



UNIVERSITY OF THE WITWATERSRAND
SCHOOL OF ELECTRICAL AND INFORMATION
ENGINEERING

PRIVATE BAG 3, 2050, JOHANNESBURG, SOUTH AFRICA

Taxi-cab Service Management System

Front-End:

Danielle Winter

Frederick Nieuwoudt

Back-End:

Stephen Friedman

Student Number:

563795

386372

360938

11 April 2016

Abstract

Contents

1	INTRODUCTION	1
1.1	Project Overview	1
1.2	Problem Statement	1
1.3	Project Objectives	1
1.4	Stakeholders	1
1.4.1	Users	1
1.4.2	Developers	2
1.4.3	Project Manager	3
1.4.4	Scrum Master	3
2	PART A: FRONT-END	4
2.1	SRS	4
2.2	SDS	5
2.3	Prototype Implementation	5
2.4	Product Backlog	5
2.5	Sprint Planning	5
2.6	Sprint Retrospective	6
3	PART B: BACK-END	7
3.1	SRS	7
3.2	SDS	7
3.3	Prototype Implementation	7
3.4	Product Backlog	7
3.5	Sprint Planning	7
3.6	Sprint Retrospective	7

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

The software requirements specification quantifies the scope and success criteria for this project.

The required elements of the front-end of the project as discussed in the SRS that were successfully implemented into the demonstrable prototype include:

- The log-in page
- The driver active job view
- The dispatch agent data entry form
- The dispatch agent active jobs view

With the inclusion of these modules into the prototype allows for the demonstration of the basic work-flow set out for the users of the application.

The modules that have yet to be implemented in future sprints that have been laid out in the SRS include:

- Adding GPS functionality to the application

The major downfalls according to the SRS was the lack of test driven development. This was specified in the SRS as the testing plan to be followed.

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
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
	Create basic interface framework	1
	Set up AngularJS customer directive	2
	Review user experience design (aesthetics using CSS)	5
	Set up google maps API	5

2.5 Sprint Planning

Sprint planning was undertaken to divide tasks in such a way that the most efficient path was taken.

In order to achieve this the task back log was represented on a gantt chart in order to visualize the product dependencies and which tasks could happen in parallel. The produced gantt char is available in the figure below.

The tasks were divided according to whether they were on the critical path or not. Creating the views for specific users was the most important for the success of the prototype. Tasks such and login authentication and the addition of Google maps functionality were not critical to the creation of a prototype that can visually demonstrate the final product.

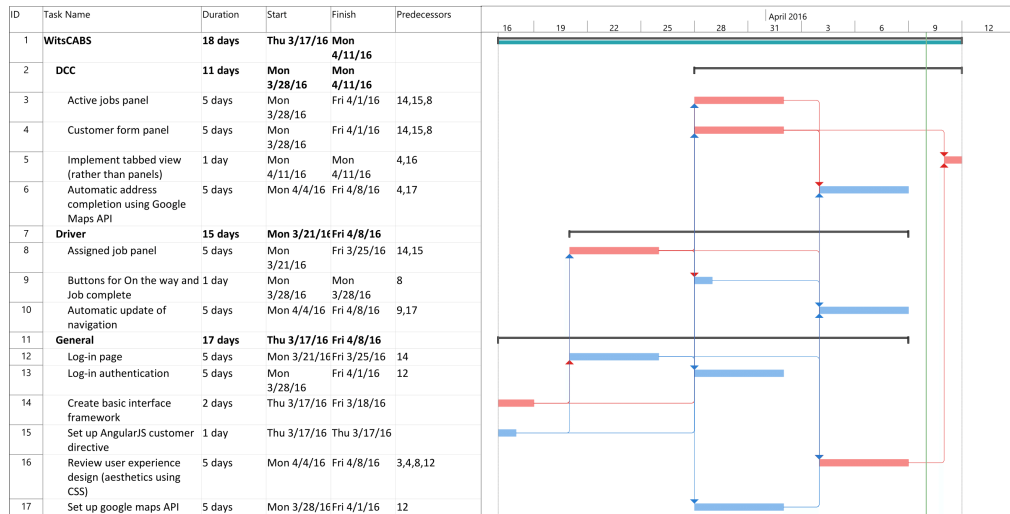


Figure 1: Gantt chart showing all required tasks and their dependencies.

2.6 Sprint Retrospective

Three sprints were completed to produce the presented front-end prototype. The features implemented in each sprint has been cataloged in the table below.

Sprint	Features
1	Create basic interface framework Set up AngularJS customer directive
2	Driver assigned job panel Create default login page Create login modal
3	Add DCC active jobs panel Add DCC customer form panel

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