DVA313 - Software Requirements Specification

November 2018

Aleksandar acimovic
acimovic.alek@gmail.com

Zacharias Leo
zlo16001@student.mdh.se

Stanislas Pedebearn
spn18013@student.mdh.se

Chingiz Esenbaev
tjv15001@student.mdh.se

Jonathan Major
jmr16009@student.mdh.se

Linus Sens Ingels
lis16001@student.mdh.se

Sireesha Kumari Kulari
ski18005@student.mdh.se

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to provide a detailed overview of an Application called BodaBoda for matching motorcycle taxi drivers and customers. The drivers could state that they are free and customers could use the app to find the closest free driver, with the cheapest price, who is willing to drive them to their destination. The app would tell the driver where to find the customer and it would track the driven distance, calculate a price and handle the payment.

1.2 Definitions, Acronyms and Abbreviation

- **Guest** - An unregistered user of the application.
- **Customer** - The customer that request a trip by a taxi driver using the application.
- **Driver** - The taxi driver that get assigned to a trip by accepting a request using the application.
- **GPS** - Global position system.
- **API** - Application programming interface. A set of routines, protocols, and tools for building software applications
- **ID** - The ID of the functional requirement
- **FR** - Functional requirement
- **DESC** - Description of the functional requirement
- **RAT** - Rational, the reason for the functionality
- **DEP** - Dependency from other requirements
- **NR** - Nonfunctional requirement

1.3 Project Scope

BodaBoda is an Android application with the purpose of connecting motorcycle taxi drivers with customers in East Africa. The customers will get the driver with the cheapest price possible, within a limited radius.

The taxi drivers set their price per kilometer and if they are matched with a trip, they can either choose to accept or decline. Declining a trip will give send it to the next driver in line. The drivers will be able to see information about their earnings.
The software will both need access to Internet and GPS locations in order to function. Most of the information will be stored on a database which is located on a web-server.

1.4 References

2 Overall Description

2.1 Product Perspective

- → The system will consist of an application, a server and a web administration.
- → The application will be able to take advantage of the mobile GPS system and the Google API as well.
- → The application will be able to handle the needs for the customer and the driver separating the functionality by a mode.
- → The GPS system will help the server with functionality like calculating distances between customer and the driver.
- → The Google API will help the server calculating the route distance, helping us estimate the price and show the route on a map for the driver if needed.
- → The auto-fill feature of destinations will also be possible with the Google API for making the customer experience more user friendly.

We also have a lot of data to handle and that requires a database. The database will be used to store account information, transactions and trip information. The web administration is something that will be excluded for now and will be for future development.

2.2 Product Features

In order for the user to use the application they first have to log in to their account. If the user does not yet have an account they can simply create one. When creating the account the user will have an option to enter a credit card number as well as choose their preferred payment method. It is possible to modify the user settings later on.

With a verified payment method, the user is able to request a trip by entering the starting address and destination of the trip. When a driver has accepted the trip the customer will be presented with the information of the ride. This information consists of the price, the name of the driver as well as the distance between the starting location and the driver.

The drivers will use the same application, where they will get to choose whether to be a customer or a driver on log in. This choice is only presented to accounts that are verified drivers. When the drivers first log in they will not be marked as available to be assigned trips. In order to be assigned trips they have to mark themselves as available, which will let the server track their GPS location to show them nearby trip requests. When the drivers get matched with a trip they are shown the information about the trip and a choice to accept or decline it. The information consists of the starting location, destination, distance to the starting point, the name of the customer as well as how much they will get paid. If the driver chooses to decline the assigned trip or does not answer before it times out it will be passed on to the next driver in line.

When a customer and a driver match they will be informed. The driver can choose to see a map of where they will meet the customer. Upon arrival of the driver they initiate the trip in the app, and then marks the trip as finished when they arrive at the destination. If something does not work out the driver can choose to cancel the trip early. Either way the customer has to pay with the app, either paying the full price they agreed to, or a fraction of it if the trip was cancelled early.

2.3 Users Classes and Characteristics

There are three types of users of the BodaBoda application.

Guests are only able to register an account or log in to a existing account.

Customers will be able to request a trip, modify the account or log out.

Drivers will be able to accept or decline requests from customers when in driver mode. Driver accounts will also be able to use the BodaBoda application as a customer by changing the mode to customer.

2.4 Operating Environment

Operating environment for the App called BodaBoda for matching motorcycle taxi drivers and customers is as listed below.

Architecture: Client/Server System

Operating system: Android/Linux

Database: SQL Server

Platform: C# .NET/Java

2.5 Design and Implementation Constraints

There are a couple of challenges when developing this product. Because this application is developed for East Africa it has to be kept in mind that there might be some issues with connection reliability, sometimes packets might have to be resent many times in order to reach the receiver.

Also, the thing to keep in mind is that the target audience often are not very technical and probably has old and outdated mobile phones. Thus, we have to make sure the product has very high usability and compatibility with older systems.

Since the taxi drivers might not have the chance to charge their cell phones during their workday the application has to be optimized for better battery consumption.

2.6 Assumptions and Dependencies

One assumption is that the GPS on all phones work the same way, or at least provide the basic information needed in the same way and format.

Another assumption is that even if there are connectivity issues there will eventually be enough of a connection to send the packets without too much latency.

It is also assumed that the taxi drivers are willing to become technical enough that we can give them more options within the app, unlike the customers who are very limited.

3 System Features

3.1 Functional Requirement

The system should satisfy the following requirements:

3.1.1 Guest Aspect

ID: FR1
TITLE: Account registration - Mobile application
DESC: Given that a user has downloaded the mobile application, then the user should be able to register an account through the mobile application. The user must provide user-name, password, name, phone-number and e-mail address. The user can choose to save credit-card information and choose their preferred payment option.
RAT: In order for a user to register on the mobile application.
DEP: -

ID: FR2
TITLE: User login - Mobile application
DESC: Given that a user has registered an account on the mobile application, then the user should be able to login to an account through the mobile application. The user must provide user-name and password.
RAT: In order for a user to login on the mobile application.
DEP: FR1

3.1.2 Customer Aspect

ID: FR3
TITLE: Modify account information - Mobile application
DESC: The customer should be able to modify any information for the account except their user-name. They can also add their credit card information if they did not do it

at account creation.
RAT: In order for a user to manage their account information.
DEP: FR1, FR2

ID: FR4
TITLE: Request a trip - Mobile application
DESC: The customer should be able to request a trip given that they are logged in. They must provide starting location and destination. The customer must also have a verified payment option.
RAT: In order for a user to request for a trip.
DEP: -

ID: FR5
TITLE: Payment - Mobile application
DESC: The customer should be able to pay for a trip given that they have requested a trip. They must have provided valid payment options to their account.
RAT: In order for a user to make a payment after a trip.
DEP: FR4

ID: FR6
TITLE: Canceling a request for a trip - Mobile application
DESC: The customer should be able to cancel a requested trip given that they have requested a trip.
RAT: In order for a user to be able to canceling a requested trip.
DEP: FR4

ID: FR7
TITLE: See payment verification.
DESC: Given that the user is logged in as a customer. The customer should be shown a payment verification after the customer has paid for the ride.
RAT: In order for the customer to verify that the ride was successfully paid for.
DEP: FR13
ID: F14
TITLE: Search for specific vehicle.
DESC: User will be able to do a custom search for vehicles.
RAT: In order for customer to customize their taxi experience.
DEP: -
ID: F15
TITLE: Report the driver.
DESC: User will be able to report a driver for unethical behaviour.
RAT: In order for prevent bad services.

DEP: -
ID: F16
TITLE: Become a driver.
DESC: User will be able to simply register as a driver with the exsiting account.
RAT: Everyone can be a driver.
DEP: -

3.1.3 Driver Aspect

ID: FR8
TITLE: Change the availability status.
DESC: Given that the user is logged in as a driver. The driver should be able to set their status to available or unavailable to receiving pending trip requests.
RAT: In order for the driver to decide when he wants to get trip requests or not.
DEP: -

ID: FR9
TITLE: Accept or decline ride requests.
DESC: Given that the user is logged in as a driver and marked as available. The driver should be able to choose if he wants to accept a ride request or not.
RAT: In order for the driver to have the option to take a ride request.
DEP: FR8

ID: FR10
TITLE: Find the trip starting point.
DESC: Given that the user is logged in as a driver and have accepted a ride. The driver should be able to get help navigating to the starting point of the trip if wanted. For instance by a map.
RAT: In order for the driver to always be able to find their customers.
DEP: FR9

ID: FR11
TITLE: Set the price.
DESC: Given that the user is logged in as a driver. The driver should be able to set their cost per kilometer.
RAT: In order for the drivers to be able choose the price of their trips.
DEP: -

ID: FR12
TITLE: Initiate the ride.
DESC: Given that the user is logged in as a driver and has accepted a ride. The driver should be able to initiate the ride in the application when he picks up the customer.
RAT: In order for the ride to be initiated.
DEP: FR9, FR10

ID: FR13
TITLE: End the ride.
DESC: Given that the user is logged in as a driver and a ride is in progress. The driver should be able to finish the trip when the destination is reached or cancel it if wanted.
RAT: In order for the ride to be finished to let the customer pay and the driver to be marked as available again.
DEP: FR12

ID: FR14
TITLE: See payment verification.
DESC: Given that the user is logged in as a driver. The driver should be shown a payment verification after the customer has paid for the ride.
RAT: In order for the driver to verify that he has been paid.
DEP: FR13

ID: FR15
TITLE: Track the earnings.
DESC: Given that the user is logged in as a driver. The driver should be able to see statistics of the past earnings.
RAT: In order for the driver to get an overview of their earnings.
DEP: -

4 External Interfaces Requirement

4.1 User Interfaces

The first time a user enters the application they are presented with a start screen consisting of an option to log in, or to register their user account if they are a first-time user.

4.1.1 Customer Interface

When the user has logged in they get the options to either request a trip, modify their account or log out from the account. Modifying the user account lets the user change their email-address, phone-number, password, edit or add their credit card information as well as setting their preferred payment method.

When requesting a trip the user simply enters the starting location of the trip, and the destination and taps the search button. If the user has no credit card information saved, or no preferred payment information they are asked to enter them here. While searching for a driver the user is presented with a "Searching for driver" indication. The user can choose to cancel the search if they wish. If there are no drivers available the user is told by a displayed message. If matched with a driver the user gets a trip suggestion containing the name of the driver, the distance to the driver from the starting point, as well as the price for the trip. The user can either accept the trip, or decide to decline it.

When the user is picked up by the driver and the ride is initiated, the application tells the user about this through a displayed message. As soon as the ride is ended the user is notified about this, and gets a button to press in order to pay for the ride. After the transaction is completed a notification is shown and the user is returned to the front page of the application.

4.1.2 Driver Interface

When a user who is verified as a driver logs in to their account they can either choose to log in as a driver or as a customer. The following text assumes they are logged in as a driver.

Upon log in the driver is not available to receive trip requests. The driver is presented with the options to modify his account, set his availability status to on or off, as well as log out from the account. The options when modifying the account is to change their email-address, phone-number, password, enter the credit card information so they can be paid, as well as setting their price per driven kilometer.

When the driver has set their availability status to on, they are eligible to receive trip requests sent by the server. If a trip request is received the driver is presented with information of the trip, consisting of the name of the customer, the starting location, the destination, the price of the trip as well as the distance to the starting location. The driver can choose to accept the ride, or decline it. If the driver accepts the trip, their availability status is set to off and they are presented with a screen with the trip information mentioned earlier. If the driver has trouble finding the starting they can choose to be shown a map that helps them navigate.

When the driver arrives at the customer they can initiate the trip by tapping a button. When a trip is ongoing the driver can choose to be shown a map that helps him navigate to the destination. The driver either ends the trip when the destination is reached, or has the option to cancel the trip early. When the customer has paid the driver is shown a transaction verification and the driver is automatically available to receive more trip requests.

4.2 Hardware Interfaces

The mobile application does not use any designated hardware and do not have any direct hardware interface. The GPS is managed by the GPS application in the mobile phones.

4.3 Software Interfaces

The GPS application is used by the mobile phones to gather geographical information for locating the customer and driver. With the help of Google API we can calculate route and distance for the customer and the driver. These will be shown in a visual representation and the data will be stored in the database. The calculation of the data will occur in the server. The mobile application will only be able to run read operations from the database but the server will be able to run read and write operations.

4.4 Communications Interfaces

It is important that the communication between the application and the server works as it depends on each other. The server are also very dependent on the database storage for this application to work as intended.

5 Nonfunctional Requirements

5.1 Performance Requirements

Some Performance requirements identified is listed below:

ID: NR1
TITLE: Support of multiple users at any given time.
DESC: Given that the users is logged in as a customer or a driver. The application should be able to handle every user at any given time.
RAT: In order for the application to handle every user at any given time.
DEP: -

ID: NR2
TITLE: Usage of the application should be available 24/7.
DESC: Given that the application is added to the market. The application should be able to serve every user at any given time.
RAT: In order for the application to serve every user at any given time.
DEP: -

5.2 Security Requirement

Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below. Specific requirements in this area could include the need to:

ID: NR3
TITLE: Encrypt the user passwords.
DESC: Given that the user has registered an account. The application should be able to protect the user by encrypting their password.
RAT: In order for the application to protect the user by encrypting their password.
DEP: -

ID: NR4
TITLE: Encrypt the user payment information.
DESC: Given that the user has registered. The application should be able to protect

the user by encrypting their payment information.
RAT: In order for the application to protect the user with encrypting their payment information.
DEP: -

ID: NR5
TITLE: Verifying user permission.
DESC: Given that the user has registered. The server should be able to verify if the user has the permission for a action.
RAT: In order for the server to verify if the user have permission to do a requested task.
DEP: -

ID: NR6
TITLE: Verifying user login.
DESC: Given that the user has registered. The server should be able to verify the validity of the user login information.
RAT: In order for the server to verify the validity of the user login information.
DEP: -

5.3 Software Quality Attributes

ID: NR7
TITLE: Application availability
DESC: The application should be available as often as possible, making sure to resend packets that were not able to be sent due to bad connection.
RAT: In order for the user to always have access to the functions of the application.
DEP: -

ID: NR8
TITLE: Application usability.
DESC: The application should be very simple to use, especially for the customer. Only necessary functions and limited options are presented.
RAT: In order for the application to be easy to use for people that are not used to technology.
DEP: -

ID: NR9
TITLE: Long battery life.
DESC: The application should not drain too much battery.
RAT: In order for the driver to not have to charge their phone all the time.
DEP: -

ID: NR10
TITLE: Application maintainability.
DESC: The focus should be on high cohesion and low coupling, as well as following a standard when coding.
RAT: In order for future developers to easily be able to maintain the code in the future.
DEP: -

Appendix A: UML Diagrams

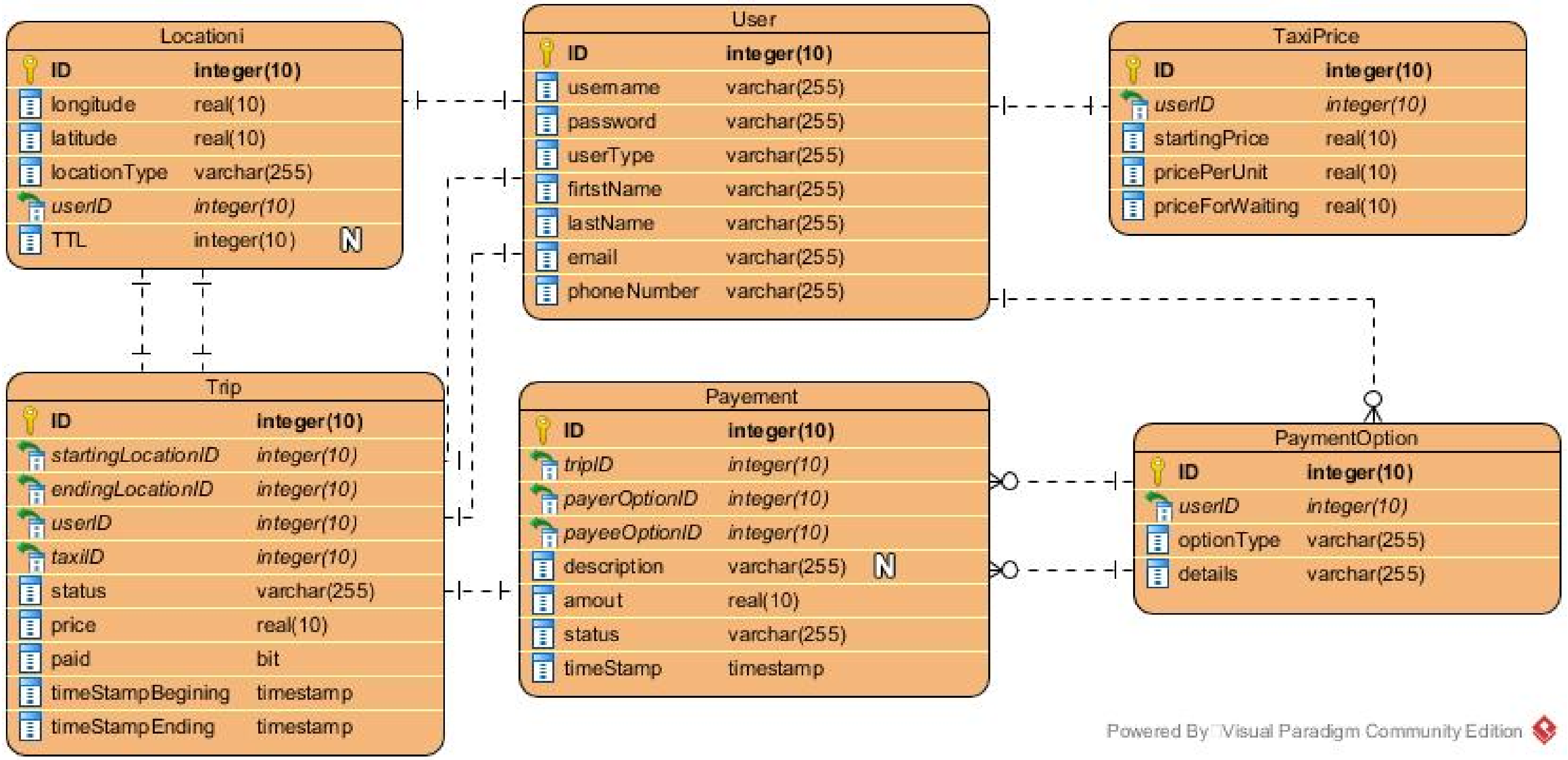


Figure 1: ER diagram

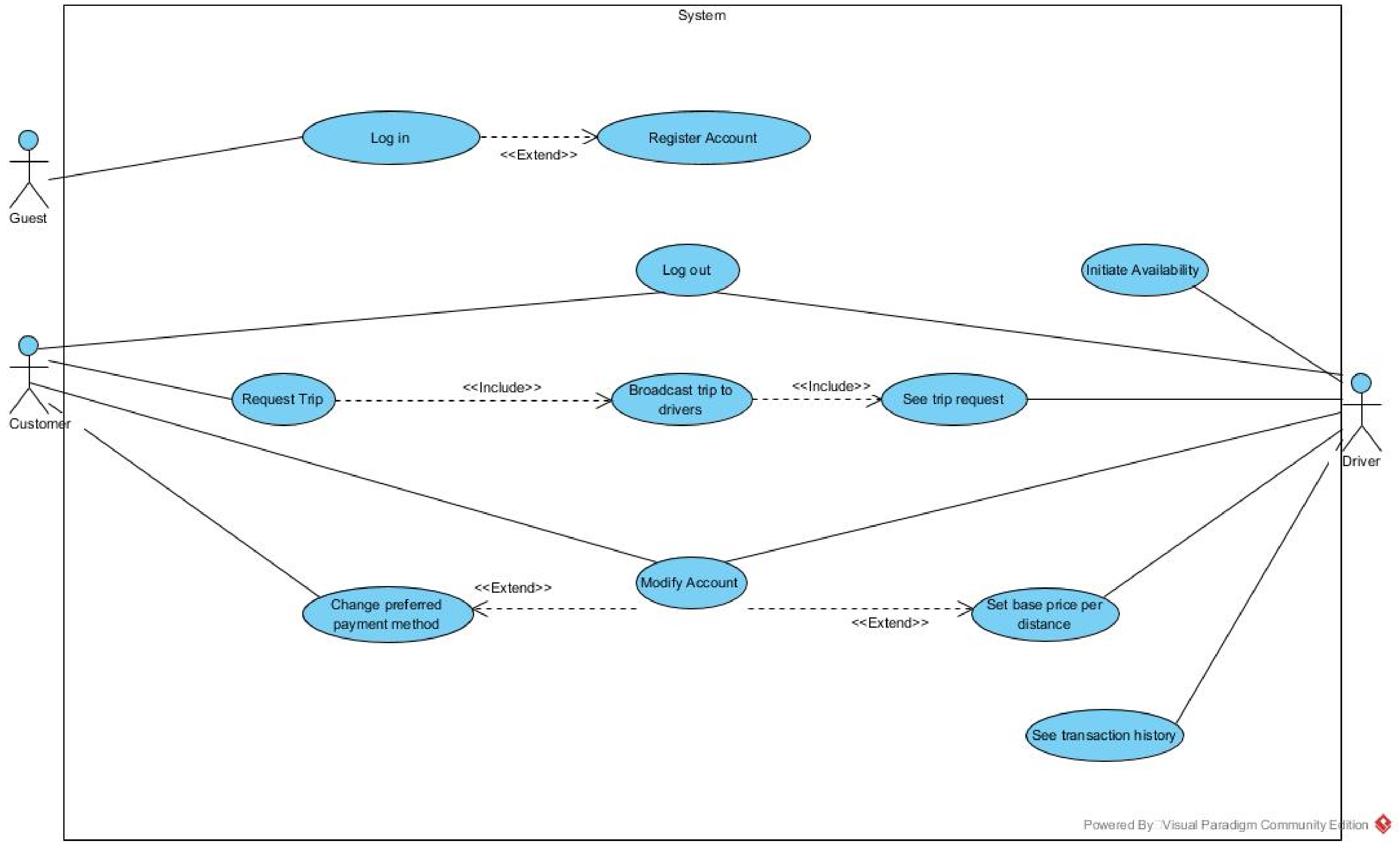
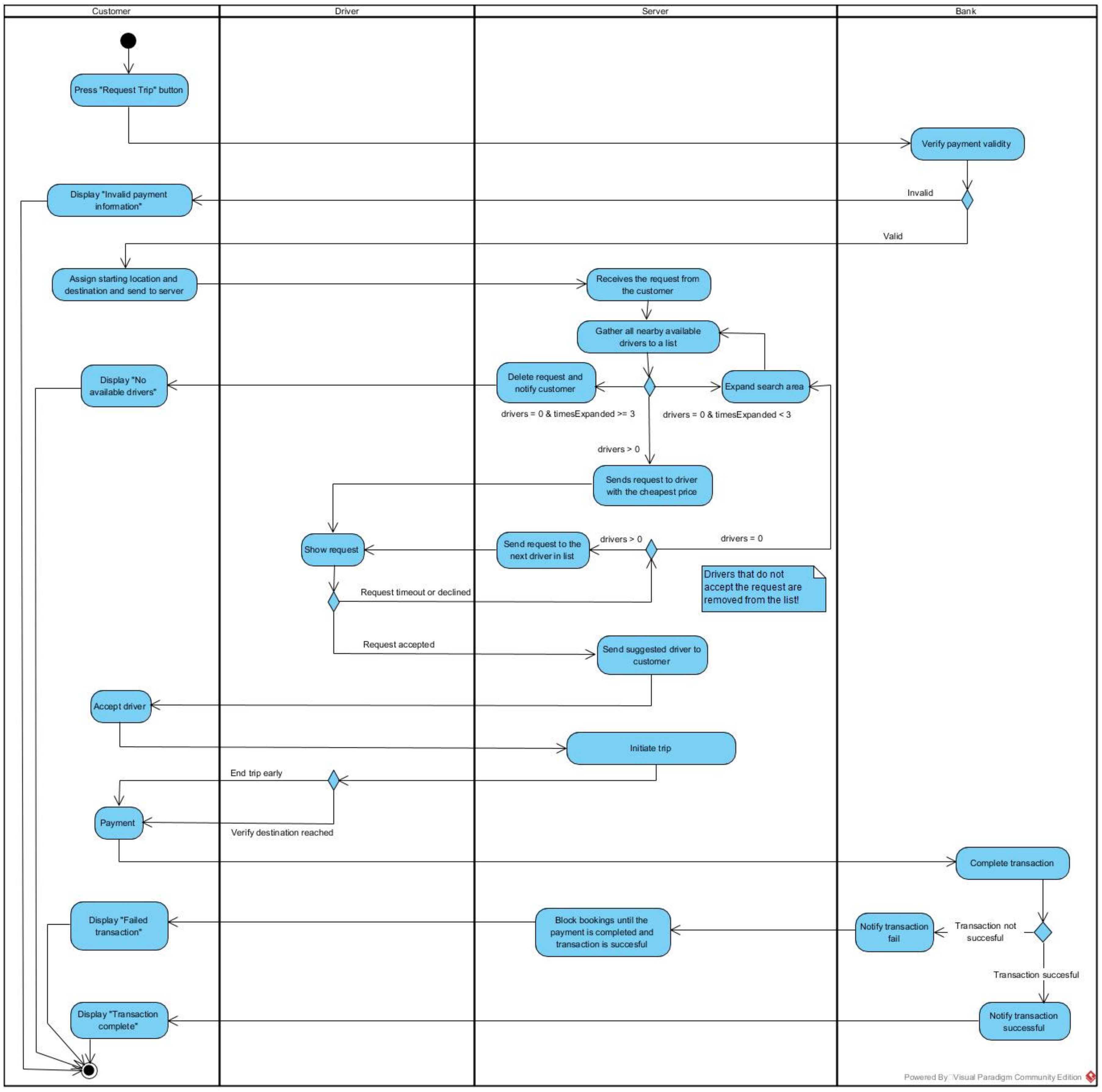


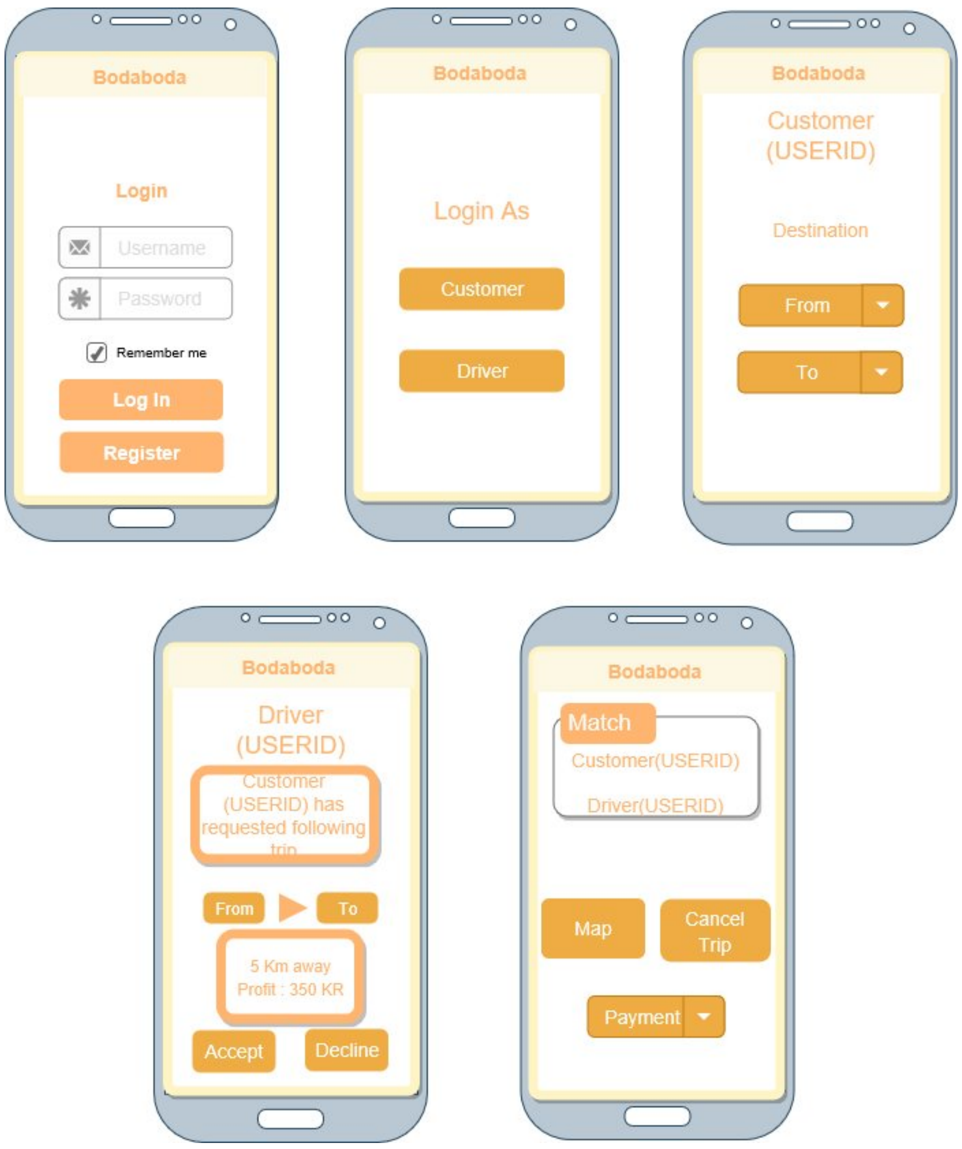
Figure 2: Use Case diagram



Customer
Driver
Server
Bank
Press "Request Trip" button
Verify payment validity
Invalid
Display "Invalid payment information"
Valid
Assign starting location and destination and send to server
Receives the request from the customer
Gather all nearby available drivers to a list
Display "No available drivers"
Delete request and notify customer
Expand search area
drivers = 0 & timesExpanded >= 3
drivers = 0 & timesExpanded < 3
drivers > 0
Sends request to driver with the cheapest price
Show request
Send request to the next driver in list
drivers > 0
drivers = 0
Drivers that do not accept the request are removed from the list!
Request timeout or declined
Request accepted
Send suggested driver to customer
Accept driver
Initiate trip
End trip early
Payment
Verify destination reached
Complete transaction
Display "Failed transaction"
Block bookings until the payment is completed and transaction is succesful
Notify transaction fail
Transaction not succesful
Transaction succesful
Display "Transaction complete"
Notify transaction successful
Powered By: Visual Paradigm Community Edition

Figure 3: Activity Diagram for a trip

Appendix A: Mockups



Customer (USERID) has requested following trip

Figure 4: Application Mockups