Second mandatory assignment: Travel card app

Teacher Jørgen Staunstrup

Jacob Avlund

Student : Larus Ingi Larusson

Course : Mobile App Development – 3016555

Date: 26-04-2017

Table of Contents

Introduction

In this project, the Rejsekort functionality is simulated. Rejsekort is a RFID-solution that Movia offers customers to use a ticket while traveling with public transportation. when a customer travels with rejsekort, he is able to scan his card at every metro/S-tog stations or in busses to check in to the system and check out by the same means. When the customer checks-out, the trip price will be calculated and deducted from the customer's account.

In this project, this functionality will be implemented. Instead of using an RFID-card, an android app will be used that's uses IBeacon signals to determine locations. In ITU, there are several WiFi Aps located on each floor that broadcast IBeacon signals. IBeacon signal consist of three main attributes :

1. **UUID**
   UUID stands for " Universal Unique Identifier " and consist of 32 hex digits split into 5 groups. The purpose of this ID is to be able to filter IBeacon signals in the network from other Ibeacons. The Ibeocon UUID used in this project is : E3B54450AB734D7985D6 519EAF0F45D9.
2. **Major**
   Major is a id that can help to distinguish IBeacons with the same UUID from each other. In ITU is is used to identify different floors of the building. In this project it will be used to identify zones like exist in the public transportations system of Copenhagen. In this project there will be 5-zones.
3. **Minor**
   Minor is als an id that distinguish IBeacons with the same UUID from each other. In ITU it is used to identify classrooms, but in this project it is not used.

There was no specific emphasis in this project, however a big amount of the time was spent on the sqlLite database and the Ibeacon readings.

In this report the Android application will be explained and how the IBeacon protocol was used in it. The database will also be explained and finally the result of testing the app.

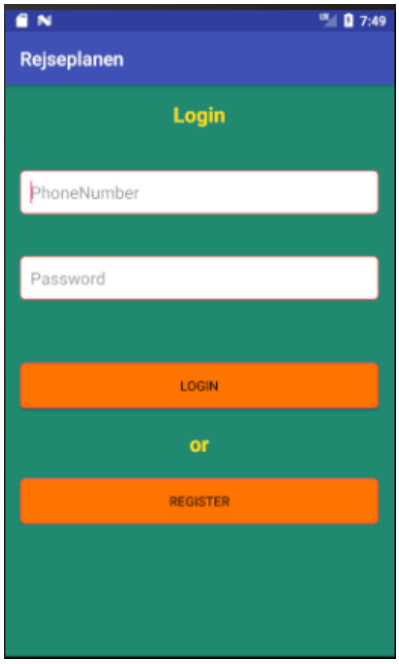
Design choices

This section will cover the most important design choices of this application. The application consists of 3 activity's ,5 fragments and a options menu.

Layout of user interface and how it is implemented in activities/fragments

In this section the 3 activity's ,5 fragments and the options menu will be explained, both the design choice and their functionality.

Rejseplanen - Activity:



Design :

This is the first activity that the user will be presented with. The purpose of this activity is to enable the user to authenticate himself before he is able to start to use the app. Since this app is running in a phone it was decided that the mobile number would be the best identifier since it is easy for customers to remember their phone number but usernames can be difficult to remember.

Functionality:

For the user to be able to authenticate himself, he has to be registered and he must enter his phone number and password.

If the user is not registered he must press the register button and he will be sent to the Register-Activity.

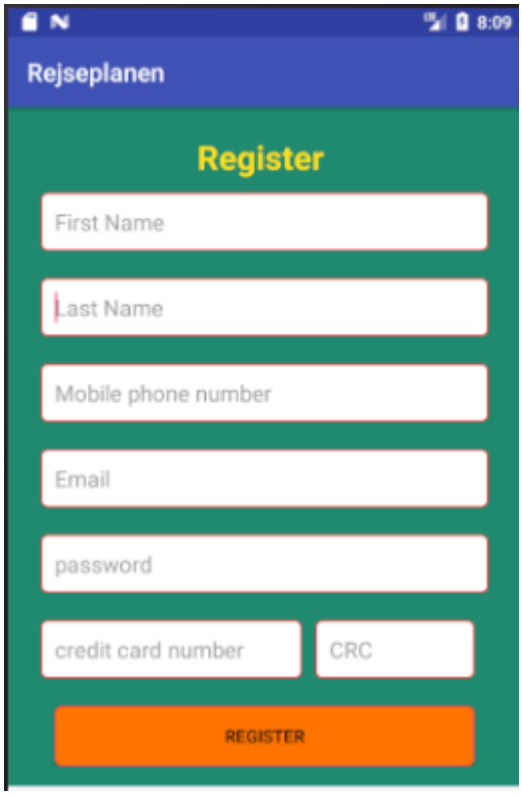
If the user is register, he must enter his phone number and password and press enter. If the login credentials are correct, then there are 2 shared preferences stored:
loggedIn is set to true and **mobile** Is set to the provided phone number. Finally the user will be sent to the Main Menu Activity.

If the user is registered but does not enter correct login credentials, a toast will be prompted with the message :" wrong password or phone number".

Constraints:

There are no constraints in this Activity, the user credentials are verified in database.

Register – Activity



Design:

Originally the user was only supposed to provide his phone number and password, but to make the program more user friendly, he is also required to enter his name. and email in the case of lost password. It was also decided to include the card details so the customer would not have to type it every time he wanted to add money to his account.

Functionality:

The user must fill in all required fields, and press Register. When register is pressed, 2 things can have happened:

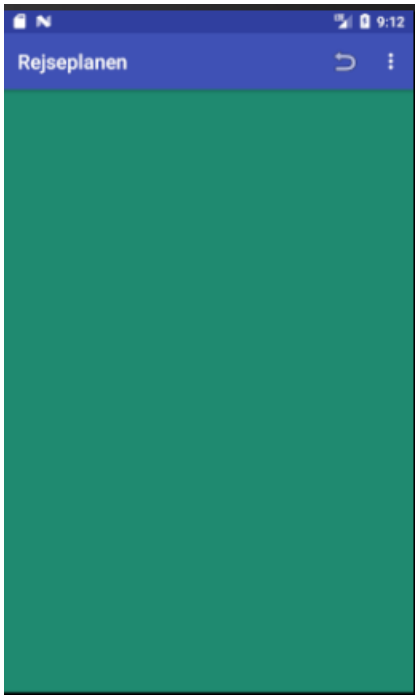
1. A User object is created and populated. Then the system will attempt to store the object in the database. If the object will be successfully stored, the user will be sent to the Main Menu activity and 2 shared preferences are stored: **loggedIn** is set to true and **mobile** Is set to the provided phone number.

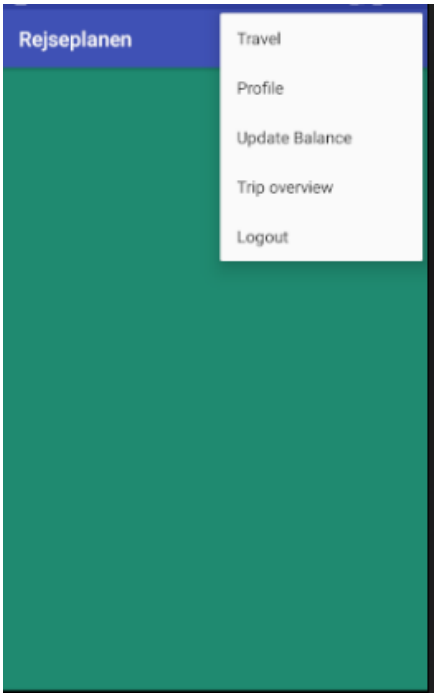
2.if the user enters a mobile number that is already in use the insertion to the database will fail. In the case the insertion fails then the user will be prompted with an Toast with the message: *" number already in use or invalid input "*.

Constraints:

Every field must have 1 or more values.

Main Menu – Activity and main_menu – Menubar





Design

The Main Menu Activity is the first Layout the user will see when he logs in and the Last activity in this applications, other layouts are fragments. For the user to navigate around the app, a menu bar was created that can be used to go to different fragments

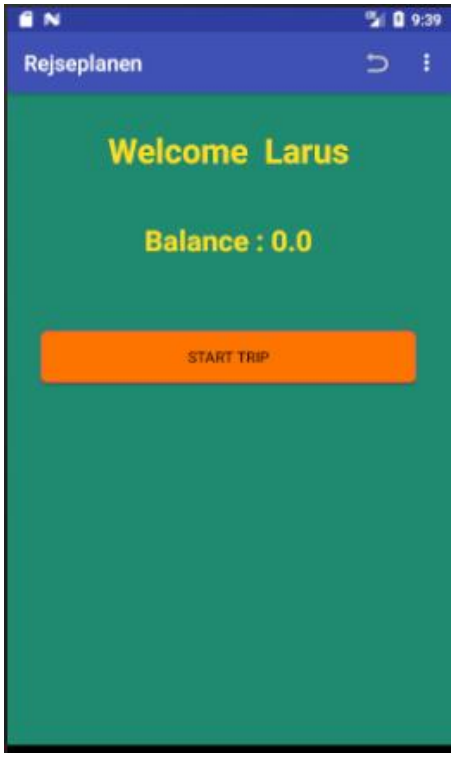
Functionality :

This activity surfs as a placeholder for all the fragments. The user can click the three dots in the corner to get the menu bar.

Constrains:

The user has to be logged in in order to get to this activity.

welcome_fragment – Fragment



Design: This fragment is intended to let the customer know that he is successfully logged in and it should surf as a “home”-layout. In this layout the customer should also be able to start traveling quickly It was also decided to have a label that would tell the customer his balance so he would be able to recognize if his balance was insufficient for traveling.

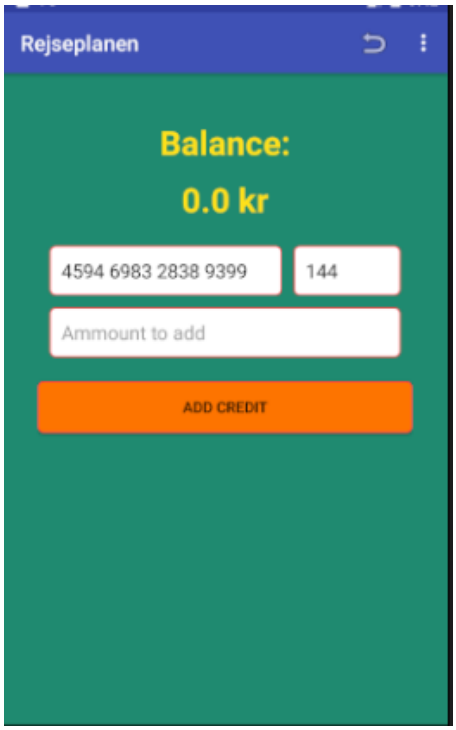
Functionality

In this layout, there are two labels populated that display the name of the customer and tell him what his account balance is. Then there is the “start trip button” that will send the customer to the Travel fragment if clicked.

Constraints:

User must be successfully logged in.

Bank_fragment - Fragment



Design:

The bank_fragment was designed to enable the customer to add credit to his account.

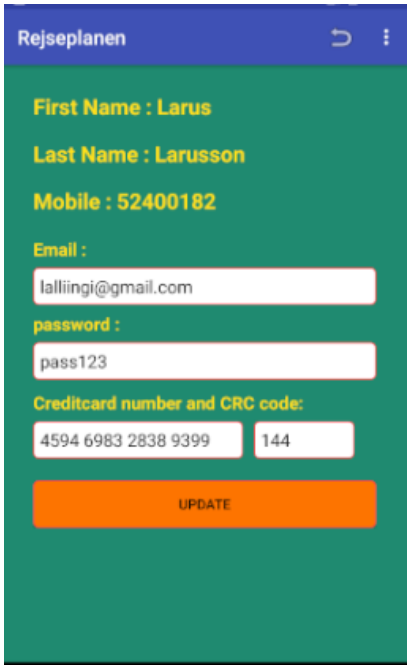
Functionality:

When the customer opens the bank fragment, the credit card field and the crc field will be automatically loaded from the database. If the customer wants to use another credit card, he can type in another credit card number and crc. When the customer has entered the amount and credit card info he can click the “add credit” button. When that button is clicked, 2 things can happen :

1) If the withdraw was successful then the balance label will be updated to the new balance, and the amount to add - editBox will be cleared.
2) If the withdrawal was not successful , then a toast is displayed that has the text : “ invalid card or balance”.

Constraints: The use has to be logged and the amount must be at least 1 digit in order to attempt to withdraw

profile_fragment – Fragment



Design :

The profile fragment was designed to enable user to view and update his information.

Functionality

In this fragment the user can see his data and change it if he wishes. If the customer wishes to change his data, he has to change the values and press the update button. When the update – button is pressed a toast will appear that can have two messages :

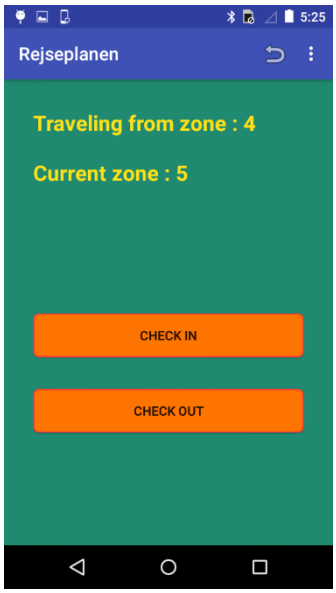
1)
If the update was successful the toast will say : “Profile updated !”

2)
If the update was not successful the toast will say : “Input is not valid !"”

Constrains:

- If the user wishes to change the credit card data, the new creditcard number has to be atleast 13 digits, and the crc must be 3 digits.
- Email and password must be atleast 1 digit
- First name, last name and mobile number cannot be changed

travel_fragment Fragment



Design :

This fragment was designed to enable the customer to check in and out of the system. When the user

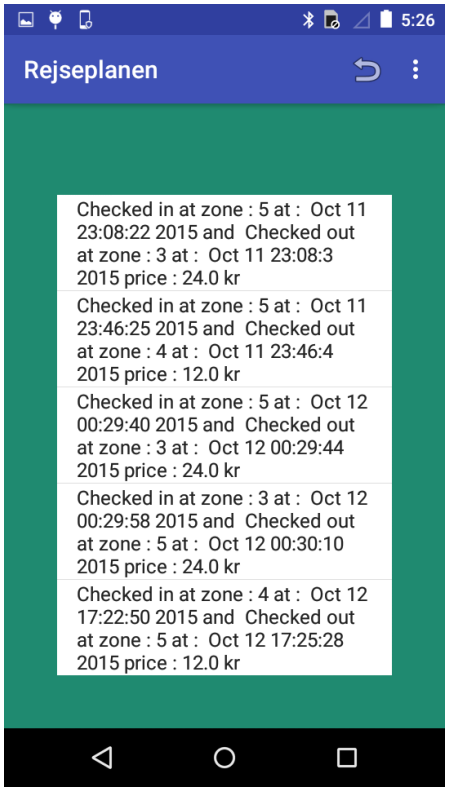
Functionality:

When the user opens this fragment it will check his balance. If the balance is sufficient the current zone label will update to the closest zone “which is the major part of the Ibeacon otherwise a toast will appear that says “balance is to low”. When the user wants to check out, he must press the check out button and then the customer will be sent to the travel_overview fragment for each zone traveled, the customer will pay 12kr.

Constraints:

The use must be logged in and have more than 60kr on his account.
To check out, the user must have traveled at least 1 zone.

trips_overview Fragment



Design:

This fragment was designed to enable the user to see his past travels.

Functionality :

When the user opens this fragment he will be displayed with a listview that shows him his past travels if he has traveled before. If he has not traveled before, a label will be shown that says: "You have not made any trips" . Th list constists of rows that display the following attributes:

Checked in at zone at: shows the zone where the user checked in and what time he did that .

Checked out at zone at: shows which zone the user checked out at and when he did that.

Constraints :

The use must be logged in and have made at least 1 trip.

Database:

In this project there were 2 types of data storage used: SharedPreferences and an SQLite database. Shared preferences are used as global variables so that current users mobile number is always assessable and then there was another Boolean called isLoggedin used so it can be checked if the user is logged in.

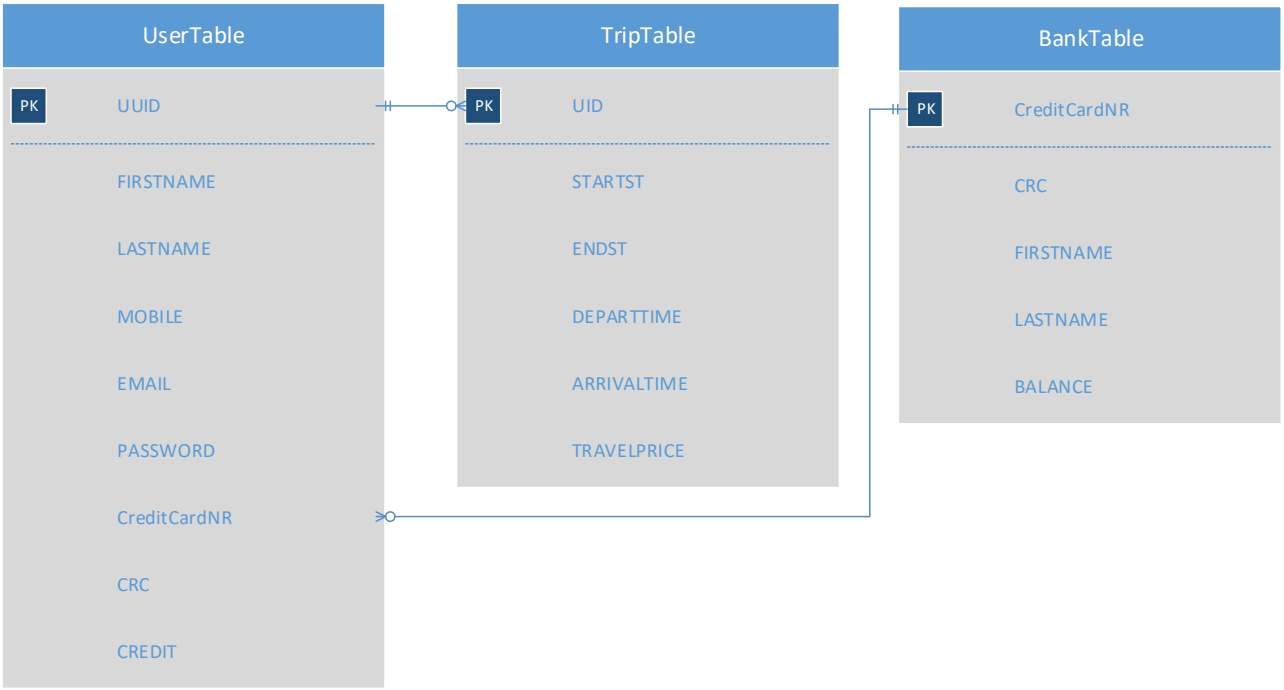
the database userbase.db was created. Userbase consists of 3 tables :

UserTable: Registers new customer and keeps data about them.

BankTable: Simulates a Bank so users can withdraw money from their credit cards and deposit on their user account in the user table.

Trips: Trips table is used to log customer travels.

Below is a diagram of the database :



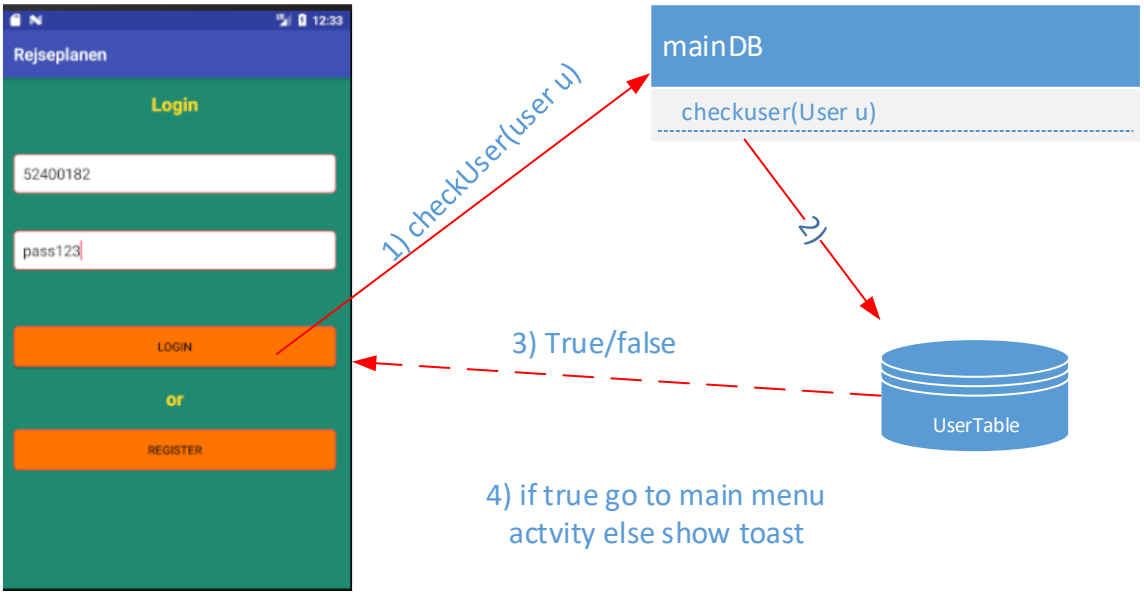
Key functions :

In this project there are 5 key functions :

1) Log in :

The log in function is responsible for validating data provided by the user in order to log in to the application.

Sequence of logging in :

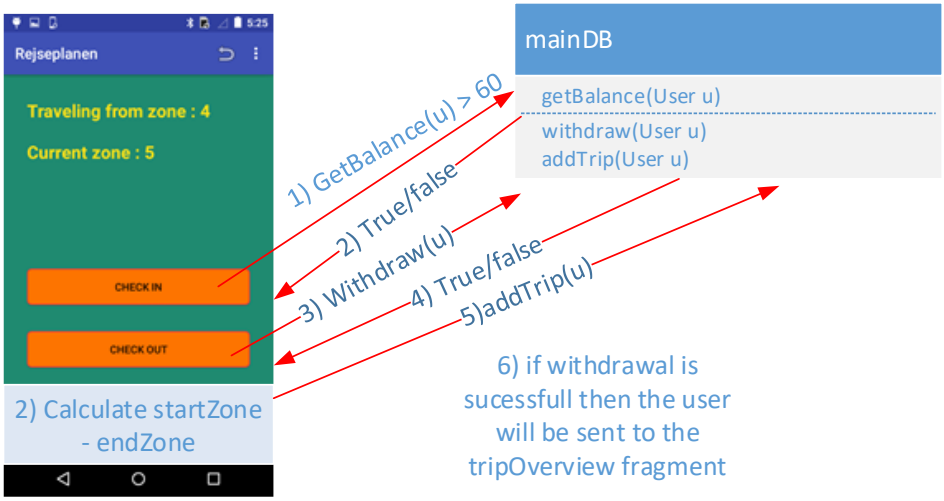


2) Register user

Is responsible for registering users to the UserTable database. Sequence of logging in :

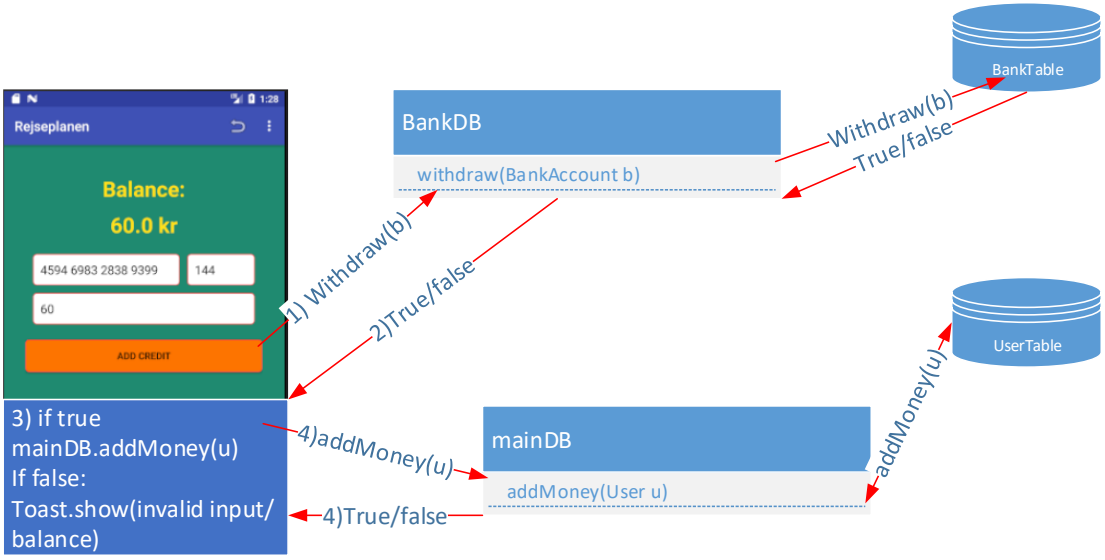
3) Make a trip

Is Responsible for letting the user check in and out of a trip. When the user will press the register button the system will first check if the user has sufficient balance, then when the user will press check in the check in label will update. When the user presses check out, the trip cost will be calculated and deducted from the user account. Finally, the user will be sent to the trip overview fragment if the withdrawal is successful. Below is the sequence



4) Add credit to account

Is responsible for adding balance to user accounts. when the user has entered the value he wants to transfer, a BankAccount object will be created and used as a parameter when the bankDB.withdraw() method is called. If the withdraw is sucessful then the new balance will be updated and a toast will notify the user that credit has been added to the user account. Below is the sequence :



5) Get overview of trips

Is responsible for show user his previous trips. If the use has no trips then a label will be displayed that says "You have not made any trips". Below is the sequence :

Testing of the app

Key function being tested	Activity / fragment	Sequence	Result
1)Log in	Rejseplanen_Activity	Enter correct phone number and password and press log in button	User gets sent to main menu activity and is "logged in"
		Enter correct username but wrong password	Toast displays : " wrong password or phone number "
		Enter wrong username but correct password	Toast displays : " wrong password or phone number "
2)Register user	Reg_Activity	Press register button without filling any input	Toast says: Number already in use or invalid input
		Enter number that has already been used.	Toast says: Number already in use or invalid input
		Insert valid data to every field apart from the creditcard and crc	User gets sent to main menu activity and is registered
		Insert valid data to every field apart from the crc field	User gets sent to main menu activity and is registered
3)Make a trip	Travel_Fragment	Press checkin with less than 60kr on account	Toast displays: "Balance to low"
		Press checkin with 60kr or more on account	"start zonelabel is updated to current zone"
		Press checkout when at the same zone as start zone	Nothing appears to happen
		Press check out at another zone	User is sent to Travel_overview fragment
4)Add credit to account	Bank_fragment	Press add credit button without having any value in amount to add field	nothing happens
		Press add credit button with 50 in amount to add button	"Balance label" is updated And credit is added to user account
		Press add credit button with invalid creditcard number	Toast displays : "invalid card or balance"

		Press add credit button with invalid crc number	Toast displays : “invalid card or balance”
5)Get overview of trips	TripsOverviewFragment	Open trips_overview fragment without having any trips	“you have not made any trips” label is shown
		Open trips_overview fragment after making atleast 1 trip	Trip is displayed in a listview

User guide

This user guide is to help users use the app. It includes instruction about how these 5-key function can be performed.

1) Log in:

First user has to be registered before it is possible to log in. To see how to login please refer to key function.2. To log in user has to enter his valid user credentials and press login.

2) Register :

Press the register button on the log in screen to access the register layout. When you are at the register layout fill inn you information and press register. Credit card should be 13 digits minimum and crc code should be exactly 3 digits. Credit card info can also be added in the update balance layout. When register button is clicked you will be sent to the welcome screen if the data is valid. Other vise review the data you inserted and try again.

3) Make a trip

To make a trip press the start trip button on the welcome screen and you will be taken to the travel view. It is also possible to press the menu bar in the right top corner and select the Travel option.

When in the travel layout make sure that you have more than 60kr in your account otherwise you will not be able to make a trip. To add credit to your account, refer to key function 4 – Add credit to account.

Wait until the current zone label is populated before you press the check in button, to check in. after checking in you should see the traveling from zone label being updated to the zone that was in the current zone label when you pressed the check in button.

To check out of the trip, you have to be in another zone and press the checkout button. Now you will be sent to the trip overview layout, if you checked out successfully.

4) Add credit to account

To add credit to your account, click on the menu bar in the top right corner and click on the update balance option.

When you are in the update balance layout, enter your credit card information and the amount you wish to withdraw. When you have entered, valid data press the add credit button and if the data entered is correct then the credit on your account will be updated. If your data is invalid a message will be displayed and you should make sure the entered data is valid.

In this version on the application then you should not that only the credit card information you entered in the registration process will be valid.

5) Get overview of trips

To get overview of trips you should select the menu bar in the top right corner and click on the Trip overview option. Then you will be sent to the trip overview layout.

When you enter the trip overview layout, you should see an overview of your trips unless you have not made any trips yet. In that case, you will see a label that displays the text that you have not made any trips.

Conclusion

In overall I am happy with this application, there was a lot to learn and I belive I managed to learn a lot from building it and doing the class exercises. Like always It would be nice to have more time and add some functionality and fix existing ones.

What could have been done better :

- It would have been nice to make the check in / out process automatic but due to time restrictions it was decied to ues check in and out buttons.
- Reliable check In, like the app is now, use can simply go back to main menu when his trip is finished without paying for his trip. If more time would be available I would add some check in functionality that would register it in database when a user would check in / out and after some specific time limit the user would be checked out and charged for the trip.
- Place database on server
- Make responsive design of layouts

There is a lot of other things that could be added or improved, but for now I am satisfied with how much I learned and look forward for learning more in the future.