Project 13 – Fleet of Delivery and pick-up Vehicle management System
Software Requirements Specifications

Sheena Philip,Linda Khumalo,Kessigan Subramanium,Phumzile Dlwathi

April 10, 2016

1 INTRODUCTION

1.1 PURPOSE

This document details the software requirements specifications for the Fleet of delivery and pick-up Vehicle Management System project. Template A.5 from the IEEE Recommended Practice for Software Requirements Specifications has been selected. The purpose of this project is to enable monitoring of all aspects of a fleet of delivery and pickup vehicle management system via a browser based management system. The system will allow streamlining of services as any action carried out by any stakeholders will be captured and monitored on the management system. Depending on the user; users will be able to view work schedules, routes and package progress among other things. The system will also provide database functionality. In overall, the aim of the project is to make the processes involved in a fleet of delivery vehicles more efficient, transparent and easy to monitor remotely whilst allowing clients to track their packages.

1.2 INTENDED AUDIENCE AND READING SUGGESTIONS

This document is aimed at:

- developers to serve as a guide during development as to what the final product should look like and what features and tests should be carried out

- the end user to see what the final product will look like

1.3 PROJECT SCOPE

The end product will be a browser based application which will implement the features described in Section 3 under system features. The product will not be available as a phone application therefore mobile users should log in to their browsers to access the product.

1.4 REFERENCES

See the enhanced project description available from the course Sakai page.

2 EXTERNAL INTERFACE REQUIREMENTS

2.1 USER INTERFACES

Various users of the system have been identified. Namely, the driver, the dispatcher, the manager, a sender, a receiver, the accounts officer, the CEO, the operational officer, the system administrator and the vehicle maintenance officer.
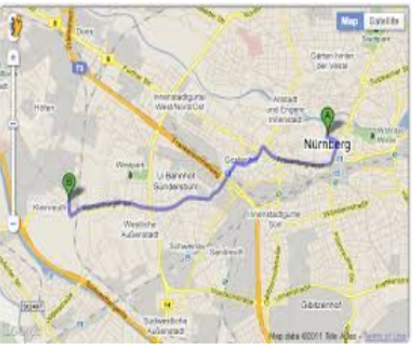
2.1.1 THE DRIVER INTERFACE/VIEW

Figure 2.1: The driver interface

2.1.2 THE DISPATCHER VIEW

The following figure shows you the dispatcher view before the dispatcher runs the algorithm to assign packages to drivers and routes.

Figure 2.2: The dispatcher interface before running the allocation algorithm

The following figure shows you the dispatcher view after the dispatcher runs the algorithm to assign packages to drivers and routes.
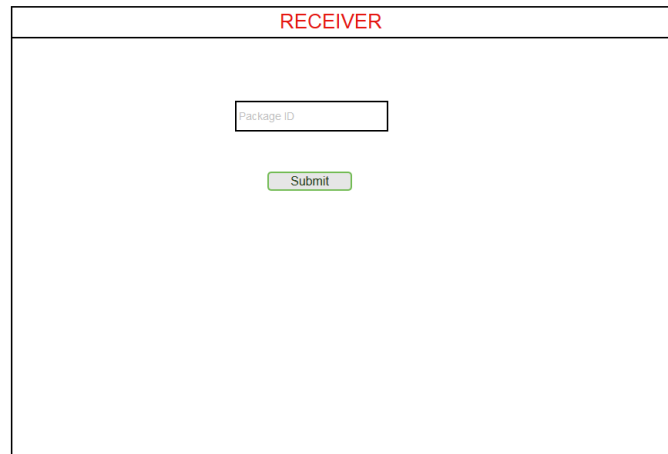
Figure 2.3: The dispatcher interface

2.1.3 THE RECEIVER VIEW BEFORE USER SUBMITS

In order for the receiver to track the progress of their package they have to enter the package notification sent to them via email notification. Figure 2.4 shows the page where the user

enters their package id.
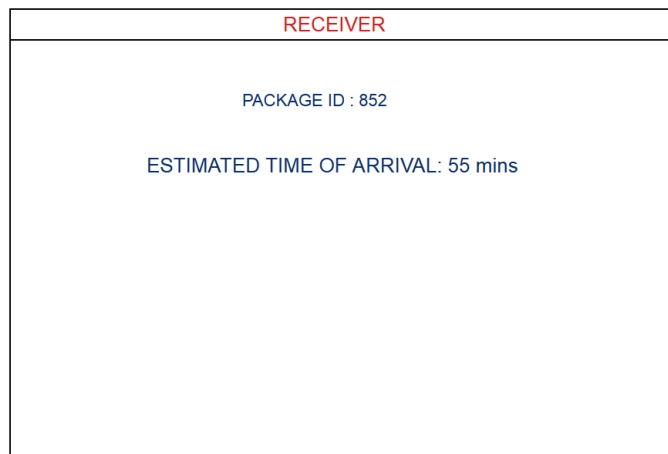
Figure 2.4: The receiver interface

2.1.4 THE RECEIVER VIEW AFTER THE USER SUBMITS

Figure 2.5 shows the interface the receiver sees after they have entered their package details. This view shows the package details and estimated time of delivery.

Figure 2.5: The receiver interface after the user submits query

2.1.5 THE SENDER VIEW

Figure 2.11 shows the sender view.

Figure 2.6: The sender interface

2.1.6 THE ACCOUNTS OFFICER VIEW

Figure 2.11 shows the sender view.

Figure 2.7: The accounts officer interface

2.1.7 THE CEO VIEW

Figure 2.11 shows the sender view.

Figure 2.8: The CEO view

Figure 2.11 shows the sender view.

Figure 2.9: The operational officer interface

Figure 2.11 shows the sender view.
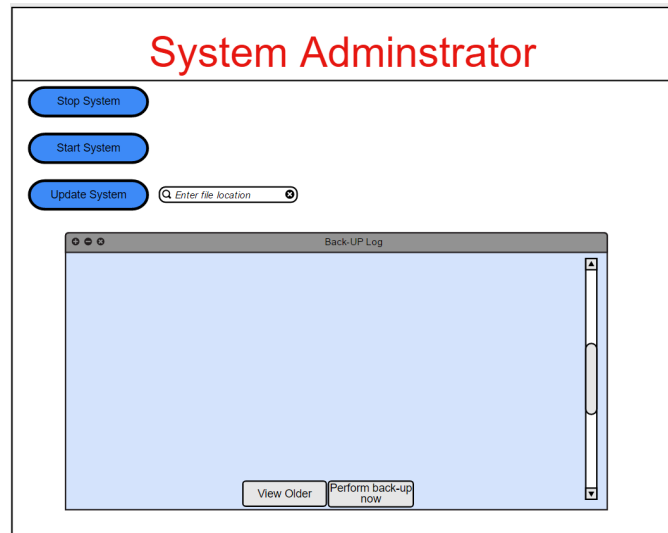
Figure 2.10: The system administrator interface

2.1.10 THE VEHICLE MAINTENANCE VIEW

Figure 2.11 shows the sender view.
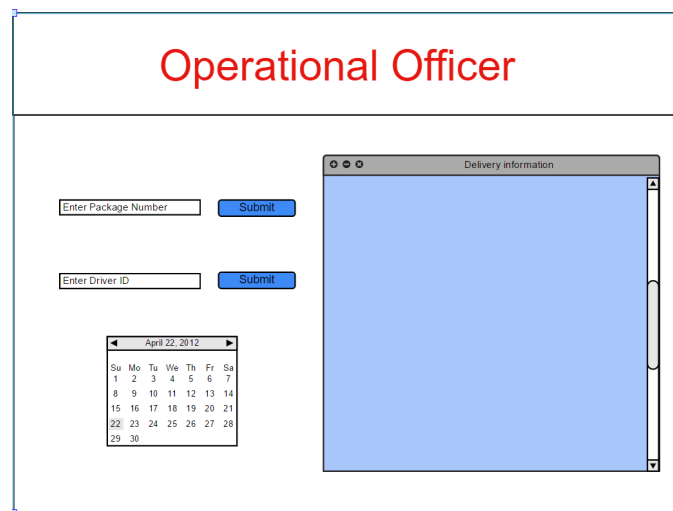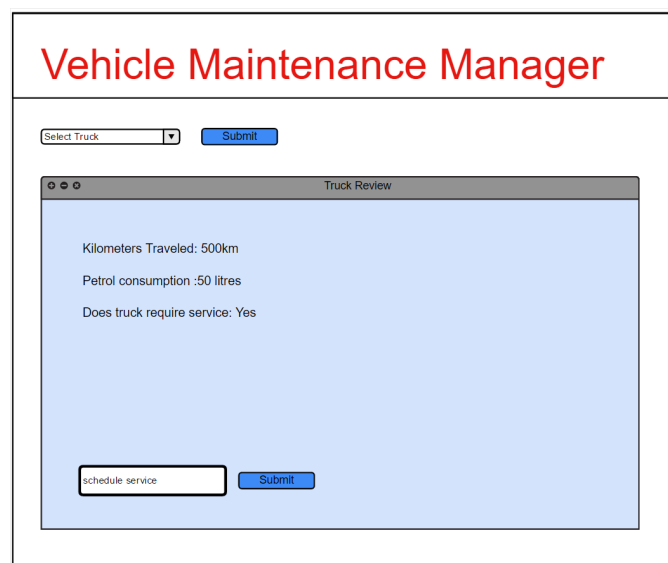

Figure 2.11: The vehicle maintenance manager interface

2.2 HARDWARE INTERFACES

- Mobile phone/ desktop/tablet

2.3 SOFTWARE INTERFACES

- Chrome or Firefox web browser enabled on device

2.4 COMMUNICATION INTERFACES

- Internet connection

- Email notifications

- Telephone calls

3 SYSTEM FEATURES

3.1 TRACKING OF PARCELS

3.1.1 INTRODUCTION/PURPOSE OF FEATURE

Parcel tracking enables the manager of the depot to see the status and locations of all parcels to be collected or delivered. This can be useful for when parcels go missing and they need to be found as the parcel would have been lost after its last known location. Parcel tracking also helps the dispatchers ensure that the correct parcels are in the correct vehicles so that deliveries can be made in the shortest possible time. The customers also benefit from this feature as they may track the parcel from its waybill number.

3.1.2 STIMULUS/RESPONSE SEQUENCE

The parcel is given its location from its point of origin, this is the pick up point that is logged when a driver firsts collects the parcel. The second location of all parcels will be the dispatch warehouse where parcels are stored and sorted. The parcel will then be sent to a delivery vehicle. As the delivery vehicle makes collections/deliveries, the parcels location will be updated.

3.1.3 ASSOCIATED FUNCTIONAL REQUIREMENTS

This feature requires an internet enable device for both the driver to relay the parcels position and the user to locate where the parcel is. These devices will then need to connect to a centralised server which is responsible for processing and storing all data requests.

3.2 TRACKING OF VEHICLES

3.2.1 INTRODUCTION/PURPOSE OF FEATURE

This feature allows the manager to have full access to the location of all vehicles in order to determine whether the drivers are following the right paths, no unnecessary stops are made and that all delays are accounted for. This can help the manager pick up any faults in the user algorithm and make the necessary changes. This is also useful for determining the location

of parcels as they can take their location data from the location of the vehicle in which they are contained.

3.2.2 STIMULUS/RESPONSE SEQUENCE

When the driver logs on to the system, the location of the driver is continuously fed through the internet to the companies servers. This is an automatic process that happens in the background when a driver is logged in.

3.2.3 ASSOCIATED FUNCTIONAL REQUIREMENTS

This feature requires an internet enabled device that also has a GPS system to calculate its position, network triangulation can also be used but this should be avoided. The device would then connect to a server via the internet which is responsible for processing and storing all data requests.

3.3 DETERMINE SHORTEST DISTANCE ROUTE

3.3.1 INTRODUCTION/PURPOSE OF FEATURE

The shortest distance route is used when a list of deliveries and pick up are acquired for an individual vehicle. A shortest distance algorithm which is based on Dijkstra's algorithm will be used. This algorithm will calculate the shortest distance between the first stop, the last stop and all the points in between. This will determine the path for the driver to take when making both pick ups and deliveries.

3.3.2 STIMULUS/RESPONSE SEQUENCE

The list of destinations are determined at the start of the day when the dispatcher runs the allocation algorithm. The system will then divide the locations into zones depending on the delivery vehicles available, i.e. each vehicle is responsible for its own zone. The system will then use the shortest distance algorithm to calculate the shortest distance for the driver between all points. This will be pushed to the driver's profile which is then accessible from an internet connected device.

3.3.3 ASSOCIATED FUNCTIONAL REQUIREMENTS

This requires processing power from a server to calculate all possible driving routes and determine the shortest route. The more efficient algorithm will lessen the load on servers. The driver will then have access to the information on the server through an internet connected device.

3.4 Determine the vehicle certain packages go into

3.4.1 Introduction/Purpose of feature

To optimise efficiency and be cost effective the system should be able to ensure that packages that are to be delivered or collected from the same area are in the same truck without exceeding the maximum weight limits placed on a truck.

3.4.2 Stimulus/Response Sequence

- The dispatcher has to press the Run Allocation Algorithm button. This results in the algorithms which determine which truck the package has to be placed in being run in the background. The inputs to the algorithms include destination/location of package, weight of package and truck specifications such as maximum weight allowed.

- When a user enters a package into the system for pick up the allocation algorithms will be run to determine which truck will pick up the package.

3.4.3 Associated Functional requirements

- The packages that will be allocated vehicles are both packages that are already at the storage house and packages that users have entered into the system for pick up.

- When a driver receives a route at the beginning of their shift they will not be allocated extra packages even if the package to be collected is in the vicinity of the driver's route. All routes and time tables are determined when drivers are allocated trucks at the start of their shift.

3.5 Allocate a driver to a vehicle

3.5.1 Introduction/Purpose of feature

When a vehicle has been allocated packages and the allocation is deemed complete; a driver will then be allocated to drive the vehicle. All drivers working on that day should be allocated at least 3 hours worth of driving work in a day. Drivers and vehicles are allocated according to availability. No driver has a specific vehicle allocated to them permanently.

3.5.2 Stimulus/Response Sequence

The dispatcher has to press the Run Allocation Algorithm button. After all the packages have been assigned trucks; the allocate drivers to trucks algorithm will be run. The inputs to this algorithm include the status of truck allocation and the available drivers list.

3.5.3 Associated Functional requirements

- This is only run when the algorithm for allocating truck has been run.

- Drivers are allocated depending on availability not how often they drive a certain vehicle.

3.6 Determine date and time of package pickup

3.6.1 Introduction/Purpose of feature

Depending on the route a package has been allocated to, it will be assigned a date and time of pickup. The further down a route a package is located the longer it will take for it to be delivered or collected.

3.6.2 Stimulus/Response Sequence

- The dispatcher has to press the Run Allocation Algorithm button. After all the packages have been assigned trucks depending on the location of the package, that is, how far down the route it is, a time will be allocated to it.

- When a user enters a package into the system for pick up the date and time algorithm will be run just after the vehicle allocation algorithm is complete.

3.6.3 Associated Functional requirements

- This allocation has to be done just after the packages have been allocated trucks.

3.7 Determine date and time of delivery

3.7.1 Introduction/Purpose of feature

Depending on the route a package has been allocated to it will be assigned a date and time of delivery. The further down a route a package is located the longer it will take for it to be delivered or collected.

3.7.2 Stimulus/Response Sequence

- The dispatcher has to press the Run Allocation Algorithm button. After all the packages have been assigned trucks depending on the location of the package, that is, how far down the route it is, a time will be allocated to it.

- When a user enters a package into the system for pick up the date and time algorithm will be run just after the vehicle allocation algorithm is complete.

3.7.3 Associated Functional requirements

- This allocation has to be done just after the packages have been allocated trucks.

3.8 Store payment information

3.8.1 Introduction/Purpose of feature

In order for a user to actually complete the process of being able to send a parcel they will first have to create an account. When an account is created, the user has to store their payment details.

3.8.2 Stimulus/Response Sequence

- The sender presses on the button create an account

- The sender then enters all their information

- The sender then presses the button submit which creates an account

3.8.3 Associated Functional requirements

This has to be done before a sender can actually send anything. A database has to be created which can store all the information

3.9 Store client details

3.9.1 Introduction/Purpose of feature

In order for a user to actually complete the process of being able to send a parcel they will first have to create an account. When an account is created, the user has to store their personal details.

3.9.2 Stimulus/Response Sequence

- The sender presses on the button create an account

- The sender then enters all their information

- The sender then presses the button submit which creates an account

3.9.3 Associated Functional requirements

This has to be done before a sender can actually send anything. A database has to be created which can store all the information

3.10 Package Storage and Package Identification

3.10.1 Introduction/Purpose of feature

This feature captures the data of the packages, when a sender sends a request for the package to be collected and after payment has taken place this feature will allocate a package number that will be used as the package identifier for the sender and the receiver and will allow both

the sender and receiver to track the package. It also calculates the available space in the storehouse, to calculate the estimated time of pick up since it will depend on the backlog storehouse.

3.10.2 STIMULUS/RESPONSE SEQUENCE

- The sender enters the package details then press submit.

- The sender enters their address then press submit.

- The sender enters the receiver's address then press submit.

- The sender creates an account for payment.

3.10.3 ASSOCIATED FUNCTIONAL REQUIREMENTS

- A database to store the package details.

- A database to store addresses.

- A server that will take care of the payments

3.11 SEND AUTOMATED EMAIL NOTIFICATIONS TO CLIENTS

3.11.1 INTRODUCTION/PURPOSE OF FEATURE

To notify the receiver that a package will be delivered to them and to send the link to the webpage that will show the receiver an estimated time of arrival of the package sent to them.

3.11.2 STIMULUS/RESPONSE SEQUENCE

- The client needs to click on the web link sent to them via email to access the receiver dashboard. Enter their package number and press submit

3.11.3 ASSOCIATED FUNCTIONAL REQUIREMENTS

- Package storage and identification functionality All sender functionalities must work before the receiver can receive anything

3.12 INVENTORY MANAGEMENT

3.12.1 INTRODUCTION/PURPOSE OF FEATURE

To make sure that there is enough storage space in the store room for the packages to be stored. This will determine the estimated time of pickup and delivery due to the backlog in the storehouse.

3.12.2 STIMULUS/RESPONSE SEQUENCE

- The sender should press submit after creating the payment account.

3.12.3 ASSOCIATED FUNCTIONAL REQUIREMENTS

- Package storage and identifier.

- A database to store the package details.

- A server that will take care of the payments.

4 PERFORMANCE REQUIREMENTS

- Loading times

- Algorithm run times

- The number of users at the same time

5 DESIGN CONSTRAINTS

- Time

- Resources (using only open source software)

- Data Acquisition

- Domain expert knowledge missing

6 SOFTWARE SYSTEM ATTRIBUTES

6.1 RELIABILITY

In order to establish the reliability of the software at a time of delivery, the following factors are required:

- Testing should be carried out concurrently with the writing of the code

- The client should be part of the design and testing processes

6.2 AVAILABILITY

This will not be defined for this project.

6.3 SECURITY

A main feature of the website is that depending on the person who is using it (sender ,dispatcher ,driver), they should only be able to access their relevant part. Therefore there some sort code that restricts communications between these different sections needs to be implemented. Cryptographic techniques should be used to protect the data. However this is beyond the scope of this project.

6.4 MAINTAINABILITY

The code will be made modular by using as many functions as possible to separate the modules. The code will be made as simple as possible, complex algorithms will be avoided. The code will be extensively commented and documented using Doxygen.

6.5 PORTABILITY

The system is meant to have one centralised server which handles all user requests. The system can be accessed by any internet enabled device as they only access a webpage.