BodaBoda - A taxi service for motorcycle drivers in East Africa

Group 1

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

This document covers the planning of the BodaBoda project. The application for the company named Okapi Finance is developed. The application provides a taxi service for motorcycle drivers in East Africa.

1.1 Client

Okapi Finance is offering financial services mainly in Africa with a mission to bank the unbanked. Okapi's services increases accessibility for financial services. They believe that everyone should be able to access financial services regardless of where they live and their status. Transparency, security, and compliance with financial regulations are key for their operations and they believe in building the Okapi network through partnership with financial institutions and other existing structures. Their goal is to be the pan-African fintech (financial technology) solution [1].

1.2 Product

The product produced is an application called BodaBoda, an application for motorcycle taxis in East Africa. The drivers can state that they are available and customers can use the app to find the closest free driver who is willing to drive them to their destination. The app will tell the driver where to go to find the customer and it will track the driven distance and calculate a price for the customer. The app can also help the driver reach the final destination. The customer can pay in her app and the driver will be able to track her received payments and see how much she has earned on a daily, weekly or monthly basis.

2 Group Organization

This group consists of 7 people, responsibilities and names of the people in the group can be seen below.

2.1 Project manager

Aleksandar Aćimović is assigned as the project manager for this project.

2.2 Project group

The team consists of seven people, namely:

- Aleksandar Aćimović - Project manager, Developer
- Linus Sens Ingels - Client communication, Developer
- Jonathan Major - Trello responsible, Developer
- Zacharias Leo - Worktime tracker, Developer
- Sireesha Kumari Kulari - Tester, Developer
- Stanislas Pedebearn - Template creator, Presentation assistant, Developer
- Chingiz Esenbaev - Presentation responsible, Developer

2.3 Role clarification

- Project manager - Manager of the project, responsible for the planning as well as making sure that everyone is doing what they should do.
- Client communication - Responsible for communicating with the client.
- Trello responsible - Responsible for keeping the Trello board clean and up to date.
- Worktime tracker - Responsible for keeping track of the working hours of each group member.
- Tester - Responsible for testing the application during the implemenetation phase.
- Template creator - Responsible for creating the templates for the documents.
- Presentation assistant - Assisting with the presentations, helping with designing them.
- Presentation responsible - Responsible for making making sure the presentations are created in time.
- Developer - A general role that will later on be divided into subgroups, depending on the expertise of each team member.

2.4 Steering group

The people evaluating the project:

- Jan Carlson
- Robbert Jongeling

3 Work organization

The group will have weekly meetings with both the steering group and the client, in these meetings the group's weekly progress and how the work has been organized will be discussed. The group will also have meetings when necessary. Communication with the client and steering group will primarily be done in form of e-mails, but will also be done in person during the meetings. Communication between the group members is done through Slack between meetings. Trello is used to always track the progress for the different activities allocated in the project. Time reporting will be measured via an Excel document where each member reports the time to the worktime tracker who then adds it to the document. The time will then be calculated to measure the total time.

3.1 Tools

The primary tools used in the project will be:

- Slack - Communication
- Git (github) - Project version and repository
- Trello - Task management
- Overleaf - Documentation
- Visual paradigm - UML
- Excel - Keep track of working hours.

3.2 Work distribution

The tasks will be divided by previous knowledge and therefore those people will be considered equals and experts on the field. In a case of impossibility of finding a common solution inside the group, the problem should be presented as soon as possible to our project manager so he can take the risk of deciding to go with one solution. Distributing teamwork evenly helps to keep the team in balance, on target and productive. Main points of work distribution will be:

- Tasks can be completed in a sequential way or in a parallel way.
- Give each member of the team an equal amount of work.
- Assign each team member a job in her area of expertise.
- Give each team member a task that she has the resources to complete. Make sure the work distributed among team members is within their capabilities, or give them the necessary supplies and information to complete their tasks.
- Allow each team member a reasonable amount of time to complete the task. Otherwise the results may be substandard.

3.3 Meetings

- Steering group - The group will briefly discuss the progress in the project. In these meetings the primary focus will be on the topic of how the work is organized between the meetings. Also, it will be discussed what was done in the previous week and what is planned to be done in the ongoing week. The meetings are there to help us deal with and remove any uncertainties.
- Client - Requirements, implementation and design will primarily be discussed. If the group is having design or implementation problems, these shall be talked about and a solution to them will be discussed. The group will also check continuously during these weekly meetings if the client is satisfied with the work we have done.
- Internal - If something is unclear or is of high priority the group will try to schedule a live meeting, so the solutions of the problems can be discussed and implemented as fast as possible. There is a group meeting every Monday before the meeting with the steering group to prepare what should be brought up. Other than that, the internal meetings are scheduled dynamically with a minimum of one meeting every week.

3.4 Ensuring product quality

Communication between the group members and weekly communication with the client are the most important ways of ensuring high quality. In this way the group can ensure that the deliverables are exactly what the client has requested and that the final product will be what was expected by the client. Documenting every decision and detail about the product will ensure that anyone can read the document and understand the reasoning behind every decision and what has to be done. Before documentation every detail has to be discussed within the group and with the client to make sure that everyone agrees to the decisions. When everyone is on the same page about the development, it will ensure the quality will be higher. By creating detailed design diagrams visualization of the decisions and the interpretation of the product is made easier, this results in another level of safety regarding correctly understanding the requirements presented by the client.

4 Schedule

The upcoming section will include all of the identified deadlines & deliverables for this project, the activities will be identified based on all the deliverables.

4.1 Deadlines & Deliverables

Project plan	22 November
Design description V1	6 December
Product V1	6 December
Project report	17 January
Design description V2	17 January
Product V2	17 January

5 Development

5.1 Functional

The system will provide a simple solution to a client who wants to take a taxi to a given location. Furthermore, the system will signal a driver to the clients location after the customer has accepted the assigned driver's price. Once the driver signals that the trip has started the system will calculate the current price for the trip based on how far both the client and driver has traveled on the given route. The goal of the system is to provide a simple and quick taxi service with high usability. The system will provide security for both the client and the driver, the client is the only one that can change the location of trips and the system will assure that the driver always gets paid the right amount, regardless if the trip is canceled halfway or completed based on the calculations stated above. Also, having in mind that the context for which the application is being developed, the application is developed to be as lightweight as possible supporting older devices and location services, it also needs to be optimized for better battery consumption.

5.2 Technical

The system will be implemented as a client-server architecture. On the back-end a RESTful application server shall be implemented using .NET CORE. For the purposes of authentication and authorization JSON Web Token (JWT) shall be implemented and for more security sensitive data shall be encrypted using RSA. The chosen database is MySQL. These two shall be converted in order to be run within a Docker container and hosted on a cloud service platform. On the front-end an Android native mobile application will be created. The Google Maps API is used for visualization as well as for calculating the best route and distance for a trip.

The git repository is structured as follows:

- BodaBodaServer (Directory)
  - Files and directories related to the implementation.
- Documentation (Directory)
  - Diagrams&Other (Directory)
    - A couple of diagrams (Files)
  - The documentation of the project (Files)
- README (File)

5.3 Users

The system will have three type of users: Guest, Customer and Driver. Admin functionality will be outside of the application.

- **Guest** - The minor user type of the system. The guest is able to download the mobile application and register to the system using it.
- **Customer** - The customer can request a trip by typing in the starting location and the ending location. When a driver has been found the application will display the name of the driver, the price of the trip and the time it will take for the driver to reach the starting point. Once both parts has accepted the trip the driver will be guided to the starting location. The customer can at any time cancel the trip and will pay the calculated price based on how far both the client and driver has traveled on the given route.
- **Driver** - If a customer wants to become a driver, she can either switch to driver mode in the configuration page or register as a driver. After the customer has switched to become a driver she will be presented with new functionality in the application. The driver can see pending trips, choose mileage price + fee on top, accept a trip, start a trip and see transaction history with statistics.

5.4 Constraints & Assumptions

5.4.1 Constraints

The biggest constraint will be that many of the customers are using older versions of Android. Which means that the application has to be optimized for use on older version devices. Since the application will be used for long durations of time by the drivers, battery life and consumption must be taken in consideration while developing and optimizing the application.

5.4.2 Assumptions

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 taking a noticeable amount of time.
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.

5.5 Initial backlog

The initial backlog items consists of:

- Requirements - The overall requirements of the project
  - Guest
    - A user login which helps the user to login to a account.
    - An account registration to let the guest make a new account.
  - Customer
    - A customer should have the possibility to modify account information for changing the user information.
    - A customer should be able to request a trip that will be shown for the drivers.
    - A customer must be able to pay for the trip as payment are required.
    - A customer should have the ability to cancel a requested trip if the customer needs to.
    - A customer should be able to see the payment verification when paying.
    - A customer will have a option to become a driver directly.
  - Driver

    - A driver should have the possibility to modify account information for changing the user information.
    - A driver should be able to change the availability status to make the driver available or not.
    - A driver should be able to accept ride requests when the customer have requested a trip.
    - A driver should get the trip starting point to be able to find the customer.
    - A driver should be able to set the mileage price for the trip.
    - A driver should be able to set a fee for the trip.
    - A driver should be the one initiating the ride when the customer have entered the vehicle.
    - A driver should be be able to end the ride when reaching the destination or canceling the trip.
    - A driver should be able to see the payment verification when the customer pays for the trip.

- Diagrams - In the beginning stage this will consist of diagrams that will help the group and stakeholders to understand the functionality of the system. In a later stage more diagrams will be produced when needed, for instance class diagram and sequence diagram.

6 Risk Management

Creating a Risk Management Plan is a critical step in any project, as it helps to reduce the likelihood of a risk from occurring. The initial risk assessment attempts to identify, characterize, prioritize and document a mitigation approach relative to those risks which can be identified prior to the start of the project. The risk assessment will be continuously monitored and updated throughout the life of the project.

Risk 1: Data loss
Example: Some files deleted by mistake. Server got crashed.
Level: High
Mitigation Strategy: Regular backups are necessary.

Risk 2: Missing the internal scheduled deliverables
Example: If someone gets stuck in one thing, there might be a risk that she will continuously look at the same issue, wasting the time that she could have used for something else.
Level: Medium
Mitigation Strategy: Flagging the issue in time.

Risk 3: Improper estimation of resources
Example: In some cases we can choose problematic areas wrong and the group assigned will not be able to resolve the issues.
Level: High
Mitigation Strategy: Based on the size, competencies of the team and available time, the scope of the project should be estimated appropriately.

Risk 4: Lack of communication within team

Example: Team Members are working in a distributed environment with different course schedules
Level: Medium
Mitigation Strategy: Every possible misunderstanding needs to be addressed as soon as possible and this will be a task for the project manager to keep track off.

Risk 5: Unavailability of a member
Example: If someone is not available for some time, the entire deliverable should not depend on that person. The group should not get stuck because one person is missing.
Level: High
Mitigation Strategy: A good documentation of an individual work is required. Also, regular meetings are helpful to know what an individual is doing. In case a person is missing other group members will be able to continue where the missing person left off with the use of the documentation. If a group member knows that she will be unavailable, she needs to inform at least a week in advance.

Risk 6: Misinterpretation of client requests
Example: Misinterpreting a requirement from the client, which might lead to a completely different product than requested.
Level: High
Mitigation Strategy: Proper estimation of what the client is looking for needs to be done. Continuous client meetings, as well as diagrams modeling the functionalities of the application helps mitigating the communication errors. This will give the client opportunities to propose changes.

Risk 7: Lack of knowledge
Example: Some members are not aware of how to work with the chosen technology.
Level: Medium
Mitigation Strategy: Mainly going with the technologies that we are familiar with. Also doing some research on the parts we are unsure about.

7 References

[1] "Okapi", Okapifinance.com, 2018. [Online]. Available: https://www.okapifinance.com/. [Accessed: 26- Nov- 2018].