University of the Witwatersrand

School of Electrical and Information Engineering

Private Bag 3, 2050, Johannesburg, South Africa

Taxi-cab Service Management System

Front End:	*Student Number:*
Danielle Winter	563795
Frederick Nieuwoudt	386372
Back End:	
Stephen Friedman	360938

11 April 2016

Abstract

Contents

1 INTRODUCTION

1.1 Project Overview

1.2 Problem Statement

The WitsCABS company owns a fleet of taxi-cabs and employs drivers for customer transportation within a city. The city is demarcated into service zones with a single taxi-stand in each service zone. Cab drivers wait for clients at their closest taxi-stand. Each driver has a smartphone, embedded with a GPS, maps and navigational tracking and access to the WitsCabs web application. A centralised Dispatch Control Centre (DCC) receives calls from customers and processes the new requests. WitsCABS requires the software application to organise their taxi-fleet, coordinate drivers and manage new job requests.

1.3 Project Objectives

- Create the software requirements specification
- Create the design documentation for front and back ends
- Sprint planning and retrospective for front and back ends
- Implementation of prototype modules for the front end interfaces
- Implementation of prototype back-end modules
- Implementation of an Agile management technique

1.4 Stakeholders

The project stakeholders are divided into Users, Developers, the Project Manager and the Scrum Master. The respective stakeholders and their roles are documented below.

1.4.1 Users

Users are categorised into:

- Drivers

- DCC agents
- Customers

Drivers operate the taxi-cabs, are assigned jobs and pick up and deliver customers. They have access to a driver-specific front-end interface of the WitsCABS application and are able to specify when they accept a job or are on their way as well as state when they have completed a job. Drivers have access to the WitsCABS application through a smart-phone. They also have access to GPS and map navigation and are under navigational tracking.

DCC agents operate the WitsCABS call centre and have computer access to a DCC-specific interface of the WitsCABS application. They have the ability to add new clients to the database and can view active jobs.

Customers phone in to the DCC and provide their details, current location and destination. They are assigned a driver, told a likely departure time and given a cost for their journey.

1.4.2 Developers

Developers of the WitsCABS application are responsible for the front and back end implementation as well as documentation for the system. The developers are:

- Frederick Nieuwoudt (front end)
- Danielle Winter (front end)
- Stephen Friedman (back end)
- Sello Molele (back end)

Frederick Nieuwoudt and Danielle Winter implemented and documented the prototype front end for the system using *angular.js* and *Twitter Bootstrap*. A pair programming approach was used for the implementation.

Stephen Friedman implemented the back end prototype framework and documentation. Sello Molele contributed to the documentation.

1.4.3 Project Manager

The project manager for the WitsCABS project is Danielle Winter. Their responsibilities include organising the development team and managing and integrating the documentation for the project.

1.4.4 Scrum Master

The Scrum Master for the project is Frederick Nieuwoudt. Their responsibilities include managing the sprint backlog and sprint retrospective.

2 PART A: FRONT-END

2.1 SRS

2.2 SDS

2.3 Prototype Implementation

2.4 Product Backlog

Interface	Features	Expected Time (Days)
DCC	Active jobs panel	5
	Customer form panel	5
	Implement tabbed view (rather than panels)	1
	Automatic address completion using Google Maps API	5
	Review user experience design (aesthetics using CSS)	5
Driver	Assigned job panel	5
	Buttons for "On the way" and "Job complete"	1
	Automatic update of navigation	5
General	Log-in page	5
	Log-in authentication	5

2.5 Sprint Planning

2.6 Sprint Retrospective

3 PART B: BACK-END

3.1 SRS

3.2 SDS

3.3 Prototype Implementation

3.4 Product Backlog

3.5 Sprint Planning

3.6 Sprint Retrospective