Department of Computer Science

Project: NavUP

Client: Department of Computer Science,
University of Pretoria

Software Requirement Specification

Team Jelly

Author(s):	Student number(s):
Mpho Baloyi	14133670
David Seonin	15063021
Cian Steenkamp	15095682
Victor Twigge	10376802
Wanrick Willemse	29560617
Idrian van der Westhuizen	15078729

Contents

1 Introduction

1.1 Purpose

The purpose of the software requirements specification (SRS) document is to highlight the requirements and functionality of the NavUP system. The SRS describes the overall design of the system and its interfaces from the various interfaces and how they are linked with one another.

The documents contains details of what the product must be able to do and describes the various users that will make use of the NavUP system so that appropriate use interfaces can be designed. The details about the products assumptions and its dependencies are also contained within the document along with the constraints of the system, this is to allow for a better understanding of what must be done and what can be done within the system, along with what must be tried to avoid.

Further specific requirements are detailed within the document and makes mention of functional requirements and other design constraints to give a finer detail of what is expected in the NavUP system. The SRS documents is intended for the clientele that will oversee the creation of the NavUP system.

1.2 Scope

1.3 Definitions, Acronyms and Abbreviations

1.4 References

1.5 Overview

The SRS firstly gives an overall description of the NavUP system and its various interfaces. Each interface is divided into a separate subsection where further detail is given about what it must do and how it interacts with other interfaces. Afterwards the SRS makes mention of the memory constraints and operations. The requirements to the site adaptation are specified further in the SRS documentation.

The SRS describes the average expected user for NavUP the system and what constraints the developers need to take into account when designing the overall system in more detail. A list of assumptions and dependencies of the user and the overall system is given that were used to design the basic functionality of the NavUP system detailed in the SRS document.

Specific requirements are given for the external interface and functional requirements. The functional requirements contains smaller logical modules and how they might work. Performance requirements that describes how the NavUP system should perform and what is expected of its performance along with design constraints are given for the NavUP system. The SRS then describes the various quality attributes the system should have in order to function reliably.

2 Overall Description

2.1 Product Perspective

2.1.1 System interfaces

The system is going to be designed in a modular fashion, where the separate functionalities are broken up to allow for multiple programmers and designers to work on the NavUP at once. The modularity also allows for better maintenance and upgrading of future software and/or hardware by allowing the programmers to only change a smaller group of modules.

The System will need to be coded in such a way that multiple types of mobile devices would be able to use the NavUP system. It must also be able to communicate with an external database/server such as ClickUP where user information can be tracked and saved for further use in other applications and functionalities.

2.1.2 User interfaces

The user interface should be designed for a mobile device, in other words the screen should not be cluttered with icons and make use of touch screen technologies and its gestures. The map of the Hatfield campus along with various points of interest should be clearly visible on the screen. The user interface must be unambiguous since not only students and staff will make use of the NavUP system, but visitors as well. Since visitors will make use of the system, a way to locate and find various building by name would be beneficial, not only for visitors, but perhaps first year students as well.

2.1.3 Hardware interfaces

The NavUP system should be able to make use of the Wi-Fi routers scattered throughout the Hatfield campus. The application itself should be able to run

on a mobile device and therefore make use of phone data alongside the Wi-fi routers and make use of the built-in GPS system on most mobile devices. The NavUP application should be able to support input from touch screen devices from the user's mobile device to communicate and request various functionalities of the system.
The system should also make use of an external database to track a user's progress for various achievement based activities. The system would also be able to use the database to direct specific help/information to the user.

2.1.4 Software interfaces

The various classes and modules programmed on the software of the system should be capable of receiving data from the hardware and communications functions of the system. The software should be able to calculate and update values on the internal system as well as the external database. Various classes and modules should be able to send and receive values from one another and these updated/received values should be able to communicate with the mobiles devices interface in order to update the map. The software should also be capable of updating the data of the external database.

2.1.5 Communication Interfaces

The mobile device used by the user should be able to communicate with the Wi-Fi routers throughout the Hatfield campus in order to update values such as coordinates on the system. The user's mobile device should also be able to send and receive data from an external database, this data can also be used to block/allow access to certain features for instance a student must be able to participate in game like activities, but a visitor does not have to. Various mobile devices should be able to communicate with their navigation systems and other mobile devices (directly or indirectly) in order to calculate and create heat maps of high user traffic in an area on the map.

2.1.6 Memory

2.1.7 Operations

2.1.8 Site Adaptation Requirements

2.2 Product Functions

2.3 User Characteristics

2.4 Constraints

2.5 Assumption and Dependencies

3 Specific Requirements

3.1 External Interface Requirements

3.2 Functional Requirements

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software System Attributes

3.6 Other Requirements

4 Appendix