UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Cos301 : Software Requirements Specification for the NavUP System

Mufamadi, Khodani
14197520

Burgers, Heinrich
150595538

Cheriyan, Midhun
17308632

Cilliers, Joshua
14267196

Leshaba, Harris
15312144

Van Hattum, Jason
15027458

Rambani, Unarine
14004489

February 21, 2017

Contents

1 Introduction

...

1.1 Purpose

...

1.2 Scope

...

1.3 Definition, Acronyms, and Abbreviations

...

1.4 References

...

1.5 Overview

...

2 Overall Description

...

2.1 Product Perspective

...

2.1.1 System Interfaces

...

2.1.2 User Interfaces

...

2.1.3 Hardware Interfaces

- An android phone or tablet.
- An iPhone or iPad.

2.1.4 Software Interfaces

...

2.1.5 Communications Interfaces

- Cellular networks
- Global Positioning Satellites (GPS)
- Wireless networking (WiFi)
- E-mail

2.1.6 Memory

...

2.1.7 Operations

...

2.1.8 Site Application Requirements

...

2.2 Product Functions

...

2.3 User Characteristics

NavUP should have three user groups: A student or staff member, a administrator, and a guest user.

Students and staff members

- Students and staff members are registered, and have student numbers.
- Students are likely to be young (Below the age of 30)
- Students and staff members are likely to have a high level of education.
- Students and staff members should have a relatively high level of technical experience, and therefore be able to use and navigate a relatively complex app.

Guest Users

- Guests are unregistered.
- The technical level and education of a guest is unknown. It might be difficult for them to navigate a complicated interface.

Administrators

- Administrators should have a high level of technical expertise.
- Administrators likely have some form of identification, such as a student number.

2.4 Constraints

- **Cost** - We do not have the funds to pay for expensive libraries and tools.
- **Time** - Most of us are third year and honours students, and so we do not have much time to work on the project. Additionally, we only have one semester to do this.
- **Skills** - Our skills are varied, but mostly undeveloped, which limits the technical complexity of our solution.
- **Scope** - Our scope is defined as a navigation system for the University of Pretoria, and so our solution should be limited as such.

2.5 Assumptions and Dependencies

...

3 Specific Requirements

...

3.1 External Interface Requirements

System Interfaces

User Interfaces

Hardware Interfaces

1. **An android phone or tablet**
   Android will be the most common device used, and will likely be the only device we develop a solution for. There are android devices with many varying specs, but we will focus on newer models in order to simplify the prototype.
   Tablets are not used often for navigation, but due to the nature of Android the app will work there as well.

2. **An iPhone or iPad**
   It is possible but unlikely that we develop an app for iOS, as only a small percentage of staff and students use iPhones or iPads.

Software Interfaces

Communications Interfaces

1. **Cellular networks**
   We will use cellular networks for downloading and uploading data and information required to use the app, such as user credentials, heat-maps, and routing.

2. **Global Positioning Satellites (GPS)**
   We might use GPS to locate the device for heat-map generation and navigation.

3. **Wireless networking (WiFi)**
   WiFi will be used where possible to download and upload data similar to cellular networks. WiFi will also be used to locate the device similar to GPS, especially inside where GPS connection may be poor.

4. **E-mail**
   We may use email for registration and login, as well as passing information on to the user.
   Input: The user's email address.
   Output: An email to the user containing information.

3.2 Functional Requirements

...

3.3 Performance Requirements

The system should:

- Be interactive to user input within 2 seconds.

- Acquire the user's location with an accuracy within an 8m radius.
- Display the user's location on a map within 3 seconds after launching.
- Generate routes for users within 2 seconds.
- Update the user's location every second when navigating.
- Give a warning or indication if the processes take longer than anticipated.
- Display a message if an error is encountered.

3.4 Design Constraints

...

3.5 Software System attributes

...

3.6 Other Requirements

...