

Bachelor MKI

Mobile Computing

SS 2020

Prof. Dr. Natividad Martínez Madrid

- Semester Project Documentation -

# Neighbour in Need

Presented by:

Fanni Tamara Marosi, 764345

[Fanni\\_Tamara.Marosi@Student.Reutlingen-University.De](mailto:Fanni_Tamara.Marosi@Student.Reutlingen-University.De)

Ebru Selin Özcelik, 764349

[Ebru\\_Selin.Oezcelik@Student.Reutlingen-University.De](mailto:Ebru_Selin.Oezcelik@Student.Reutlingen-University.De)

6. Semester

Submitted on: 15.07.2020

## Table of Contents

1	Introduction.....	3
1.1	Motivation .....	3
1.2	Goals of the project .....	3
2	Personas and requirements analysis .....	3
2.1	User groups and personas .....	3
2.2	Storyboards .....	7
2.3	Requirement specification.....	8
2.3.1	Functional requirements.....	8
2.3.2	Non-functional requirements .....	13
2.3.3	Nice-to-have requirements .....	14
3	Mock-Ups .....	14
4	Some design decisions about the implementation .....	16
4.1	Firestore.....	16
4.2	How to save the current user .....	17
4.3	Main, Register and Login Activities .....	17
4.4	Upload Ad-Image and Userimage .....	17
4.5	Bottom Menu .....	17
4.6	RecyclerView.....	17
4.7	Create Advertisement Activities.....	18
4.8	Sending an Email to another user .....	18
5	Current Design .....	18
6	Results.....	20
6.1	Time schedule and project progress .....	21
6.2	Testing and evaluation .....	21
6.3	Degree of completion of our requirements .....	21
7	Feedback.....	22
8	Conclusion.....	23

# **1 Introduction**

This elaboration was written as part of the module “Mobile”. We implemented an Android application for the communality, which is called “Neighbour In Need”. Users can create and search for advertisements.

## **1.1 Motivation**

In times of the Coronavirus we thought about implementing an application, which helps people to find/offer help/things to others in their neighbourhood. The two of us wanted to help in the time of Corona lockdown, but we made the experience that there weren't many platforms in order to connect to our neighbours and find someone who is looking for help. With this experience, we decided that there is need of such an application. It should be easily accessible for everyone and intuitive to understand and use.

## **1.2 Goals of the project**

The goal of this project was to implement an Android App, which has real benefits for the users. We wanted to create an App, which people would really use in their daily life. Users can register and create an account by specifying some information about themselves. This information contains a username, a password, an Email-address, the city, they live in and the post code. When logged in, users can create and search for advertisements, where others in their communality offer/search for things or help. They can also create advertisements. Users could connect via the App and build their community.

# **2 Personas and requirements analysis**

This section includes our personas and their use case scenarios, storyboards and the requirement specification.

## **2.1 User groups and personas**

Our user group is very heterogeneous and for this reason we have three personas representing different user groups. Everybody except children could theoretically use and benefit from the App.

Amy Sue is a young student, Petra a middle-aged woman and Thomas is an older man. We introduce them and their use case scenarios on the following pages.



Name: **Amy Sue**

Gender: Female

Age: 24

Hobbies: Cooking, Tai Chi, playing Cello

Studies: Biology

Amy Sue lives in Tübingen since 5 years. She is a friendly and outgoing girl. In her free time she plays with her dog and the neighbour's children. The children love her dog. She drinks a glass of multivitamin or pomegranate juice daily.

### Key characteristics:

- Passionate about her studies
- Spends a lot of time with her dog
- Enjoys watching movies in the cinema
- Very reliable and helping person
- Loves Italian food

### Goals:

- After she has finished university, she wants to work in a zoo
- Likes to help others, whenever she can
- Wants to go on a road trip

### Use Case Scenarios:

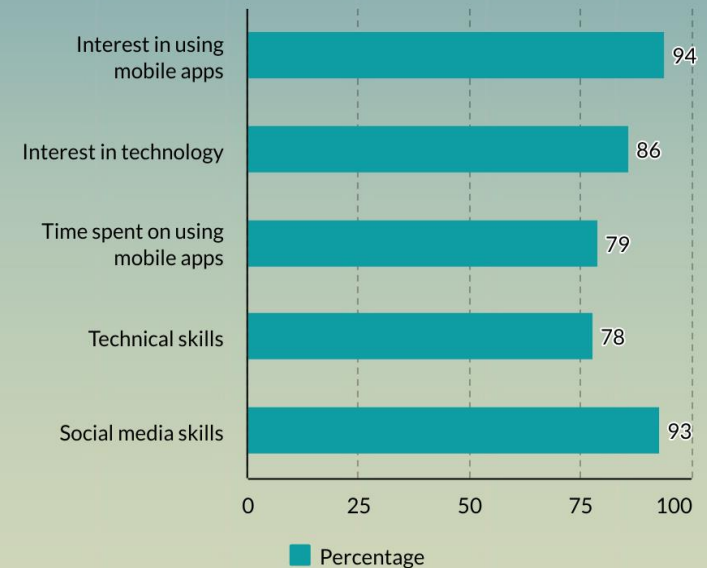
#### Searching for help:

Amy Sue doesn't own a car, but loves drinking juice. She struggles carrying her drinks home. She would like to have someone to drive her to the supermarket or lend her a car.

#### Offering help:

In semester break she has a lot of free time and would like to offer her help to others in need.

### User behaviours





Name: **Petra**

Gender: Female

Age: 39

Hobbies: Cooking,  
Writing, Reading

Petra works as a free journalist for a health magazine. She can research and work from home. She is married and has two children, the older one (7 years old) is adopted. Her husband works as a craftman. Most of the time he is not at home. On the weekend they spend time together.

#### Key characteristics:

- Do-It-Yourself lifestyle (crafty mom)
- She has a deep well of creativity to keep her kids occupied on a rainy day
- Positive attitude

#### Goals:

- She wants to make her stuff from scratch, produce her own food
- Do sports daily
- Spend time with her family and kids

#### Use Case Scenarios:

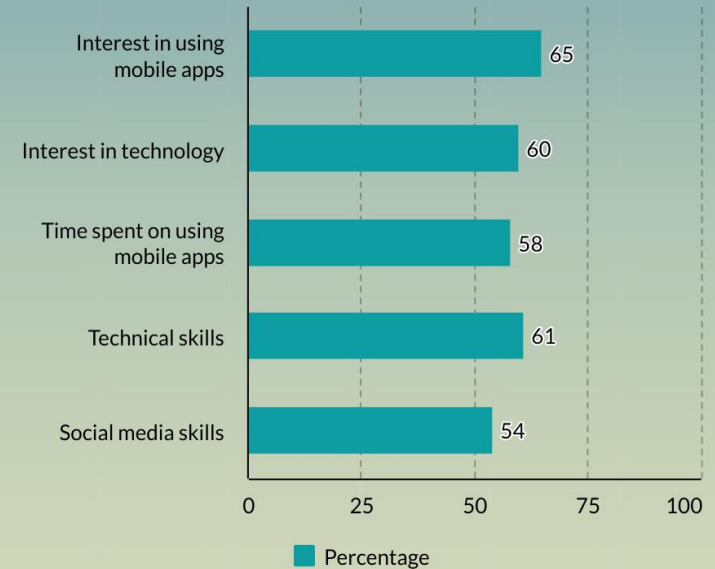
Searching for a gift:

She likes to handcraft with her children and therefore she often needs old toys and other old stuff.

Lend something:

Owens a dehydrator and could lend this to others, e.g. for drying spices. A dehydrator is quite expensive and so there will be interest in borrowing one.

#### User Behaviours





Name: **Thomas**

Gender: Male

Age: 82

Hobbies: Reading,  
gardening,  
watching TV

Thomas worked his whole life as a teacher for chemistry and mathematics. He retired 16 years ago and had a great time with his wife. Unfortunately she died a few months ago. It is not the same for him since she has gone. He has a son, who lives one hour away. Usually they see each other twice a month. In his free time he likes to play Boule with some friends.

### Key characteristics:

- Spends a lot of time in his garden
- Loves his grandchildren
- Sometimes feels alone, since his wife died

### Goals:

- Would like to run in a short marathon
- See his grandchildren succeed in life
- He wants to win a gardening contest

### Use Case Scenarios:

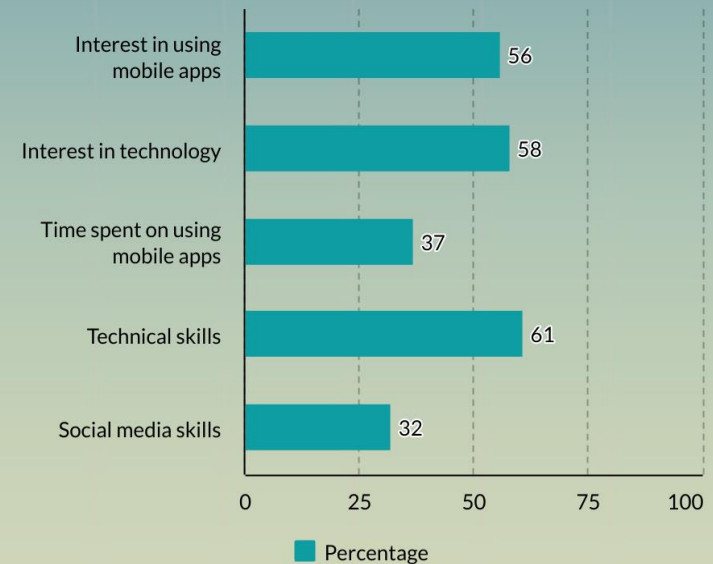
#### Borrow something:

Twice a year he needs some expensive gardening equipment. Therefore it would be great, if he could borrow these instead of buying them.

#### Gift something:

He wants to gift the leftover vegetables and herbs from his garden.

### User Behaviours





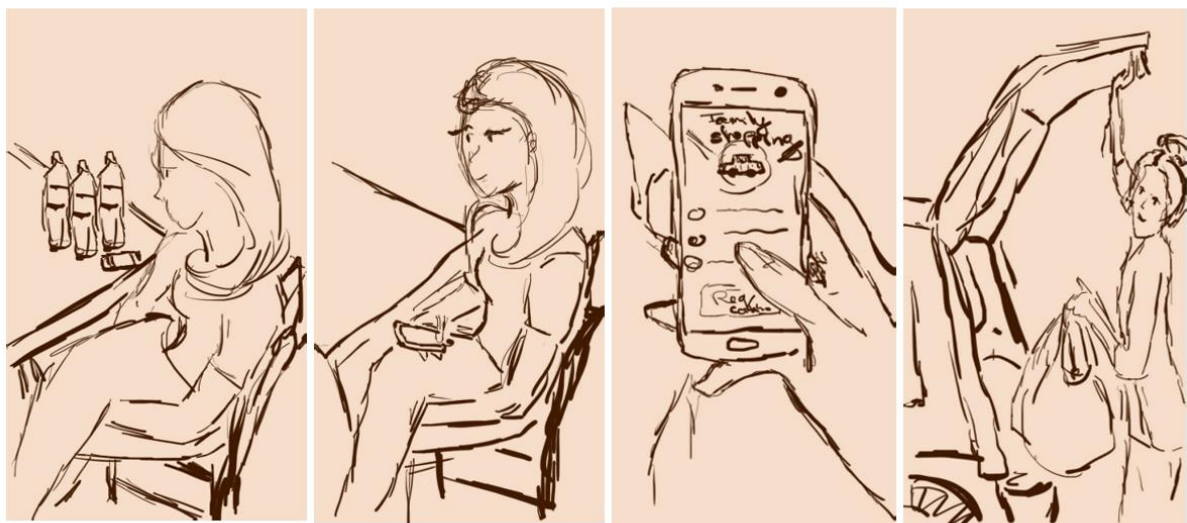
## 2.2 Storyboards

We created two storyboards. Figure 1 shows Thomas, when he realises that he planted too many carrots. He lives by himself and fears to be forced throwing them away (picture 1). In the second picture he thinks about what he could do in order to solve this problem. With the help of our App he could easily take a picture of the food and create a gift advertisement (picture 3 and 4).



*Figure 1: Thomas planted too many carrots*

Amy Sue is depicted below on figure 2. In the first picture she looks sad, because she has no more bottles of juice left. Then she thinks about possible solutions (picture 2). She comes up with the idea that she can use our app and creates an advertisement, where she asks for help (picture 3). In the last picture we see her happy, because someone helped and took her to the supermarket.



*Figure 2: Amy Sue has no more bottles of juice left*

## 2.3 Requirement specification

This section contains the functional and non-functional requirements.

In order to reach an easier navigation and separation, we divided the advertisements into two main und three subcategories.

There are two main categories:

- "Search for help"
- "Offer help"

In both main categories the user can choose between three subcategories:

- "Give something for free"
- "Borrow/Lend something"
- "Offer help" (e.g. doing the shopping, work in the garden)

### 2.3.1 Functional requirements

In the following we list and explain our functional requirements. The level of importance is stated via a scale from zero to five:

1 = not that important

2 = nice to have

3 = quite important

4 = important

5 = main or basic functionality.

Requirement	Create an account
Number	1
Precondition	The user has already installed the application and has an Email-address.
Description	The user should create an account with his/her personal information. EditText-> button (-> save in DB)
Success criteria	The user must be able to enter all the necessary information; name, password, Email, phone number, post code. An active connection to the database is needed.
Level of importance	5

Requirement	Log in to own account
Number	2



<b>Precondition</b>	The user has already installed the application and created an account.
<b>Description</b>	The user can log in to the account by providing the username and the password.
<b>Success criteria</b>	The system must verify the username and the password provided by the user by comparing with the data saved about him/her. If everything matches, the user will be logged in to own account by providing own username and password.
<b>Level of importance</b>	5

<b>Requirement</b>	<b>Create an advertisement</b>
<b>Number</b>	3
<b>Precondition</b>	The user has already chosen the main and subcategory.
<b>Description</b>	The user should be able to create an advertisement within both main categories ("Offer" and "Search"). Here the advertiser must be able to specify one of the three subcategories ("Give something for free", "Borrow/Lend something", "Offer help"). The advertisement should contain additional information about the date, the way of shipping, post code and a short description.
<b>Success criteria</b>	The user must be able to enter all the necessary information: post code, date, way of shipping, description. All the provided information will be saved in the database. Other users should be able to see the ad, when searching.
<b>Level of importance</b>	5

<b>Requirement</b>	<b>Search for an advertisement</b>
<b>Number</b>	4
<b>Precondition</b>	The user has already chosen the main and subcategory.

<b>Description</b>	The user should be able to search for an advertisement within both main categories ("Offer help" and "Search for help"). Here the advertiser must be able to search within the three subcategories ("Give something for free", "Borrow/Lend something", "Offer help").
<b>Success criteria</b>	The system will only show advertisements, which fit to the searching criteria. It will show the most important information of the advertisements: name of the item and name of the city.
<b>Level of importance</b>	5

<b>Requirement</b>	<b>Pick one advertisement from the results</b>
<b>Number</b>	5
<b>Precondition</b>	The user has searched for an advertisement and can see the result list.
<b>Description</b>	The user can pick one specific advertisement from the list of results in order to see all the other information. Therefore, the user needs to click on the container for the chosen advertisement.
<b>Success criteria</b>	The system will show all the available information from the offering.
<b>Level of importance</b>	4

<b>Requirement</b>	<b>Request personal information</b>
<b>Number</b>	6
<b>Precondition</b>	The user has already selected one offering and is on the screen, where all information is shown about this item.
<b>Description</b>	When interested in interacting with one advertiser, the user should be able to request the contact information of the advertiser. In order to do so, the user can find a button at the bottom of the screen.

<b>Success criteria</b>	The system will only show advertisements, which fit to the searching criteria.
<b>Level of importance</b>	4

Requirement	Account management/ Settings
<b>Number</b>	7
<b>Precondition</b>	The user is logged in.
<b>Description</b>	The user can choose to edit his/her personal information and photo by clicking on the "Profile" symbol. The symbol will be in a menu at the bottom of every screen.
<b>Success criteria</b>	The user can now choose to edit his profile information, which includes; photo, personal information (and his/her ratings).
<b>Level of importance</b>	3

Requirement	Edit personal information
<b>Number</b>	8
<b>Precondition</b>	Setting Menu -> Personal Information
<b>Description</b>	The user can edit his/her personal information by clicking on the "Edit" symbol.
<b>Success criteria</b>	The system will show all the saved information of the user and the user can now choose to edit his profile information, which includes; name, post code, address, Email.
<b>Level of importance</b>	3

Requirement	Advertisement management "Offer"
<b>Number</b>	9

<b>Precondition</b>	The symbol in the menu bar at the bottom was clicked.
<b>Description</b>	The system will show all the advertisements the user has created.
<b>Success criteria</b>	The user will get a list, which includes all the advertisements the she/he has created and can edit or delete the advertisements.
<b>Level of importance</b>	3

<b>Requirement</b>	<b>Advertisement management “Borrow”</b>
<b>Number</b>	10
<b>Precondition</b>	The symbol in the menu bar at the bottom was clicked.
<b>Description</b>	The system will show all the advertisements the user has recruited.
<b>Success criteria</b>	The user will see a list with all the recruited advertisements. By clicking on a list item more information will be shown.
<b>Level of importance</b>	3

<b>Requirement</b>	<b>Rate other users</b>
<b>Number</b>	11
<b>Precondition</b>	The user must already have requested personal information from the other one.
<b>Description</b>	The user can rate the other by selected one to five stars and can create a comment.
<b>Success criteria</b>	The rating will be saved and shown under the user’s profile.
<b>Level of importance</b>	2

<b>Requirement</b>	<b>Sort the result list</b>
<b>Number</b>	12

<b>Precondition</b>	The result list is open.
<b>Description</b>	When searching for advertisements, the user will be able to sort the list of results by clicking the sort button. Sorting criteria: distance and user rating.
<b>Success criteria</b>	The system will sort the list according to the chosen criteria.
<b>Level of importance</b>	1

### 2.3.2 Non-functional requirements

In the table below we specify our non-functional requirements.

<b>Non-functional Requirement</b>	<b>Description</b>
Performance	It shouldn't take more than 3 seconds to load a screen. Furthermore, we want to reduce network usage and computing to save battery-life.
Portability	The application is supposed to run on different Android systems.
Screen Adaption	Devices have different screen sizes, so the app should be able to scale according the target's screen size.
Usability	The used symbols should be easy to understand, and the user should learn fast to interact with the system. The used colours are limited to an amount of four, in order to have not too many colours.
Security	We decided to save the data provided by the user encrypted and we only ask for information, which is necessarily needed. We constraint the input fields, e.g. to a certain amount of characters or for the post it is only possible to type in numbers. It means the user must pass the correct data type and the correct format within the correct range to pass the login/register screen.
Reliability	The system should reliable perform the function the user asked for. We want to send a short feedback to the user performing actions, e.g. if you request contact exchange from another user.
Accessibility	All the buttons should not be too small, so that everybody can easily use them. We decided not to use the colours red and green together, so this doesn't confuse our users with red-green-weakness.
Availability	The application should be downloadable in the Google Play Store, where users can easily access and install it.

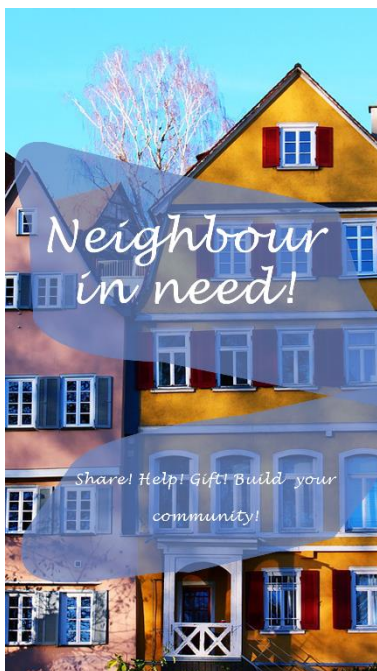
### 2.3.3 Nice-to-have requirements

We also detected requirements, which would be nice to have, but are not necessarily needed for the main functionality of the application. We planned to implement them just in case we would have time left at the end of the semester.

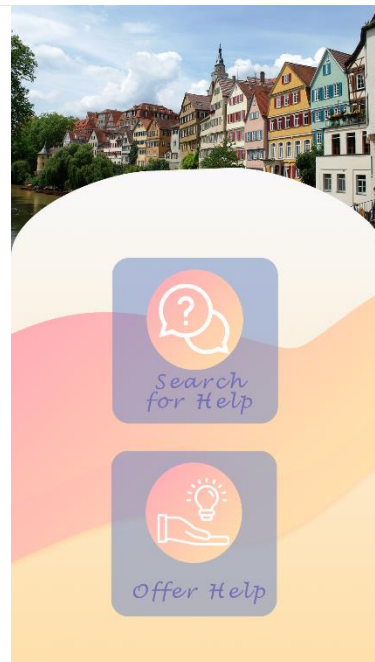
- *Chat function*
- *Rating system*, where users can rate and write comments about other users, after they interacted with each other
- *Geolocation*, in order to verify the address at the very beginning of the registration and for the search function, and also *Maps integration* for the search function
- All the *legally* basics

## 3 Mock-Ups

Here we show our Mock-Ups. They were created in Photoshop and we tried to make the design of the App like them.

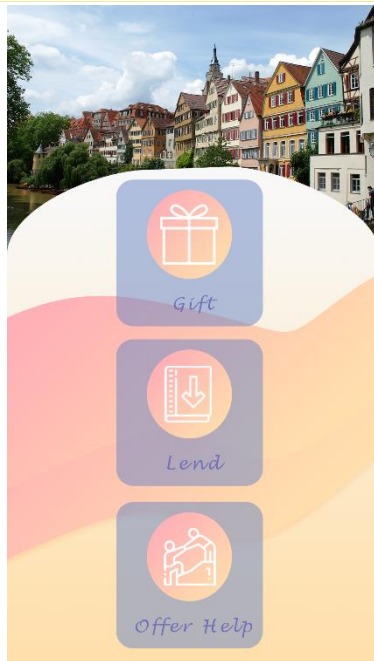


*Start Screen*



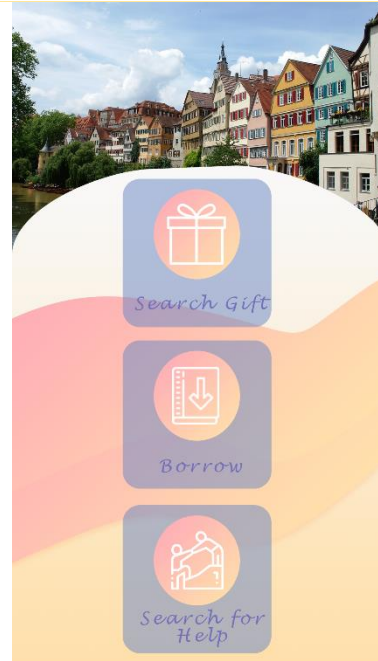
*Choose between the two main categories*





Choose between three sub options

Offer



Choose between three sub options

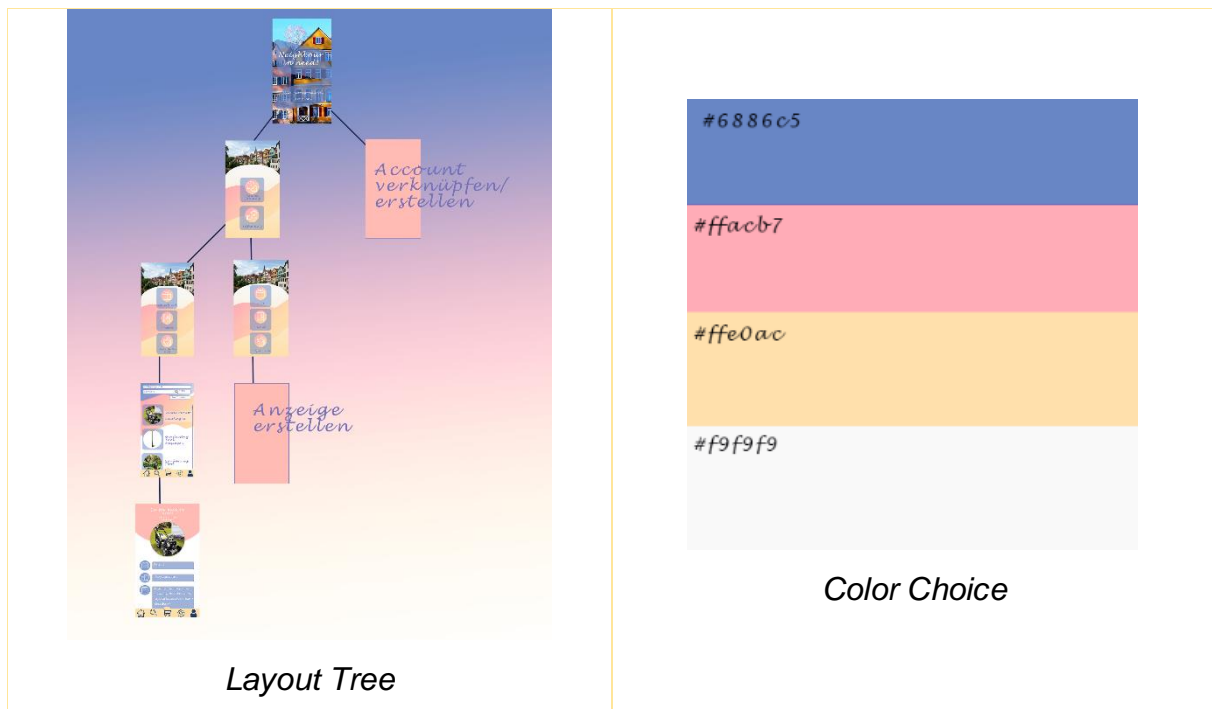
Search



Show search results



Show the details of the picked item



## 4 Some design decisions about the implementation

In this chapter we show some of our implementation designs and decisions. Of course, we don't explain every single feature, but decided to show the more complex ones or the ones we had to think about a little bit longer.

### 4.1 Firebase

As we implemented a multi-user application, we needed a remote database to store the data and synchronize across all clients. For our database connection, we used Google Firebase Realtime Database, because our data structure is flat. Google offers Firebase Realtime Database, which is a cloud-hosted real time database. Data is stored as JSON representation.<sup>1</sup> In figure 3 and 4 below you can see the attributes of the user and the advertisements.

Test User 1

```

{
  "city": "Reutlingen",
  "description": "default",
  "email": "ebru-selin@web.de",
  "image": "https://firebasestorage.googleapis.com/v0/b/nei...",
  "name": "Test User 1",
  "password": "diesonnescheint",
  "postcode": "72766"
}

```

Figure 3 User node in the database

searching for flowers

```

{
  "city": "München",
  "date": "15.07.2020",
  "days": "five",
  "description": "I am searching for real or deco flowers and pla...",
  "image": "https://firebasestorage.googleapis.com/v0/b/nei...",
  "mainCategory": "Search",
  "name": "searching for flowers",
  "shipping": "Will fetch them.",
  "subCategory": "Gift",
  "user": "Test User 1"
}

```

Figure 4 Advertisement node in the database

<sup>1</sup> <https://firebase.google.com/docs/database>

## 4.2 How to save the current user

After successful login an instance of the user will be saved in `Prevalent.currentUser`. This can be always be accessed from the other Activities of the Application. We build in a Remember me! Checkbox so that we can access the app again and again without the need to login every time. Whenever a user has entered username and password and the checkbox is clicked, the app will remember that user and it will automatically login in the MainActivity. We used the Paper<sup>2</sup> dependency to realize this feature. The dependency writes the user information to the Android memory so it will remember it the next time we use the App. First, we initialized the Paper library and then we could use the read and write methods to save and call the user data stored in the Android memory.

## 4.3 Main, Register and Login Activities

In the Main activity, the user can login if he has already created an account or click on the register button and create a new one. If he clicked the Remember me Checkbox the user will automatically enter the app. The Login Activity will access the user data in the Firabase-database and if he user data is correct it will save the user keys in Paper, than get the user to the next Activity. The Register Activity will check the data in the text views and it will generate one new user instance in the Database if the data was correct.

## 4.4 Upload Ad-Image and Userimage

Our images will be stored in Firebase store and the app can access these images with the help of the Firebase database. In the database we are uploading a Uri of the Firabase storage where the images are really stored and we are accessing them with this Url.

## 4.5 Bottom Menu

The BottomNavigationView is the main navigation, which the user can use to switch between Activities. Android already has an xml template for menu with items. In the item, we can save the icon and a title. This menu will be used within our NavigationView. The Navigation view enables the switch between items by ItemID.

## 4.6 RecyclerView

For displaying search results of our advertisements (see figure 5), we used a RecyclerView in combination with an Adapter class. We also used it in order to display the user's own advertisements. It is recommended to use RecyclerView, when the data, which should be displayed, could change at runtime. The Adapter manages the collection of data and binds it to the view.

We also implemented a ClickListener for all Views in the RecyclerView. Another activity, showing all information about the advertisement will be displayed, when an

---

<sup>2</sup> <https://github.com/pilgr/Paper>

advertisement is clicked.<sup>3</sup> The user can filter the search results by clicking on the items in the drop-down menu (shown in figure 6 and 7). The search function helps users to search for their city.

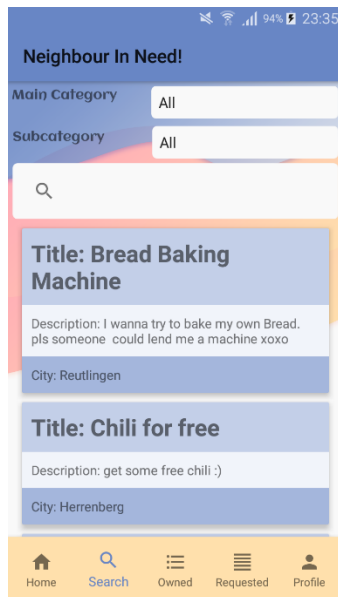


Figure 5 Search function

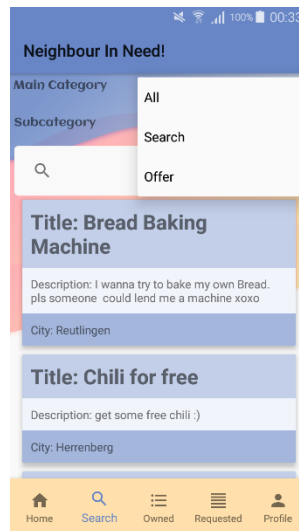


Figure 6 Filter search results by main category

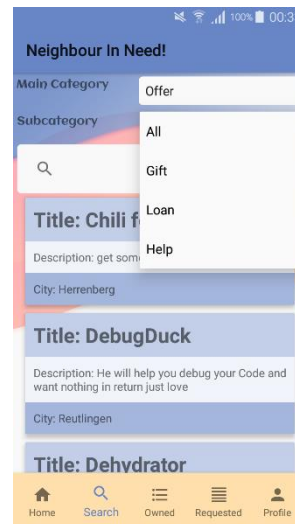


Figure 7 Filter search results by subcategory

## 4.7 Create Advertisement Activities

All Activities that start with CreateAd are somehow related to process of creating an Advertisement. In the CreateAdSubcategory the user can choose between Search and Offer. After clicking one of the buttons the user can choose between the three subcategories. If the user has successfully chosen a subcategory, he can create his advertisement. The user needs to fill all the fields and add an image to the ad.

## 4.8 Sending an Email to another user

When a user is interested in an advertisement, he can contact the advertiser via Email. At first, we didn't know, how to implement this requirement, because we read about security risks, when starting a new intent for fulfilling the task.<sup>4</sup> After consultation with the lecturer, we decided to start a new intent anyway, because it is just a prototype.

## 5 Current Design

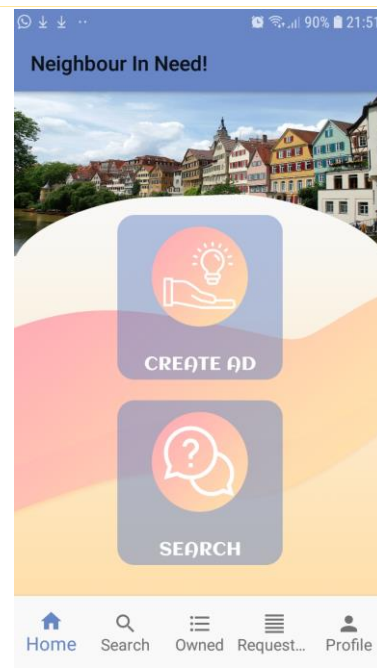
In this chapter the current design of our App is shown.

<sup>3</sup> <https://guides.codepath.com/android/using-the-recyclerview>

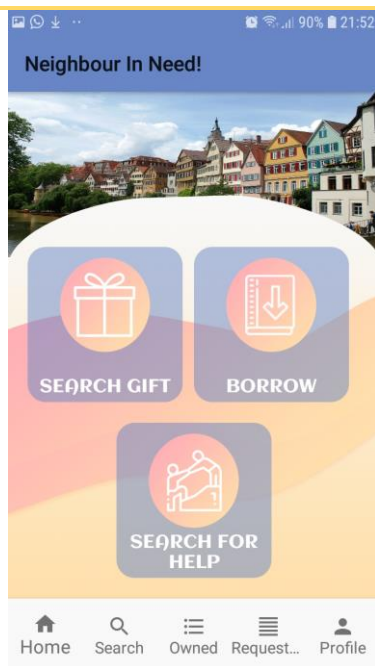
<sup>4</sup> <https://developer.android.com/training/articles/security-tips>



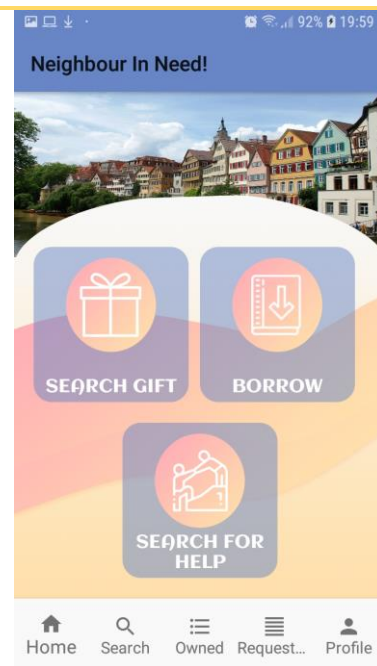
*Start Screen*



*Choose*



*Choose between three sub options*  
*Offer*



*Choose between three sub options*  
*Search*



## 6 Results

This chapter contains our time schedule, the project progress and the results including the testing procedure and the current status of the project.



## 6.1 Time schedule and project progress

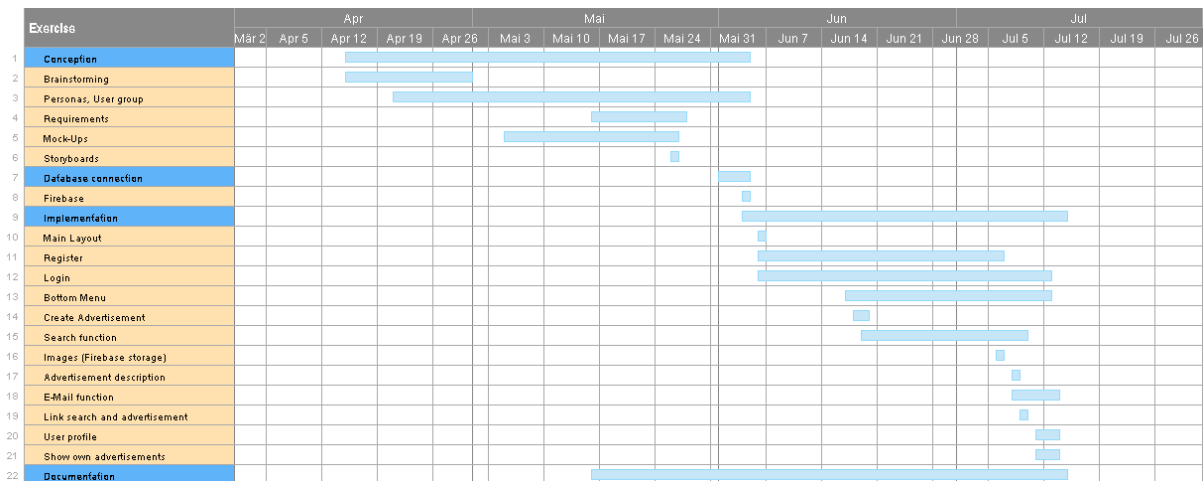


Figure 8 Time schedule

We didn't set a schedule at the beginning but prioritized the requirements and began with the highest ones. We developed the layout and its functionality at the same time. In figure 8 you can see a time schedule, which we made at the end of the project, in order to see which functionality took how much time.

## 6.2 Testing and evaluation

We decided not to write any explicit tests but tested every feature via the application UI, by checking the values in the database or via logs in the code.

We also asked our friends to test the application and give feedback by clicking through.

## 6.3 Degree of completion of our requirements

Functional Requirement	Level of importance
Create an account	5
Login	5
Create an advertisement	5
Search for an advertisement	5
Pick one advertisement from the list of results	4
Request personal information	4
Account management/settings	3

Edit personal information	3	1 = not that important 2 = nice to have 3 = quite important 4 = important 5 = main or basic functionality
Advertisement management „Offer“	3	
Advertisement management „Borrow“	3	
Rate other users	2	
Sort result list	1	

In the table above we list our functional requirements. We managed implementing the ones without a box around. We didn't implement the ones with the red boxes around. We have implemented those with the orange box around to a lesser extent. The plan was that the user can edit personal information and own advertisements. In the current status, the user can see them but not edit them.

It was hard to test some of our non-functional requirements, especially the performance and availability. We tested the App on four physical smartphones, in order to see the portability and screen adaption. Concerning the usability, we limited the colours, avoided red and green and used self-explaining icons in the UI. We reused the UI elements on different layouts, tried not to make the buttons too small and to make the interaction easily understandable. We tested it with the test persons. Moreover, we constrained the input fields, by checking the number of letters and for the post code we only allow numbers. This point could be improved by doing it more granular, establishing more restrictions for the password and encrypting it. Furthermore, we give feedback to the user by showing Toasts and loading bars.

We didn't implement the nice-to-have features, but it was not the intended.

## 7 Feedback

„Finde die App optisch sehr schön und die Idee an sich sehr toll! Es sollte mehr Menschen geben die anderen helfen und so eine Plattform bietet einem eine erste Anlaufstelle, falls man etwas sucht. 😊 Account erstellen war einfach. Alles sehr gut lesbar finde ich.“

Versuche gerade eine Ad zu erstellen, ist ein wenig komisch wenn man etwas sucht ein Bild einfügen muss:) auch das alles mehr als 8 Buchstaben enthalten muss ist grad komisch weil das soll ich bei Shipping reinschreiben?

Verbesserungsvorschlag:

- Vielleicht das es nicht immer 8 Zeichen sein müssen, sondern das die Requirement ist das überall etwas drinsteht?

Ansonsten ist die App leicht zu bedienen und sehr übersichtlich 😊“

„Sieht doch schick aus :D“

„Sieht nice aus“

„Suchfunktion ist nicht so gut, wenn ich dog eingebe findet es nix, obwohl die heute ad die angezeigt wird ohne suche dogsitting ist... Es sagt immer erst no advertisements found, aber wenn ich es abschicke geht's dann doch.“

„Alles in allem gefällt's mir sehr gut. Funktioniert soweit alles und das Design ist schlüssig“

„Bei der Registration Screen ist zu klein, man kann nicht auf dem Button klicken.

Gerät: Android 8 huawei p10 lite ich glaub 5zoll. “

„Crash beim starten. Nach einem crash hat er versucht mich wieder einzuloggen, aber nicht geschafft. Man muss es dann nochmal beenden. Bei mir sieht man den Button kaum 😊 beim eintippen muss man auch immer wieder auf zurück um auf die anderen Felder zu gelangen. Das nervt bissle. Design cool. Bei search und offer bekomme ich beides mal ein Absturz 😞 sieht aber alles du aus. Was ich mich noch frage: Wieso kann man nachdem man create ad anklickt gleich nochmal suchen? Und ich weiß ohne Kontext nicht was eine ad ist. Design gefällt mir aber sehr gut :) „

„Nach Bugs hab ich jetzt nicht speziell geschaut. Ich könnte maximal versuchen irgendwelchen Müll in die Felder einzutragen. Also ob die App mit Buchstaben, Zahlen, Umlauten, Leerzeichen etc klar kommt. Ich bin eigentlich mit allem klar gekommen. Dass man mindestens 8 Zeichen braucht ist beim Testen ein bisschen umständlich aber im realen Gebrauch wird das vermutlich kein Problem sein. Ich bin mir tatsächlich nicht sicher ob man unbedingt ein Bild für jede Anfrage braucht.“

„Das Profilbild wird verzerrt dargestellt und nicht automatisch zugeschnitten. Bei nicht korrekten Eingaben wärs nice wenn die Felder rot umrahmt hervorgehoben werden. Und z.B. bei Eingabe der Postleitzahl automatisch das passende Dorf eingefügt wird. Lauter Luxusfunktionen für uns mittlerweile verwöhnten Usern. Nach Login geht es nicht mehr weiter. Hab zurück gedrückt und jetzt bin ich wieder im Loginmenü. Passt also. Vielleicht ein Timeout programmieren der einen wieder zurück bringt. Hab mich eingeloggt und herumgedrückt. Gefällt mir insgesamt als Basis. Nur diese Schriftart mag ich nicht.

Verbesserungsvorschlag -> benutzerdefinierte Design - > Fonts etc. “

„Also beim ersten auswahlscreen ist ja create ad und search. Und wenn man auf create ad klickt kommt nochmal search. Das hat mich etwas verwirrt. Konnte gerade nochmal bisschen rumklicken und es hat alles gut funktioniert“

## 8 Conclusion

The focus in this project was on working with Android Studio. The work consisted of two parts. In one part we designed the layouts using the editor or the xml-file. The other part contained implementing the logic of the app, which we did in Java.

Altogether we were very motivated, because we liked the idea of our App a lot. Due to lack of time and conflicts with other projects, we couldn't implement all the

requirements, we specified. In summary we managed to implement most of the requirements, especially the main features.

We liked Android Studio, because it has a quite fast learning curve and we found some helpful tutorials on YouTube. Overall you can find a lot of tutorials and solutions on the Internet.