FLEET OF TAXI SERVICE MANAGEMENT SYSTEM - SOFTWARE DESIGN SPECIFICATION

Danielle Winter (563795), Frederick Nieuwoudt (386372), Stephen Friedman (360938) & Sello Molele (0604606X)

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

1. INTRODUCTION

1.1 Purpose of this Document

This Software Design Specification (SDS) details the design for the WitsCABS application. The back- and front-end interfaces are described separately. The front-end and back-end integration is also described. The remainder of Section 1 details the project scope and abbreviations and definitions used in the remainder of the SDS. Section 2 addresses the basic system architecture and Section 3 gives a more detailed description of how the system will be implemented.

1.2 Project Scope

The taxi fleet management system is a tool that provides a service where a user can request a taxi and the system will dispatch the closest available driver.

The system back-end will be required to store information about drivers, customers, and the current pricing scheme. The system will keep historical data regarding trips that have been taken for auditing and dispute resolution.

The system front end will be presented in the form of a web interface. This will provide a means for Dispatch Control Centre agents to input new customer's details and for drivers to manage their current job requests. The web interface will also provide a quoted price for the client.

For the purposes of this project, the entire system functionality is described in the SDS. However, only a few base modules and components of the code are implemented as a prototype of the system in order to show basic functionality.

1.3 Definitions

1.3.1 Definitions

- WitsCABS: The name of the software application including front- and back-ends
- Driver: The taxi-cab drivers employed by the WitsCABS company to pick-up and deliver customers
- Customer: Clients of the company who request lifts

1.3.2 Abbreviations

- DCC: Dispatch Control Centre

2. SYSTEM ARCHITECTURE

2.1 Front-end Views

The front-end of the WitsCABS application is presented as two separate views in an HTML web-application. The front end is coded in angular.js using Twitter Bootstrap for a unified visual appearance and pop-up functionality. The respective interfaces are the DCC view and the Driver view. These views are linked through a common log-in screen view where the respective users select and log in to their interface.

2.1.1 DCC View

2.1.2 Driver View

2.1.3 Log-in View

2.2 Back-end Implementation

The back-end of the WitsCABS application consists of two main parts. The part that directly interfaces with the front-end is an Apache web server which runs on Ubuntu and executes Python scripts in order to interact with the database and external APIs. The part of the back-end that stores the data is an SQLite database.

2.2.1 Web Server

The role of the web server in the WitsCABS application is primarily to act as an intermediary between the front-end and the database. It is also required to connect to external APIs in order to deliver functionality that would be too complex to implement in the application.

The three external APIs are Paym8, SMSPortal and Google Maps. The Paym8 API is used to charge customers for Trips that are taken using WitsCABS. The SMSPortal API is used to notify Customers about the status of their Driver. The Google Maps API is used to translate addresses in to GPS Coordinates, as well as route from one location to another and determine the real distance between two points.

2.2.2 Database

The role of the database in the WitsCABS application is to store data so that different users can interact with the same data without directly transmitting the data between each other and so that data can be kept for auditing and dispute resolution if need be.

For the database queries, either many simple queries could be implemented or few complex queries could be implemented. Complex queries were chosen as they take advantage of optimisation in SQL. Complex queries require more development and testing time, but once implemented and verified produce faster results than many simple queries.

The database consists of the following tables:

- Users: This table contains information to verify a User when they log in. The users' username and hashed password are stored. The hash of the password is stored so that nobody that has access to the WitsCABS database can compromise the users' passwords.
- Agents: This table contains the real name of DCC agents and is used by the login process to determine if the user is a DCC agent.
- Drivers: This table contains the real name of Drivers and is used by the login process to determine if the user is a Driver. This table also keeps track of the status and location of the Driver so that the best driver is chosen for each Trip.
- Cost: This table contains rates that can apply to trips.
- Customers: This table contains details for Customers, including contact information for SMS notification and financial information in order to charge Customers for Trips that are taken.
- Locations: This table stores addresses against GPS coordinates, so that friendly names can used for informative purposes, while accurate GPS coordinates can be used for distance calculations.
- Trips: This table contains information relating to each Trip undertaken. It allows DCC agents to view all active Trips, it allows Drivers to be easily assigned to Trips and it allows for the retention of historical data for auditing or dispute resolution purposes.

3. DETAILED ARCHITECTURE AND FEATURES