

Software Requirements Specification for
WitsCABS
Lab 2 Submission - Group 10

1171733 - Jason Stuart (GL), 669006 - Rashad Akoodie,
1064934 - Robert Basson and 886515 - Amine Boukrout

August 29, 2017

Contents

1 Introduction 3

1.1 Purpose 3

1.2 Overview 3

2 Analysis 4

2.1 Team Administration 4

2.1.1 Team Management 4

2.1.2 Team Meetings 5

2.2 System Architecture Analysis 5

1 Introduction

1.1 Purpose

The purpose of this document is to describe what is required for the development of WITSCABS (a lift service similar to the one provided by Uber). This document is meant to convey how we have decided to develop our system and the functionality required for the first release. This will be achieved through a description of the scope of the software being designed, so that the project bounds are clearly defined, in order to prevent scope creep. The document will then for both the front and back ends describe their functional requirements. There will also be a description of the software required to be integrated with our software solution in order for it to be functional according to the requirements. In addition to this, a fully descriptive documentation of all programming techniques as well as software engineering techniques that we as a group will follow during development will be included. These techniques will be substantiated in order to provide a clear systematic approach to the complete design of our project. Furthermore, all resources consulted will be included as well.

1.2 Overview

The document will be broken up into multiple sections. As already mentioned before in the document, we have given an introduction as to why this document has been typed up, as well as a scope of what the project is all about and a basic description of the required features needed to be implemented.

In section 2, an analysis of how the project will be conducted is discussed. This includes a discussion of system architecture types, software required for the project, as well as how the team will be managed and handled during development.

In section 3, a list of formal required functions will be provided for both front-end software as well as the backend.

2 Analysis

2.1 Team Administration

2.1.1 Team Management

Our team has decided to take the Agile SCRUM method as our SDLC (software development life cycle). We chose this approach based on multiple reasons. Due to the short time constraint given to us by the client, a rapid development approach is needed, one that is satisfied by Agile development.

We also require quite a bit of feedback from the client, hence this iterative approach is best as it allows us to be in constant communication with the client, with small changes in between, allowing us to quickly change anything the client is unhappy with. Another advantage of SCRUM specifically, is that we will constantly be meeting up each day either in person, or by Google Hangouts to check up on the group and what each member has done over the last day, allowing us to keep a high morale and high sense of worth in the team, even though the time requirement is strictly short. Since our team is highly motivated already and have experience in development, they can be trusted to handle their tasks efficiently, another benefit of the SCRUM process, where each developer has more responsibility than in other methodologies. Also, since our team is relatively small, SCRUM will also work, as there will not be as much time required to structure and organize the team as a whole, increasing productivity. Also, since SCRUM focuses more on the actual development and less on the documentation, since we are going to be showing each change to the client, it benefits the productivity even more.

We have also considered the disadvantages of SCRUM. We understand that there is a higher chance of Scope creep with Scrum, but this is why we have already defined the scope of the project in the beginning of this document, to define clear boundaries to the project, to prevent this from happening. We have also been very clear with our functional requirements needed for a successful project, to prevent inaccurate measurement of time estimations.

Furthermore, SCRUM has the possibility of failure, if at least 1 member loses interest or focus. This is why we have nominated a clear leader to take charge and to keep the team motivated, regardless of what may come. If a team member were to leave, this would also have a large inverse effect on the project, however this will not happen, as each member is fully committed to this project, as this project results in marks required to graduate.

Finally, it may be hard to quantify quality as the software is rapidly evolving, however since we do not have a quality control team, we can work on quality between each developer to keep the project from slipping quality wise.

2.1.2 Team Meetings

The team will conduct the project as follows:

Each team member will be required to be at each sprint planning meeting every 2 weeks, from the start of the project. This meeting will be organized by the Group Leader, i.e. the scrum master.

Each member will be required to be available for the daily stand up meeting, preferably in person, otherwise via Google Hangouts. Each member should be able to describe to the team what they have worked on over the past 24 hours. Thereafter each member will describe what they will work on for the next 24 hours.

The scrum master will ensure that the project is moving at a reasonable pace and that all team members are motivated at all times.

2.2 System Architecture Analysis

After thorough analysis of existing architectures and analyzing the functional requirements listed later in this document, the team has decided to make use of a 2 part Client - 1 part Server architecture. The first client software will be the mobile app that allows the drivers of the WITSCABS service to receive communication from the system as to who the driver should pick up, etc.

The other client software will be run on a desktop in the call center, so that as a client calls in, the operator can input all the details necessary for the system to compute the best driver to use. The server will be the bridge between these two client software services, where the server determines the best driver for the job that arrives and pushes the required information through to the driver on his mobile app.

All this communication will be facilitated through an Internet connection, so when designing the system, it must take into consideration that an internet connection might not always be available for the drivers. There is no need to have any further server or client interfaces thereafter.

3 Functional Requirements

3.1 Back-End Server

Accept incoming connections from desktop applications, which contains details related to the passenger in need of transport.

Analyze which service zone the passenger is located in.

Run an algorithm to find the closest driver ready to pick up a passenger

Send a push message to this specific driver with the details of the passenger to fetch as well as their destination.

Allow drivers to mark if available or not via a status change

Be notified when a driver is close to pick up point via SMS.

Be notified when passenger is there at their destination, to automatically mark drivers as complete, i.e. the driver is transitioning back to their house or service taxi rank. Thereafter the driver can mark they are ready for another job.