



Bilkent University

Department of Computer Engineering

Senior Design Project

Project name: Wheelancer

Project Specification Report

Onat Korkmaz
Muharrem Berk Yıldız
Muhammed Maruf Şatır
Ümit Çivi
Erdem Ege Eroğlu

Supervisor: Fazlı Can

Jury Members: Shervin Rahimzadeh Arashloo, Hamdi Dibekliolu

Innovation Expert: Murat Ergun

October 11, 2021

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

1. Introduction	3
1.1 Description	3
1.2 Constraints	4
1.2.1 Implementation Constraints	4
1.2.2 Security Constraints	5
1.2.3 Technological Constraints	5
1.2.4 Language Constraints	5
1.3 Professional and Ethical Issues	5
2. Requirements	6
2.1 Functional Requirements	6
2.1.1 Logistics Optimization and Matching	6
2.1.2 User Reviews	6
2.1.3 Detailed User Profiles	6
2.1.4 Comprehensive Reporting	6
2.2 Non-functional Requirements	7
2.2.1 Accessibility	7
2.2.2 Accuracy	7
2.2.3 Availability	7
2.2.4 Backup and Recovery	7
2.2.5 Extensibility	7
2.2.6 Performance	8
2.2.7 Reliability	8
2.2.8 Security	8
2.2.9 Testing	8
2.2.10 Usability	8
2.2.11 Exception Handling	8
2.2.12 Legal and Regulatory Requirements	9
2.2.13 Maintainability	9
3. References	10

1. Introduction

The shipping sector has developed significantly in the last decades due to the advancements in transportation and other technologies such as e-commerce. All of these advancements led to a dramatic increase in the amount of cargo that has been shipped by individuals and companies. In order to tackle this huge demand, many companies that focus on the delivery of cargo have been founded throughout the last decades. Nevertheless, the abundance of the cargo companies caused some problems. For example, people are struggling to choose a trustworthy cargo company that will not harm their goods. Moreover, the disappearance of their cargo makes people hesitate to utilize cargo services. There are also well-known companies such as UPS that have accomplished billions of package deliveries in 2018[1]. Although these companies can deliver goods without a problem, the cost can be expensive even if you want to send a small package. Last but not least, in the pandemic, demand for cargo shipment has increased due to the fact that people started to use e-shopping more frequently. Moreover, the pandemic caused many people to lose their jobs. Some of these people who own a vehicle started to use apps like Uber, BlaBlaCar to gain the money they need to sustain a healthy life.

Therefore, in this project, we aim to design an application that will retain the good qualities of cargo companies while reducing the price for the transportation and help people by enabling them to have another source of income in the pandemic.

With this project specifications report, we aim to explain the description of our shipment application, the constraints we will enforce, functional and non functional requirements, and the professional and ethical issues that will arise during the development of our project Wheelancer.

1.1 Description

Wheelancer is a mobile application that runs on Android operating system. It provides a matching service for individuals who want their cargo to be carried and individuals who want to carry cargo to earn money. The matching system will work as the shipper will select a destination and the senders will specify the destination address and the size of the load and can add note information about the load for the cargo. When the possible matches between senders and shippers occur, the

senders and shippers can communicate with each other. Before they agree upon shipment, the possible price with respect to the specified vehicle and distance will be calculated by the app. Although the potential price is declared by application, the sender and shipper can agree upon another price if they are not pleased by the recommended price. The payment method includes cash and online payment. If the online payment is chosen by the users, then the application will take the money from the user and will keep the money. If the transportation is completed without a problem, the money will be sent to the shipper. If some problem occurs in the shipment process, the money will be sent back to the sender. The shipper must open their GPS so that the sender can see the live location of the cargo.

In case some companies want to ship their goods, they can choose an option for massive delivery. They also must choose who will be responsible for the loading of goods into the vehicle and unloading of the goods from the vehicle. The shippers with the large load size and with the matching option of the responsibility of the loading will be possible matches.

Furthermore, shippers will be able to view the path between the current location and the destination.

There will be a user star review system so that the sender can understand whether a particular shipper is trustful or not. There is also a badge system which will be given to carriers that have completed successful deliveries and earned the trust of customers, which will be an incentive for carriers to work harder for more income. The punctuality of the shippers will be rewarded with the badges. Nevertheless, the shippers with the low rate of punctuality will take the badges indicating their punctuation.

Moreover, there will be an insurance option to protect the load in case some problem occurs. The insurance type or price changes depending on the product value and it is optional.

1.2 Constraints

1.2.1 Implementation Constraints

- As for version control, Github and Git will be used to collaborate while implementation.

- Javascript, Node.js and Express.js framework will be used for REST API.
- Java, Android JDK will be used to develop android clients.

1.2.2 Security Constraints

- The carriers can not know or see the content they are carrying because of privacy purposes. Therefore any doubts about the carriers are minimized.

1.2.3 Technological Constraints

- Carriers must keep their location services on through their mobile phones during the delivery session.

1.2.4 Language Constraints

- Wheelancer will initially serve in only English. We are planning to add additional language support according to demand.

1.3 Professional and Ethical Issues

Privacy of any data, whether entered by the user or the admin, is an issue that needs to be considered thoroughly. During the development of the app, security of the data will be one of the top priorities to prevent any data leaks to protect the privacy of the users.

Other cargo companies, such as Yurtiçi Kargo, may not be willing to cooperate with the app for various reasons. Therefore, different strategies that will benefit both sides should be considered so that Wheelancer can be used extensively.

Since our application heavily relies on individuals, it is prone to incidents. To prevent incidents such as disappearance of cargo due to carrier mistakes, all carriers in our app will sign a contract that holds them responsible for their mistakes and we do not guarantee against any fraud.

2. Requirements

2.1 Functional Requirements

Wheelancer will be capable of following functionalities:

2.1.1 Logistics Optimization and Matching

In order to start the process of delivery suitable users and carriers should be matched. Carriers specify their travel route with specific departure and a calculated arrival time according to the route. Users specify their cargo with departure and arrival location.

- Carriers can specify max load and delivery vehicle type.
- Multiple cargo can be delivered by one carrier.
- Reward and penalty system for punctuality.
- Courier service for delivery companies.
- A cargo can be delivered using multiple carriers.

2.1.2 User Reviews

Our application heavily relies on the reliability of the carriers so providing them a place to portrait their performance of previous deliveries is crucial. For that reason, Wheelancer will allow users to rate for carriers after a complete delivery. Users will be able to rate performance of the carrier in terms of speed and quality. Encouraging carriers to put extra care and provide users even more qualified services.

2.1.3 Detailed User Profiles

Apart from user reviews, carriers can showcase trustworthiness with their badge and trophies. Every delivery will be logged in our system and with each reached milestone they will earn badges which will indicate experience and loyalty.

2.1.4 Comprehensive Reporting

Each trip has a delivery report that can be examined by the user. The report includes:

- For carriers, revenue of the delivery can be examined which is calculated by subtracting expenses gathering vehicle fuel consumption and fuel price.

- For users, live tracking of the vehicle's position and status of delivery.

2.2 Non-functional Requirements

2.2.1 Accessibility

- Since the new and useful features such as “turn by turn navigation”, Android Marshmallow 6.1 or newer version is required in order for users to run our system [2]. Node.js will be used for the REST API.

2.2.2 Accuracy

- The application will ask for shipment specifications. Particularly, if there are fragile items in the cargo, drivers can be informed about the cargo box in advance.

2.2.3 Availability

- The application uses the GPS signals of drivers, so even if he/she does not keep the internet connection open, the client can be informed about the location of the cargo.
- Our initial audience will be Turkey, thus, bug fixes and updates should occur at midnight. So that it is aimed to protect our users from inaccessibility to the system.

2.2.4 Backup and Recovery

- Our database will be MySQL. Therefore, MySQL undertakes backup and recovery of the data.
- Personal data and cargo preferences should be stored in local and global storage so as to any data will not lost.

2.2.5 Extensibility

- It must be open to developments on the basis of new features and new functionalities. (Payment methods, security precautions)

2.2.6 Performance

- List of drivers should display to users in 4 seconds.
- Data updating should not take more than 0.5 second.
- Query responses take under 1 second.

2.2.7 Reliability

- Users need to sign the user agreement in order to provide application reliability. Just in case, their information may be shared with the court or police.
- Any crash on account of software, should not occur in the system.

2.2.8 Security

- The users must sign up with their private credentials.
- The application provides the integrity of the customer's private information.

2.2.9 Testing

- The web server, mobile application and database will be tested regularly to eliminate errors or any mistakes. Some sort of test:
 - Load tests to measure performance according to actual user behaviors.
 - Reliability tests will be made to control the web server and function properly.

2.2.10 Usability

- The users will be able to give their suggestions as feedback to developers.
- The user will not spend time learning the functions of the application. The GUI will be intuitive and user friendly so user interactions would be easy.

2.2.11 Exception Handling

- During execution, if any error/exception occurs, the user will be shown and explained to the user. In this way, the user will be explained what to do when he/she encounters an error.

- If the user encounters an error that has not been explained by the developers, he/she can contact the developers about this exception.

2.2.12 Legal and Regulatory Requirements

- The application would not allow drivers to know about cargo content.
- The contract signed by the users will prevent crimes significantly.

2.2.13 Maintainability

- Subsystems will be loosely coupled so that maintenance will be done easily. Moreover, the integration of any new module during updates will be easy.

3. References

- [1] “*(PDF) mapping human activity volumes through remote ...*” [Online]. Available: https://www.researchgate.net/publication/344494578_Mapping_Human_Activity_Volumes_Through_Remote_Sensing_Imagery. [Accessed: 08-Oct-2021].
- [2] “*Android versions comparison: Comparison tables,*” SocialCompare. [Online]. Available: <https://socialcompare.com/en/comparison/android-versions-comparison>. [Accessed: 06-Oct-2021].