name of the chosen group (if the user is not admin, a message will be promted „You are not admin“).

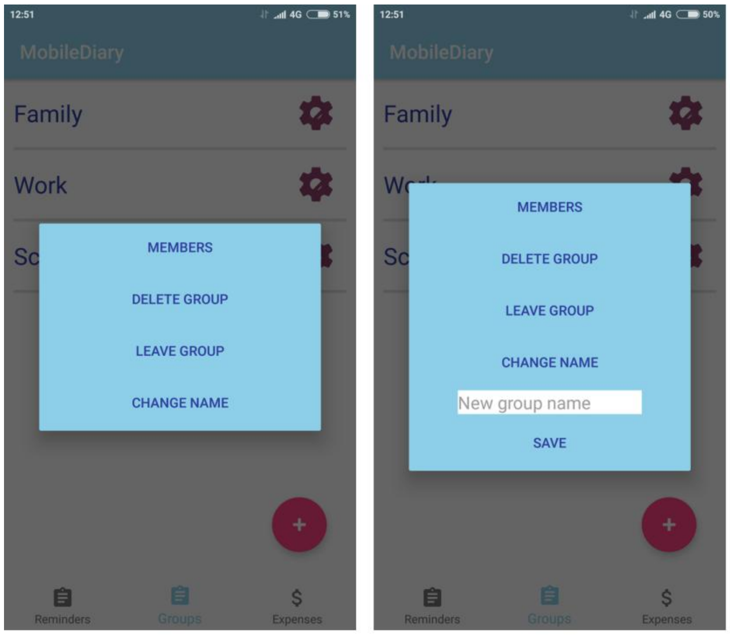


Figure 5. Group settings (left), changing group name (right)

Clicking on the group name from the list opens a view with the group reminders associated with that group (see Figure 6). Tapping on the reminder title opens a view where the user can edit the reminder (see Figure 3 (left)). The fields are already filled with the data related with the reminder and the user can change any field they deem necessary. In order to add a new reminder, the user can click on the red floating action button in the bottom right corner.

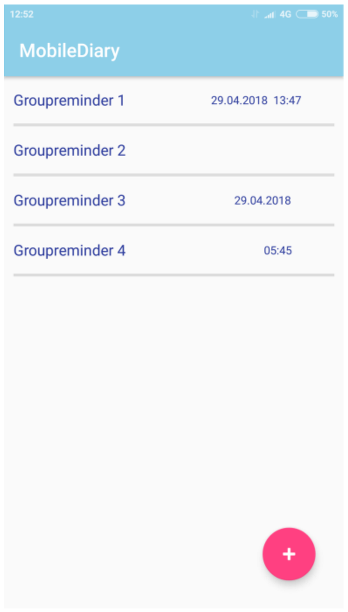


Figure 6. Group reminders list

Members are displayed in a list, showing their email-address and two buttons next to them (see Figure 7 (left)) . Admins have the functionality to grand admin privileges to other members in the group or kick them if necessary. The red floating action button in the bottom right corner allows all of the group members to add new members to the group (the email address must be a registered account in the application).

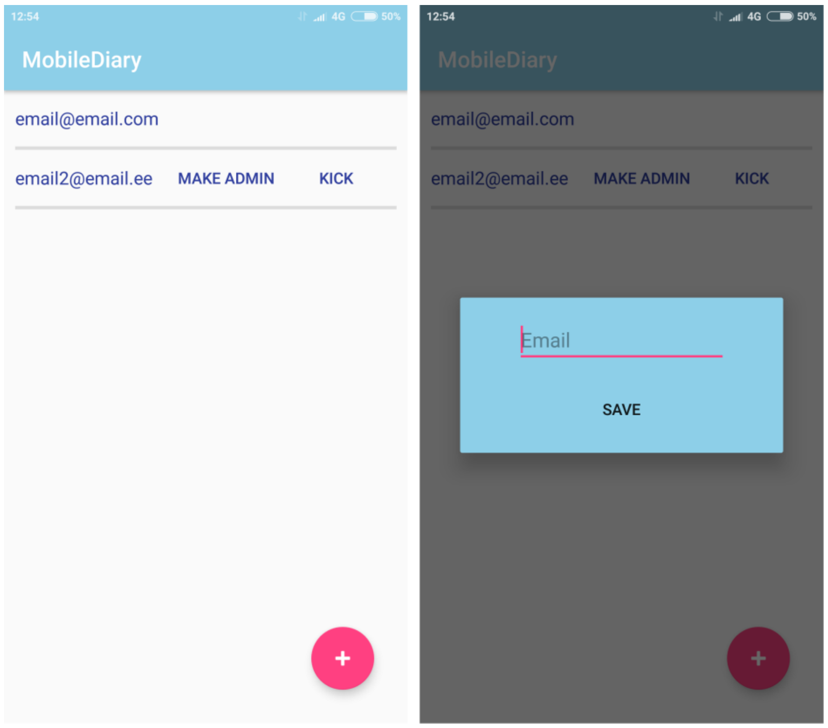


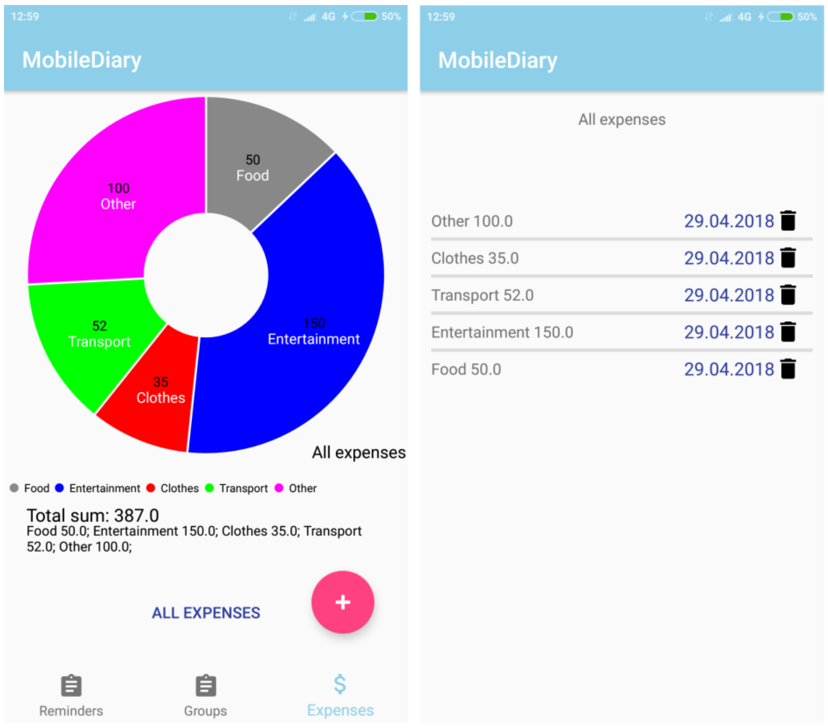
Figure 7. Group members list (left), adding a new member to group (right)

3.10 Expenses view

The main portion of the view displays all the expenses in a donut-shaped pie chart which are categorized by the type of the expense. Each sector's size is determined by the total expense in that category. The color legend for the chart is right below the it.

The total sum of the expenses and also the sum of the expenses of each category is brought out in a textual form below the color legend.

Tapping on the „All expenses“ button opens a new view which shows all the expenses in a list. From there the user can click on the expense in order to edit it (see Chapter „Adding/ Editing an expense“) or click on the button shaped like a „trash bin“ to delete the desired expense. The expenses in that list are sorted by their creation date (earlier dates are in the bottom of the list).



Total sum: 387.0
Food 50.0; Entertainment 150.0; Clothes 35.0; Transport 52.0; Other 100.0;

Other 100.0	29.04.2018
Clothes 35.0	29.04.2018
Transport 52.0	29.04.2018
Entertainment 150.0	29.04.2018
Food 50.0	29.04.2018

Figure 8. Expenses view (left) , expenses list (right)

If there are not any expenses added by the user, the chart is replaced with a text „No chart data available“ and clicking on the „All expenses“ button results in a pop up message saying „There are no expenses“.

3.11 Adding/ Editing an expense

In order to add an expense the user has to click on the red floating action button in the bottom right corner of the expenses view (see Figure 8 (left)). This opens a view where the user can add a new expense or edit an existing one.

The view consists of the following (see Figure 9):

1. Text field for cost (only allows numeric symbols)
2. Text field for content
3. „Choose category“ button which opens a dialog window from which the user can choose the category for the expense

4. „Set date“ button which opens the dialog window to choose the date (see Figure 3 (middle))
5. „Cancel“ button to cancel the adding/ editing of the expense
6. „Save“ button to save the expense

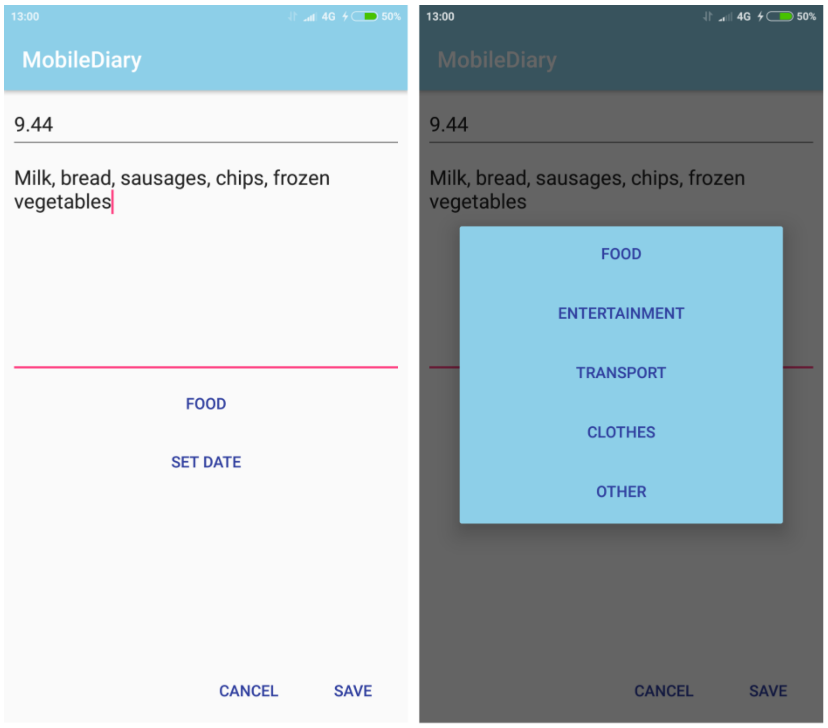


Figure 9. Add/edit an expense view (left), category dialog window (right)

Filling in the cost is required, leaving it empty and trying to save the expense will result in a pop up message saying „Please fill in cost“. All the other information associated with the expense can be left as is, saving the expense without choosing the date will automatically choose the current date and category will be saved as „other“. Content will remain empty.

3.12 Server

3.12.1 DigitalOcean

DigitalOcean, Inc. is an American cloud infrastructure provider. The corporation headquarters are located in New York City and the company has data centers worldwide. DigitalOcean provides developers cloud services that help to deploy and scale applications that run simultaneously on multiple computers. As of December 2015, DigitalOcean was the second largest hosting company in the world in terms of web-facing computers[14,15].

3.12.2 Freenom

Freenom (previously known as Freedom Registry) is a company based in Amsterdam, Netherlands. It is engaged in the provision of services for domain name registration and DNS resolution. Freenom was launched by the founder and current CEO Joost Zuurbier in December 2012. Freenom is part of the OpenTLD B.V. group that is globally active in the field of registering country code top-level domains (ccTLD), and through its subsidiarys Dot TK and Dot ML, the company provides domain registration, domain monetization and support services. Author chose Freenom for the domain name provider since there are free domain extensions available and domain name is necessary for enabling encrypted HTTPS on web servers[16].

3.12.3 Let’s Encrypt

Let's Encrypt is a Certificate Authority (CA) that provides free TLS/SSL certificates, thereby enabling encrypted HTTPS on web servers[17].

3.12.4 Flask

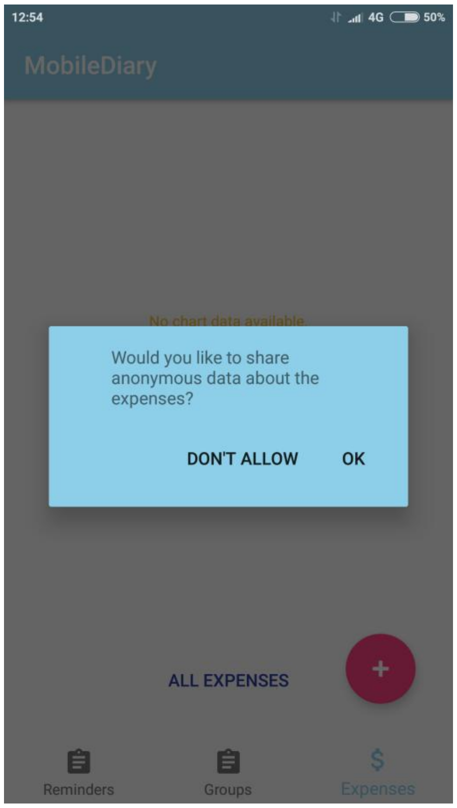
Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. It is BSD licensed. Flask is called a micro framework[18] because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program[19,20].

3.12.5 PostgreSQL

PostgreSQL, often simply Postgres, is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards compliance. PostgreSQL is ACID-compliant and transactional. PostgreSQL has updatable views and materialized views, triggers, foreign keys; supports functions and stored procedures, and other expandability[21]. PostgreSQL is developed by the PostgreSQL Global Development Group, a diverse group of many companies and individual contributors. It is free and open-source, released under the terms of the PostgreSQL License, a permissive software license[22,23].

3.13 Datamining

When the user open the expenses view for the initial time, a dialog window will promt a message asking the user if they are willing to share anonymous data of their expenses (Figure 10). If they allow it, then once a month their previous month's expenses will be uploaded to the server without associating that data with the user. Only the expense cost, type and date will be uploaded.



Would you like to share anonymous data about the expenses?

Figure 10. Expenses view initial dialog window

The application collects anonymous data about the expenses from users who have allowed it. The collected data consists only of expense cost, type and creation date. Expense description and user id are not stored in the server's database and therefore cannot be linked to a specific user.

The collected data will be displayed as charts on the website https://www.mobileapplication.ga/expenses.

The first chart will display all the gathered expenses in a form of pie chart. Sectors are divided into the different expense types and on each sector is shown the type's cost by percentage from the total cost.

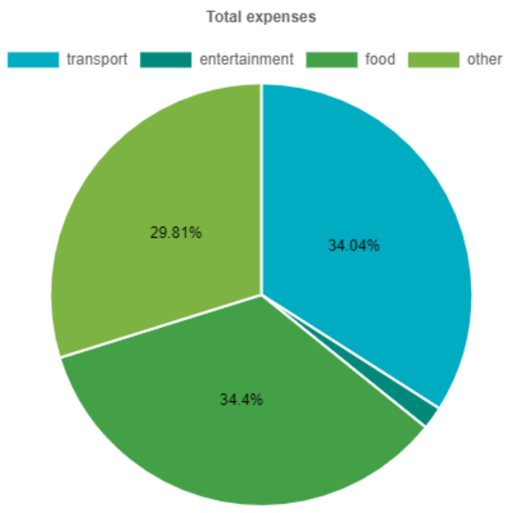


Figure 11. Piechart of expenses

Additionally, the data is formatted into horizontal bar charts and each differently colored bar represents a different type of expense. The chart shows data by year and that data is also divided into subgroups by the month. Website visitor can change the displayed chart by chaning the year (clicking on the buttons above the chart).

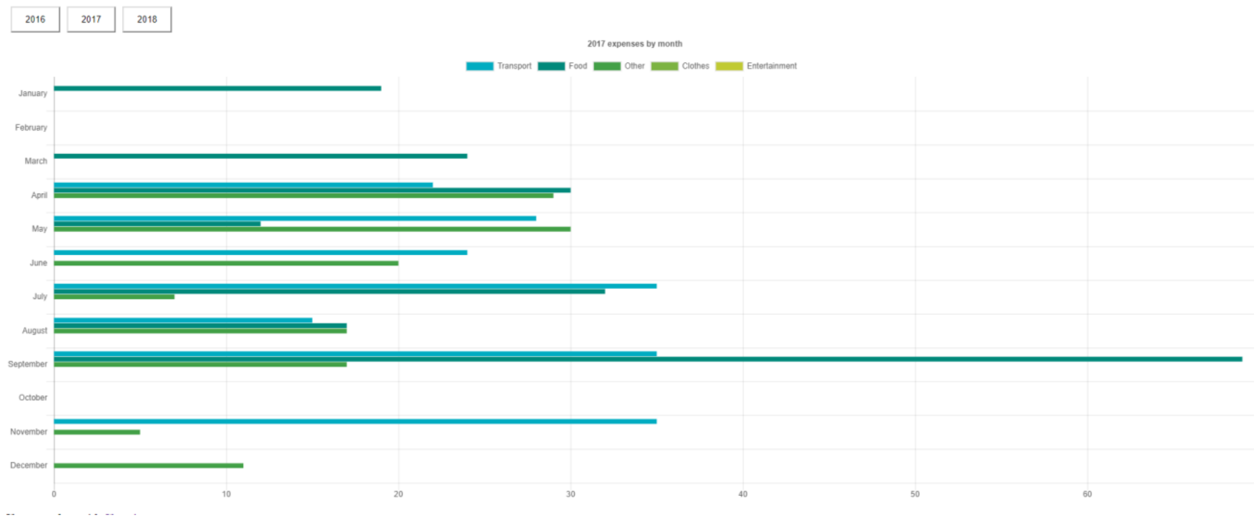


Figure 12. Horizontal barchart of expenses, by year, divided by months

The collected data will show how users usually spend their money. This data can be used for further analysis.

Furthermore, the gathered data will help the author to understand where and how much the application users spend their money. With this data the author can modify the application accordingly. For example, if the most represented category is food, the application will get subcategories for each type. If the most represented category is other, the application will get more categories to choose from or will give the user to create their own type/category for expense.

4 Testing and evaluation

The memory (RAM) consumption is around 20 Megabytes (Based on Android version, the allocated RAM size is around 32 Megabytes or higher) and the installed application itself takes around 8 Megabytes in the storage. The application itself is rather idle and uses CPU and internet connection only during user interactions. The figures below (Figure 13, Figure 14) indicate the CPU, bandwidth and RAM usage with the corresponding actions (listed below the figures). The battery usage (according to the AccuBattery application) during heavy use for 3 minutes (navigating through the views, adding/editing/deleting/sorting reminders, adding/deleting expenses, creating new groups and adding group members and reminders) shows that the Reminders and finance management application uses around 3.9% - 4.7% of battery in an hour on a Redmi Note 4 with 4000mAh battery.

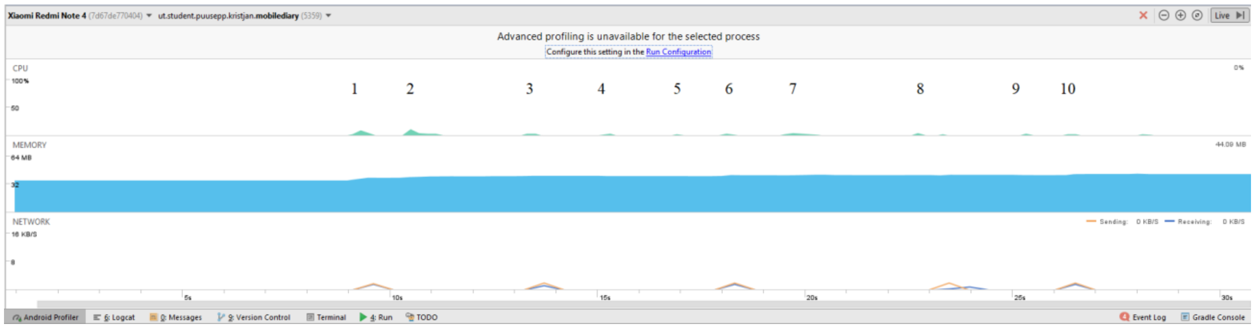


Figure 13. CPU, RAM and bandwidth usage 1

1. Open groups view
2. Open dialog window to add a new group
3. Add new group and reload the list in the view
4. Open group settings
5. Click „members“
6. Open view with members
7. Open dialog window to add new member email
8. Save new member
9. Click back button
10. Open the groups view and load the groups

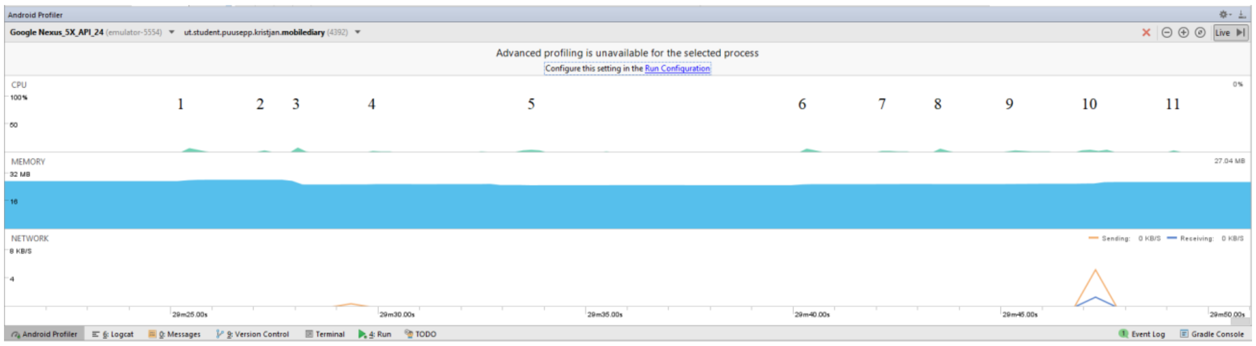


Figure 14. CPU, RAM and bandwidth usage 2

1. Click on red floating action button to add a new reminder
2. Open keyboard and fill in the title
3. Open date dialog window and choose date
4. Open time dialog window and choose time
5. Open keyboard and fill in the content
6. Click „Save“ button to save the reminder and be brought back to reminders view
7. Open sorting menu
8. Click on sort by title (A-Z) and refresh the reminders list in the view
9. Click on the kebab menu button in the top right corner
10. Sync reminders
11. Refresh the list in the view

5 Conclusions and future work

5.1 Summary

Owning a smartphone nowadays has become a norm and to many people even a real necessity. Life has become more fast paced and more people need to keep track of more things. It has become quite common to monetize applications and have ads showing while using the application. Almost all popular Android phones have atleast some uninstallable applications which cluster the phone's screen. This is why combining similar or daily-used applications to one application may become more necessary and may speed up daily affairs.

As the result of the thesis, an Android applcation was developed. The application provides the functionality to create personal reminders, create different groups with other application users and therefore add reminders to a desired group. Additionally, the software allows the user to add their expenses to keep track of their budget. The expenses are displayed in a a donut-shaped pie chart and can be viewed in a list.

5.2 Future work

Keeping track of the current trends, updating and improving applications have become very important in today's world because there are always arising new applications and better updates for existing ones which means that the competition in the software field is huge.

There are multiple ways in order to improve the current state of the application, including security, ease of use and speed.

The important upgrades include push notifications, setting alarms and ability to choose from between more languages and adding expenses categories. The notification functionality would be an excellent update because having the phone notify the user about new group reminders should enhance the user experience and help them keep track of the reminders. Setting alarms would assist the user in case they have forgotten about a reminder. Having more languages available means that the application could potentially reach more people since users intermittently prefer applications in their mother tongue. Providing more categories when adding an expense may help the users to better keep track of their expenses.

6 References

[1] „Are smartphones a necessity?“, Debate.org, [Online]. Available: http://www.debate.org/opinions/are-smartphones-a-necessity. [Accessed 29th of March 2018]

[2] „Mobile App Advertising Rates (2018)“, Mobyaffiliates, [Online]. Available: http://www.mobyaffiliates.com/guides/mobile-app-advertising-cpm-rates/. [Accessed 29th of March 2018]

[3] „New data on why people hate ads: too many, too intrusive, too creepy“, VIEODesign, [Online]. Available: https://www.vieodesign.com/blog/new-data-why-people-hate-ads. [Accessed 29th of March 2018]

[4] „Why Android phones now come with so many more google apps than before“, Gizmodo, [Online]. Available: https://gizmodo.com/why-android-phones-now-come-with-so-many-more-google-ap-1639529342. [Accessed 30th of March 2018]

[5] „With smartphones like these, why do we need laptops?“, Computerworld, [Online]. Available: https://www.computerworld.com/article/3241233/smartphones/with-smartphones-like-these-why-do-we-need-laptops.html. [Accessed 30th of March 2018]

[6] „What is Android“, Androidpit, [Online]. Available: https://www.androidpit.com/what-is-android. [Accessed 30th of March 2018]

[7] „Smartphone OS“, IDC, [Online]. Available: https://www.idc.com/promo/smartphone-market-share/os. [Accessed 30th of March 2018]

[8] Talbot, James and McLean, Justin. (2014). *Learning Android application programming*, pages 1-3. Boston (Mass.) : Addison-Wesley. [Accessed 29th of April 2018]

[9] „Everything you need to build on Android“, Android Studio, [Online]. Available: https://developer.android.com/studio/features.html. [Accessed 31st of March 2018]

[10] „Upgrade“, Apache Commons, [Online]. Available: https://commons.apache.org/proper/commons-io/upgradeto2_4.html. [Accessed 2nd of April 2018]

[11] Iverson, Will. (2005). *Apache Jakarta Commons*, pages 19-20. Visited on address http://ptgmedia.pearsoncmg.com/images/0131478303/downloads/Iverson_book.pdf [Accessed 29th of April 2018]

[12] „Home“, PhilJay/ MPAndroidChart, [Online]. Available: https://github.com/PhilJay/MPAndroidChart/wiki. [Accessed 20th of February 2018]

[13] „10 best reminder apps for Android“, AndroidAuthority, [Online]. Available: https://www.androidauthority.com/best-reminder-apps-for-android-654628/. [Accessed 31st of March 2018]

[14] „How DigitalOcean Won Over Investors“, Entrepreneur“, [Online]. Available: https://www.entrepreneur.com/article/247635. [Accessed 10th of January 2018]

[15] „Company overview of Digital Ocean, Inc“, Bloomberg, [Online]. Available: https://www.bloomberg.com/research/stocks/private/snapshot.asp?privcapid=243910980. [Accessed 10th of January 2018]

[16] „Company“, Freenom, [Online]. Available: http://www.freenom.com/en/aboutfreenom.html. [Accessed 11th of January 2018]

[17] „About Let’s Encrypt“, Let’s Encrypt, [Online]. Available: https://letsencrypt.org/about/. [Accessed 11th of January 2018]

[18] „What does “micro” mean?“, Flask, [Online]. Available: http://flask.pocoo.org/docs/0.10/foreword/#what-does-micro-mean. [Accessed 30th of March 2018]

[19] „Flask extenstions“, Flask, [Online]. Available: http://flask.pocoo.org/extensions/. [Accessed 30th of March 2018]

[20] „Flask extenstions“, Flask, [Online]. Available: http://flask.pocoo.org/docs/0.12/extensions/#extensions. [Accessed 30th of March 2018]

[21] „What is PostgreSQL?“, PostgreSQL, [Online]. Available: https://www.postgresql.org/docs/current/static/intro-whatis.html. Accessed [31st of March of 2018]

[22] „Contributor Profiles“, PostgreSQL, [Online]. Available: https://www.postgresql.org/community/contributors/. Accessed [31st of March 2018]

[23] „What is PostgreSQL?“, Postgrestutorial, [Online]. Available: http://www.postgresqltutorial.com/what-is-postgresql/. Accessed [31st of March 2018]

7 Appendices

A. Architecture of the application

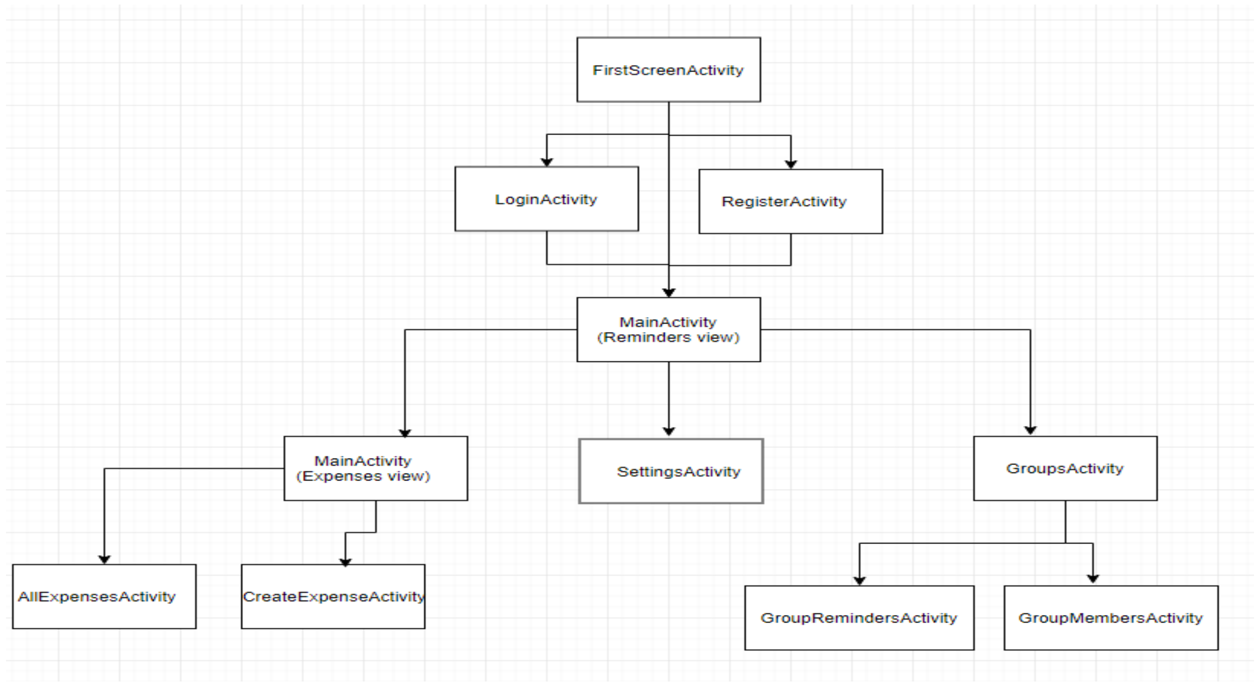


Figure 15. Architecture of the application

B. Use case diagrams

Associated use cases are use case 2 (logging in) and use case 9 (creating a group reminder).

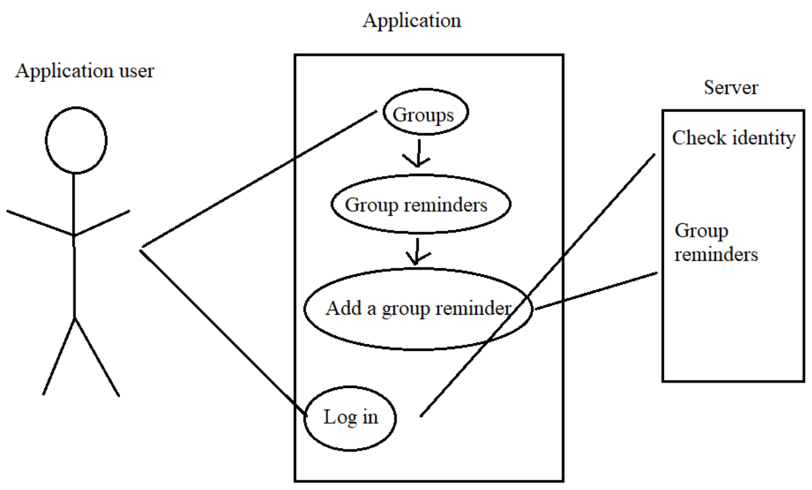
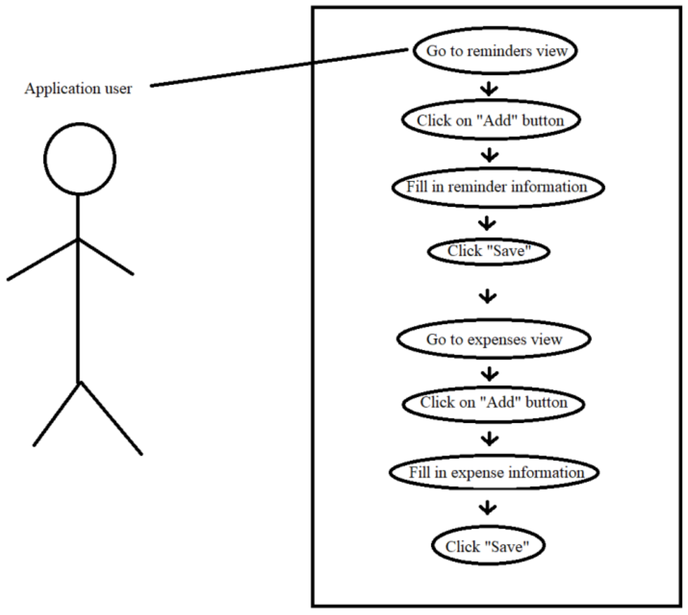


Figure 16. Use case diagram - adding group reminder

Associated use cases are use case 4 (creating a reminder) and use case 19 (adding an expense).



C. Use cases

Use case 1. Creating an account

Preconditions

1. Application is installed and opened.
2. The first screen is displayed

Primary actor

1. Application user

Main scenario

1. User clicks the „Register“ button
2. The application shows the register view
3. User fills in the email field
4. User fills in the two password fields
5. User clicks the „Register“ button

Postconditions

1. The account is successfully created
2. The user is brought to the reminders view

Alternative scenario(s)

1. User fills in two different passwords
    a. A message will pop up noting that the passwords do not match
2. User clicks the „Login“ button in the register view
    a. User is brought to the login view

Use case 2. Logging in

Preconditions

1. User has created an account
2. User has opened the application

Primary actor

1. Application user

Main scenario

1. User clicks the „Login“ button
2. User fills in the email field
3. User fills in the password field
4. User click the „Login“ button

Postconditions

1. User successfully logs in
2. User is brought to the reminders view

Alternative scenario(s)

1. User enters wrong email or password
    a. A message will pop up noting that the credentials are incorrect
2. User clicks the „Register“ button in the login view
    a. User will be brought to the register view

Use case 3. Continuing to the application without logging in

Preconditions

1. User is in the first screen view

Primary actor

1. Application user

Main scenario

1. User clicks the „Continue without logging in“

Postconditions

1. User is brought to the reminders view

Alternative scenario(s)

1. User misclicks on the „Register“ button
    a. User presses the back button
    b. User is brought to the first screen view

Use case 4. Creating a reminder

Preconditions

1. User is in the reminders view

Primary actor

1. Application user

Main scenario

1. User clicks on the „Plus-sign“ button (Floating action button)
2. The user is brought to the reminders creation view.
3. User fills in the title
4. User fills in the content
5. User clicks on the clock button
    a. User chooses the time from the pop up window
6. User clicks on the calendar button
    a. User chooses the date from the pop up window
7. User clicks on „Save“ button

Postconditions

1. Reminders is created
2. User is brought to reminders view

Alternative scenario(s)

1. User clicks „Save“ button without filling in title
    a. Reminder is saved without title
    b. Reminder is diplayed in the reminders view with the title „No Title“
    c. The user is brought to the reminders view
2. User clicks „Save“ button without choosing date or time
    a. Reminder is saved without date or time
    b. The user is brought to the reminders view
3. User clicks „Save“ button without filling in the content
    a. The reminders is saved without the title
    b. The used is brought to the reminders view
4. User clicks on the „Cancel“ button
    a. The reminder will not be saved
    b. The user is brought to the reminders view

Use case 5. Editing an existing reminder

Preconditions

1. User is in the reminders view

Primary actor

1. Application user

Main scenario

1. User clicks on the desired reminder
2. User is brought to the reminder creation view
3. The fields are already filled with existing data
4. User changes desired fields
   a. User can rename title
   b. User can change content
   c. User can choose a new time
   d. User can choose a new date
5. User clicks on the „Save“ button

Postconditions

1. Reminder is saved with new information
2. User is brought to the reminders view

Alternative scenario(s)

1. User clicks the „Cancel“ button
   a. Reminder will not be saved
   b. User is brought to the reminders view

Use case 6. Deleting a reminder

Preconditions

1. User is in the reminders view

Primary actor

1. Application user

Main scenario

1. User clicks on a desired reminder
2. User is brought to the reminder creation/editing view
3. User clicks on the trash bin icon in the bottom left corner

Postconditions

1. The reminder is deleted
2. The user is brought back to the reminders view

Alternative scenario(s)

1. The user clicks „Cancel“ button
    a. The reminder will not b deleted
    b. The user is brought back to the reminders view

Use case 7. Creating a group

Preconditions

1. User is in the groups view

Primary actor

1. Application user

Main scenario

1. User click the red floating action button in the bottom right corner of the view
2. A dialog windows opens with an empty input field
3. User enters desired group name
4. User click „Save“

Postconditions

1. New group is created
2. The list view of groups is refreshed

Alternative scenario(s)

1. The connection gets interrupted
    a. A message will pop up saying „Connection lost“
2. User fills in the wrong name for group
    a. User deletes the group (check use case 15)
    b. User repeats the steps of this use case

Use case 8. Adding a member to group

Preconditions

1. User is in the groups view

Primary actor

1. Application user

Main scenario

1. User clicks on the gear-shaped settings button next to the desired group name
2. A dialog window will pop up
3. User clicks on „members“
4. User is brought to the members view (of that group)
5. User clicks on red floating action button in the bottom right corner of the view
6. A dialog window will pop up
7. User writes desired email in the input field of the pop up
8. User clicks „Save“

Postconditions

1. New member is added to the group
2. Members list in the view is refreshed
3. A pop up message will say „New member added“

Alternative scenario(s)

1. User is not admin of that group
    a. A pop up message will say „You are not admin“

Use case 9. Creating a group reminder

Preconditions

1. User is in the groups view

Primary actor

1. Application user

Main scenario

1. User clicks on the name of the desired group
2. User is brought to the group reminders view
3. User clicks on red floating action button in the bottom right corner of the view
4. User is brought to the reminder creation/editing view
5. User follow the steps 3..7 in the use case 4

Postconditions

1. User is brought back to the group reminders view
2. New reminder is saved

Alternative scenario(s)

1. Connection gets interrupted
    a. A pop up message will say „Connection lost“

Use case 10. Sorting (personal) reminders

Preconditions

1. User is in the reminders view

Primary actor

1. Application user

Main scenario

1. User clicks on the „Sort“ drop down menu in the top right corner
2. A drop down menu opens with options:
    a. Sort
    b. Created date (newer first)
    c. Created date (older first)
    d. End date (newer first)
    e. End date (older first)
    f. Title (A-Z)
    g. Title (Z-A)
3. User clicks on the desired option

Postconditions

1. The reminders list is sorted by the according to the clicked option
2. The reminders list in the view is refreshed

Alternative scenario(s)

1. User clicks on unwanted option
    a. User repeats the steps of this use case

Use case 11. Marking (personal) reminder as done

Preconditions

1. User is in the reminders view

Primary actor

1. Application user

Main scenario

1. User clicks on the square box next to the desired reminder

Postconditions

1. The box changes into a check-shaped sign

Alternative scenario(s)

1. User clicks on wrong box
    a. User clicks on the check-shaped sign
        i. The reminder will me marked as undone
        ii. The check-shaped sign changes back into a box

Use case 12. Hiding completed reminders from the reminders view

Preconditions

1. User is in the reminders view

Primary actor

1. Application user

Main scenario

1. User click on the kebab menu button in the top right corner of the view
2. A drop down menu opens with options:
    a. Settings
    b. Sync
    c. Hide completed
    d. Log out/Log in
3. User clicks on the third („Hide completed“) option

Postconditions

1. The completed reminders will be hidden from the list in the reminders view

Alternative scenario(s)

1. User had already hidden completed reminders
    a. The drop down menu third option will be „Show completed“
        i. Clicking on it will show the completed reminders alongside with other reminders

Use case 13. Syncing (personal) reminders

Preconditions

1. User in the reminders view

Primary actor(s)

1. Application user

Main scenario

1. User click on the kebab menu button in the top right corner of the view
2. A drop down menu opens with options:
    a. Settings
    b. Sync
    c. Hide completed
    d. Log out/Log in
3. User clicks on the second („Sync“) option

Postconditions

1. The user reminders will be syncronized between the device and server
2. The reminders list in the view is refreshed

Alternative scenario(s)

1. There is no internet connection
    a. A pop up message says „Connection lost“

Use case 14. Editing group name

Preconditions

1. User is in the groups view

Primary actor

1. Application user

Main scenario

1. User clicks on the gear-shaped settings button next to the desired group name
2. A dialog window will pop up
3. User clicks on „change name“
4. An input field appears below the button „change name“
5. User fills in the desired new name for the group
6. User clicks „Save“

Postconditions

1. New group name is saved

Alternative scenario(s)

1. User types in wrong group name
    a. User repeats the steps of this use case

Use case 15. Deleting group

Preconditions

1. User is in the groups view

Primary actor

1. Application user

Main scenario

1. User clicks on the gear-shaped settings button next to the desired group name
2. A dialog window will pop up
3. User clicks on „delete“

Postconditions

1. The group is deleted

Alternative scenario(s)

1. There is no internet connection
    a. The group will not be deleted
2. The user is not admin of that group
    a. A pop up message says „You are not admin“

Use case 16. Leaving group

Preconditions

1. User is in the groups view

Primary actor

1. Application user

Main scenario

1. User clicks on the gear-shaped settings button next to the desired group name
2. A dialog window will pop up
3. User clicks on „leave“

Postconditions

1. User leaves the group and is no longer associated with said group

Alternative scenario(s)

1. User leaves the wrong group
    a. User needs to contact with an admin of that group to reapply
2. User is the only member of that group
    a. The group will be deleted

Use case 17. Deleting group reminder

Preconditions

1. User is in the groups view

Primary actor

1. Application user

Main scenario

1. User clicks on the name of the desired group
2. User clicks on the desired group reminder
3. User is brought to the reminder creation/editing view
4. User clicks on the trash bin icon in the bottom left corner

Postconditions

1. The reminder is deleted

Alternative scenario(s)

1. The user is not admin in that group
    a. The reminder will not be deleted
    b. A pop up message says „You are not admin“

Use case 18. Editing group reminder

Preconditions

1. User is in the groups view

Primary actor

1. Application user

Main scenario

1. User clicks on the name of the desired group
2. User clicks on the desired group reminder
3. User is brought to the reminder creation/editing view
4. User edits the desired fields
5. User clicks „Save“

Postconditions

1. The reminder is updated
2. The user is brought back to the group reminders view

Alternative scenario(s)

1. User edits the wrong reminder
    a. User repeats the steps of this use case twice
        i. First to fix the wronly edited reminder
        ii. Secondly to edit to correct reminder

Use case 19. Adding an expense

Preconditions

1. User is in the expenses view

Primary actor

1. Application user

Main scenario

1. User clicks on the red floating button in bottom right corner
2. User is brought to the expense creation/editing view
3. User fills in the cost
4. User fills in the content
5. User chooses the category from the pop up dialog window
6. User chooses the date from the calender (a pop up dialog window)
7. User clicks „Save“

Postconditions

1. User is brought back to the expenses view
2. The pie chart is updated
3. The text fields for total sum and separate sums are updated

Alternative scenario(s)

1. User does not fill in the cost
    a. A pop up message says „Please fill in the cost“
2. User does not choose category
    a. The expense category will be saved as „Other“
3. User does not choose date
    a. The expense date will be saved as the current date
4. User does not fill in content
    a. The expense will be saved with empty content

Use case 20. Editing an expense

Preconditions

1. User is in the expenses view

Primary actor

1. Application user

Main scenario

1. User clicks on „All expenses“ button
2. User is brought to the all expenses view
3. User clicks on desired expense
4. User edits desired fields
5. User clicks „save“

Postconditions

1. The expense is saved with new information
2. The user is brought back to the all expenses view

Alternative scenario(s)

1. User edits wrong expense
    a. User repeats the steps of this use case twice
        i. Firstly to fix the wrongly edited expense
        ii. Secondly to edit the desired expense

Use case 21. Viewing all expenses

Preconditions

1. User is in the expenses view

Primary actor

1. Application user

Main scenario

1. User clicks on „All expenses“ button

Postconditions

1. User is brought to the all expenses view

Alternative scenario(s)

1. User misclicks on the red floating action button for adding new expenses
    a. User clicks „Cancel“ or back button
    b. User is brought back to the expenses view
    c. User repeats the steps of this use case

Use case 22. Deleting an expense

Preconditions

1. User is in the expenses view

Primary actor

1. Application user

Main scenario

1. User clicks on „All expenses“ button
2. User is brought to the all expenses view
3. User clicks on the trash bin icon next to the desired expense

Postconditions

1. The expense is deleted
2. The list in the view is refreshed

Alternative scenario(s)

1. User deletes the wrong expense
    a. User has to recreate wronly deleted expense

Use case 23. Logging out

Preconditions

1. User is logged in

Primary actor

1. Application user

Main scenario

1. User click on the kebab menu button in the top right corner of the view
2. A drop down menu opens with options:
    a. Settings
    b. Sync
    c. Hide completed
    d. Log out/Log in
3. User clicks on the fourth options („Log out“)

Postconditions

1. User is logged out
2. User is brought to the first view

Alternative scenario(s)

1. There is „Log in“ instead of „Log out“ in the options
    a. The precondition had not been filled

Use case 24. Allowing data collecting

Preconditions

1. The user has not chosen to allow or disallow to share anonymous data in the pop up dialog window (the user has dismissed the dialog pop up window)

Primary actor

1. Application user

Main scenario

1. User clicks on „Ok“

Postconditions

1. The user has allowed to share data about expenses anonymously
2. Once a month the expenses data is uploaded to the server

Alternative scenario(s)

1. User clicks on „Don’t allow“
    a. No data about expenses is gathered

D. Settings

The user can change the application's language by clicking on either of the options: estonian or english (Figure 17).

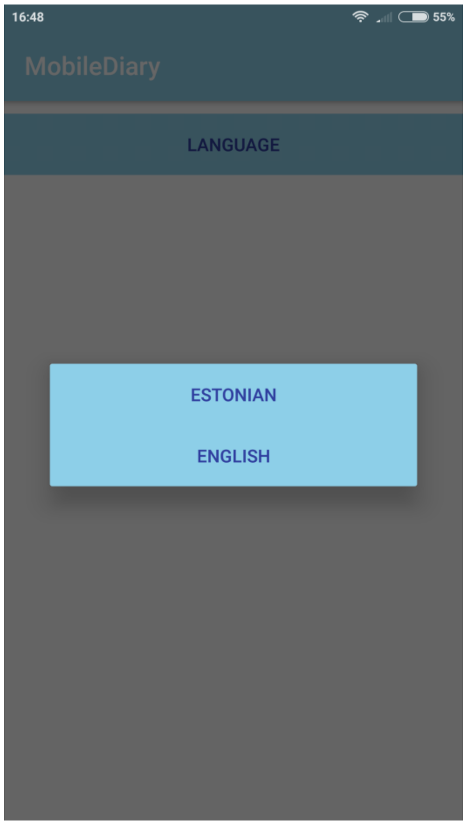
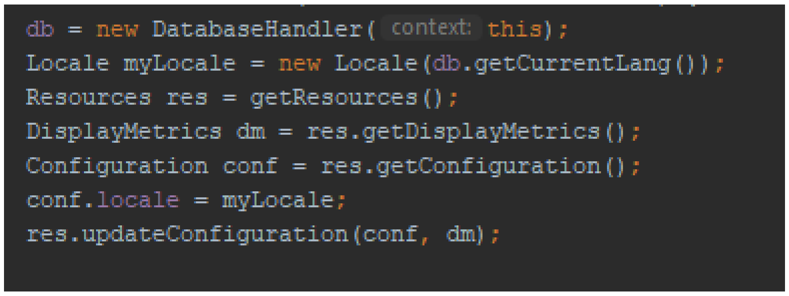


Figure 17. Settings view (langage dialog window)

Part of the code which handles the language of the application is depicted below (Figure 18). The language preference is stored in the database of the Android device.



```
db = new DatabaseHandler( context: this);
Locale myLocale = new Locale(db.getCurrentLang());
Resources res = getResources();
DisplayMetrics dm = res.getDisplayMetrics();
Configuration conf = res.getConfiguration();
conf.locale = myLocale;
res.updateConfiguration(conf, dm);
```

Figure 18. Code snippet (handles application's language)

E. Source code and APK file

The source code is available here:

https://bitbucket.org/qristjan/mobilediary/src/master/

The downloadable APK file is available here:

http://kodu.ut.ee/~qristjan/bachelorsthesis_apk_file/

F. License