FLEET OF TAXI SERVICE MANAGEMENT SYSTEM - LABORATORY EXERCISE 1

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

A software database system for taxi service delivery is proposed in this Software Requirements Specification. The objective of this document is to describe in detail the user requirements as well as the specifics of what the system requires and how it will be developed.

The aim of the system is to respond to a customer request for transport to a location. It will then find an available driver for the job and deploy their taxi for service. It will also allow for users to manage profiles and administrators to manage diver profiles.

2. Front End

The front end of the system is presented as a user interface to customers. The front end takes in customer requests as the input and returns a list of available taxis and predicted journey time as well as delay until taxi arrival. The front end shall be coded in html using angular.js.

The front end of the system will be managed by Danielle Winter and Frederick Nieuwoudt.

3. Back End

The back end of the system is a SQL database. It will store details about drivers, customers, cost and will keep historic data. For drivers it will need to store at least the driver's availability, location, personal details (including photograph) and car details (including number plate). For customers, at least basic information is stored, such as name, cellphone, location and payment details. A standardised cost per kilometer will be specified and the travel expense for the customer calculated according to the distance travelled. A link between driver and customer, along with start and end locations will be stored in order to track historic information.

The back end of the system will be implemented by Sello Molele and Stephen Friedman.

4. Business Logic

When the application receives a new request for travel, it will determine the closest available driver to potentially assign to the task. Based on the length and location of the route, the customer will then be shown the cost and reject or accept the order. On completion, the customer will be charged for the trip.