Software Requirements Specification
COS-301
The Inevitables

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Drew Langley	11039753
Lyle Nel	29562695
Dawie Pritchard	13104340
Peter Rayner	14001757
Hendrik Jan van der Merwe	15101283

Clients: Morkel Theunissen and Maria Ramaahlo

23 May 2017

Contents

Introduction

Many mobile applications exist to assist their users in successful travel from one location to another, by vehicle, or on foot.
Though these applications exist and work very well, solutions for disabled persons such as visual and physical impairments do not exist.

This document serves to formally introduce and quantify the Object Sensor and Mobile Navigation application, this project is part of the COS 301 (Software Engineering) Capstone project available for bid in 2017. This project was successfully tendered by TheInevitables.

Purpose

The purpose of this document is to provide a detailed description of the requirements presented by the Object Sensor and Mobile Navigation application, this application is intended to assist the students of the disability unit with navigation around the University of Pretoria. Additionally this document will include possible constraints and technical requirements such as how the system wil interact with external applications, such as the users.

This document is intended to comply with the IEEE 830-1998 SRS Standard. This document is also intended for use by clients Morkel Theunissen and Maria Ramaahlo for approval such that the system implementation may begin.

Scope

Definitions, Acronyms and Abbreviations

- IEEE - Institute of Electronic and Electrical Engineers.
- GPS - Global Positioning System.
- GIS - Geographic Information System.
- WiFi - Wireless Fidelity.
- WCAG - Web Accessibility Guidelines.
- AR Tag - An Augmented Reality Tag that is printable and facilitates physical tagging to be read using machine vision at a later stage.
- ArUco - A specific binary square fiducial marker implementation that that is the main candidate for AR Tagging in this project.
- Fiducial marker - A machine readable object in the physical world that is used to tag objects that are deemed relevant the navigation system.

From here on the DU@UP Object Sensor and Mobile Navigation application will be referred to as NavUP.

References

- IEEE 830-1998 SRS Standard - `https://standards.ieee.org/findstds/standard/830-1998.html`
- DU@UP Object Sensor and Mobile Navigation - document supplied by clients, before project tender.

Overview

These are the sections that follow;
The Overall Description, which describes the overall system, the modules and the interfaces that exist within the system, the system functions, the characteristics of the users, the systems' possible constraints and the assumptions and dependencies required for the system.
The Functional Requirements section lists the requirements the system should adhere to and describes use cases.
The Specific Requirements section is mainly intended for developers and is dedicated to defining the external interface requirements, functional requirements, performance requirements, design constraints, quality requirements and software system attributes.
The development of mobile communications allows for the development of applications for the independent navigation of persons with visual disabilities. The combination of mobile technologies, navigation systems and low tech devices will assist students to safely and successfully navigate their way on campus.

Overall Description

Product Perspective

Similar to existing navigation applications such as Google Maps and Waze, the NavUP is a standalone mobile application that acts as a campus navigation system for students both with and without disabilities. It encompasses many technologies which will be required to provide the requested and necessary functionality as described in the functional requirements section.

Obstacles described previously such as poles and low hanging roofs, intersections on campus(between IT Building and EB building) will be fitted with fiducial markers. These markers will be detected using machine vision and will be used to warn users when they are in proximity to certain obstacles or landmarks.
The system will use the user's smart device's hardware interfaces such as GPS to accurately get the user's location data, and navigate them to their required destination.

Product Functions

The NavUp system will support a variety of functions, the main function focuses on reliable, accurate navigation and presentation of geographical data, both indoors and outdoors. Other functionality includes providing information, and special routes for students with disabilities.

The navigational functionality will consist of getting the user's current destination accurately and then determining a suitable route to the venue the user requests.

Students with visual disabilities already have defined routes which the system must cater for.

Information provided to users will consist of venue information, geographical information, and information regarding the proximity of obstacles to visually impaired users.

Users will be given the fastest route to their destination, unless they are physically impaired, which results in a route that takes lifts and wheelchair access into account.

Due to the project focusing on the assistance of visually and physically impaired students and users, the application will conform to the WCAG to ensure these users with ease of use and accessibility.

The system will be developed to run on mobile devices, and will be acquired by users from the "Google Play Store".

The application will be perceivable, operable, reliable and follow the University of Pretoria's colour scheme and branding guidelines visually.

Normal users will have the option to notify the system of a potential hazard or obstacle to impaired students. This will be accomplished by allowing them to drop a location pin and will work as a crowdsourcing technique to identify and capture dynamic obstacles.

The application will also be secure to avoid theft or loss of data.

User Characteristics

The main category of individuals that will make use of NavUP will be designed for will be visually impaired students of the University of Pretoria.
The main body of students and staff will also have access to the application. These students will interact with the system via the mobile application. This means they will not require information regarding how the software works, however it is important to keep their information safe. They will also be required to have access to and knowledge of how to use a smart device.

Other users would include maintenance workers and administrative users, responsible for updating information pertaining to the application. These administrators will require knowledge about how the software works so as to maintain it correctly.

Constraints

The following is a list of possible design constraints related to NavUP:

- Security - The users personal information and current location should not be accessible to the public.
- Accuracy - The users location should be found whether the user is indoors or not. The location of the user should also be found in terms of which floor they are on in the building.
- Performance - The system should be able to handle a large amount of users making use of the software at the same time.
- Reliability - The application should still operate when one or more data access points are no longer available.
- Accessibility - The application should be easily accessible to everyone that requires it.
- Usability - The applications interface should be easy to use and understand.
- Size - The application should not require too much memory in order to operate.
- Users - The nature of some users enforces constraints such as conforming to the WCAG for accessibility issues.

Assumptions and Dependencies

The application will be developed to operate from a user's smart device, these smart devices usually have all the necessary hardware fitted. It is therefore assumed that all users of the NavUP will have access to one of these smart devices. It is then assumed that these devices have the required hardware dependencies and capabilities to run the application correctly. Such dependencies and hardware include WiFi, GPS and cellular network data connectivity.

It is also assumed that the users will be students of the university, thereby having access to the information regarding their venues and times is provided to them.
In terms of usability, it is assumed the users will have knowledge regarding how to use mobile devices and their applications, this includes activating voice controls and so on.

Assumptions relating to the system itself include the beliefs that the system will be kept up to date with the latest venue information and GIS data. It is also assumed that the fiducial markers will always be operational. These dependencies relate to the accuracy and reliability of the system and it is therefore assumed and recommended that a maintenance schedule be put in place.
A final assumption relating to continuous, reliable performance of the application is dependent on server capacity.

Functional Requirements

REQ1: Navigation

Provide navigation functionality.
This involves getting a users current position accurately and providing a route to the user's destination.

REQ2: Usability

The system should allow all kinds of users access to helpful navigational information. These users include students, visually and physically impaired students, staff and guests. This involves accessibility and usability issues.

REQ3: Provide Different Routes

Provide different routes based on the user's needs and preferences. This could be for physically or visually impaired users, or by a request for the fastest route taking stairs, ramps, lifts, wheelchair access and obstacles into consideration.

REQ4: Mobile deployment

The application should be developed so that it may be downloaded from the Google Play Store.

REQ5: User Profiles

The system should allow users and administrators login capability and maintain certain information regarding them securely.

REQ6: User Data Persistence

The system should allow users to send information to the system (drop a pin). Examples of such information include location data to allow us to crowdsource dynamic obstacles such as spillages or construction obstacles obscuring traffic or creating potential hazards.

REQ7: Accessibility

The application should cater for users with access difficulties and impairments. This involves voice recognition activation and compliance with the WCAG. Visually impaired users should be informed of impending obstacles to ensure safe navigation to their destination.

REQ8: Reliability

The application should operate reliably both indoors and outdoors.This is to ensure navigation to all locations correctly on campus.

REQ9: Integration with campus services

The application should integrate with campus services such as the security department or informational systems.

REQ10: Aesthetics

The application appearance should be aesthetically pleasing while conforming to the University of Pretoria's branding styles and colour scheme.

REQ11: Machine vision

Targeted toward the visually disabled users, the system will be able to read fiducial markers placed around campus. The marker will be read using the user's smartphone camera, thus the user must point the camera in a general forward direction to make use of this feature. Since the user might not always be aware of the exact orientation of his/her device, it is currently under consideration to provide haptic feedback to assist the user in keeping the camera oriented, which will lead to a greater chance of marker detection. The placement of fiducial markers should be placed on walls whenever possible but may be painted on floors where it is not possible. Walls are preferred for two reasons, namely, it is much more

cost effective since the marker may be printed on a sheet of paper, secondly, the marker has a greater chance of being detected, with an increase in detection range, when front facing, as opposed to being read at an angle, which is always the case with floor painted markers.

REQ12: Landmark notification

Targeted toward the visually disabled users and making use of machine vision as specified in the above mentioned functional requirement, the system will notify the user of nearby landmarks that are relevant to the user. Since visually disabled make use of landmarks to a large extent to navigate around campus, thus the intention of landmark notifications is to assist the user with self orientation in order to make the user more effective at moving around campus autonomously. The choice of landmarks and exact nature of the notification will be determined by consulting with Mr Juan Erwee, the technical officer at the disability unit, on a case by case basis.

REQ13: Hazard/obstacle notification

Targeted toward the visually disabled users and making use of machine vision as specified in the above mentioned functional requirement, the system will notify the user of nearby obstacles and hazards that are relevant to the user. The intention of hazard and obstacle notification is to assist the user in avoiding common hazards and obstacles he/she encounters due to a visual disability. The exact choice of hazard and obstacle as well as the manner in which the information will be conveyed will be determined by consulting with Mr Juan Erwee, the technical officer at the disability unit, on a case by case basis.

Use Cases

Use Case for REQ1: Navigate user to their desired location

1. Use Case ID: UC1
2. Precondition: User is running the NavUP application and enters the navigation module
3. Postcondition: Client device receives and displays navigational information to destination.
4. Actor-System interaction model:

Figure 1: Use Case Diagram - UC1 Navigate user to their desired location

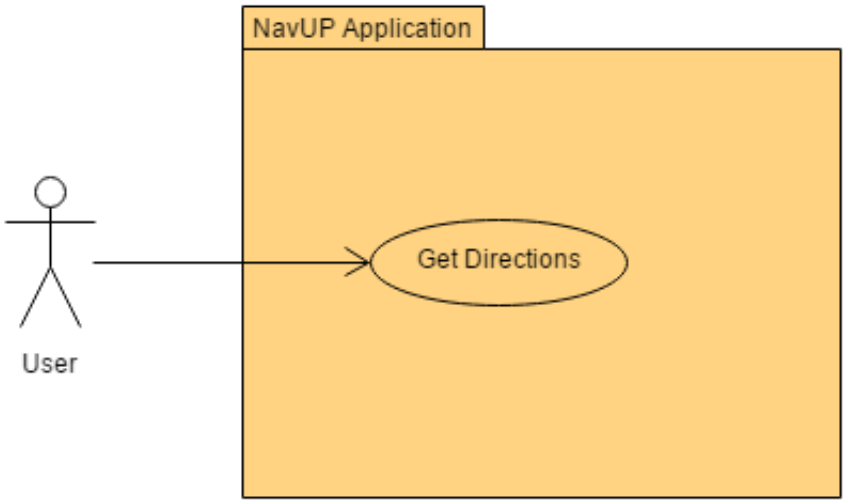

Table 1: UC1 - Navigate user to their desired location

Actor: User	**System: NavUP**
	0. The NavUP system displays the main window
1. The user clicks on the navigation button in the main menu	2. The system displays the navigation page to the user
3. The user clicks on the Get Directions button on the Navigation page	4. The system prompts the user for their required destination
5. User enters desired destination	6. The system calculates and displays the route and directions to the destination

Use Case for REQ1: Obtain Current Location

1. Use Case ID: UC2
2. Precondition: User is running the NavUP application and requires their location
3. Postcondition: Client device receives and displays the user's current location data.
4. Actor-System interaction model:

Figure 2: Use Case Diagram - UC2 Obtain Current Location

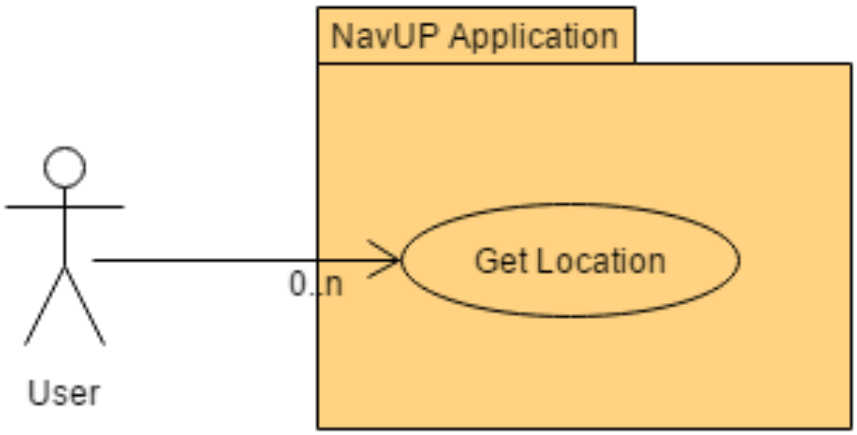

Table 2: UC2 - Obtain Current Location

Actor: User	**System: NavUP**
	0. The NavUP system displays the main window
1. The user clicks on the navigation button in the main menu	2. The system displays the navigation page to the user
3. The user clicks on the Get Location button on the Navigation page	4. The system displays the user's location

Use Case for REQ1: Search for Location

1. Use Case ID: UC3
2. Precondition: User is running the NavUP application and searches for a given location.
3. Postcondition: Information regarding the searched location is provided.
4. Actor-System interaction model:

Figure 3: Use Case Diagram - UC3 Search for location

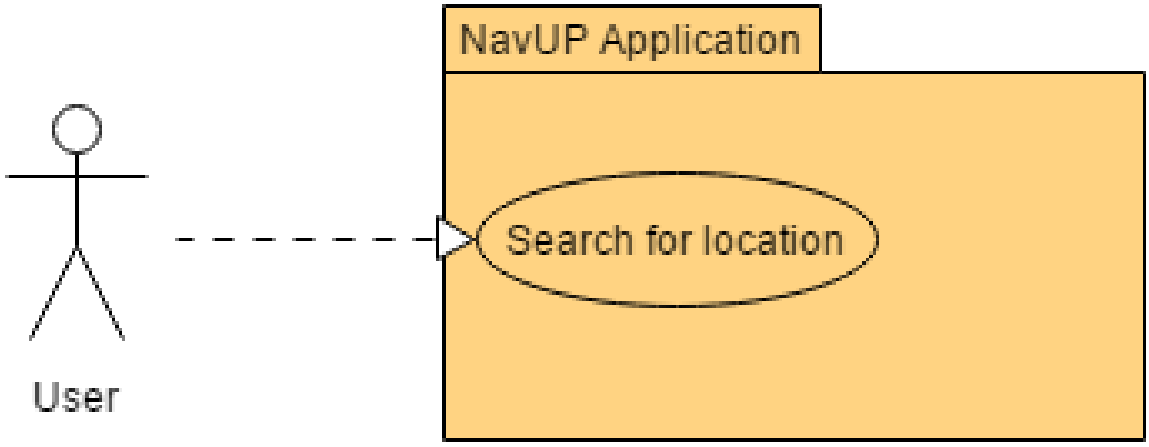

Table 3: UC3 - Search for location

Actor: User	**System: NavUP**
	0. The NavUP system displays the main window
1. The user clicks on the navigation button in the main menu	2. The system displays the navigation page to the user
3. The user clicks on the Search for Location button on the Navigation page	4. The system prompts the user for the location they wish to search
5. User enters desired location	6. The system provides information regarding the searched location

Use Case for REQ3: Provide special routes for users with disabilities

1. Use Case ID: UC4
2. Precondition: The user is running the application and requires a special route as they have a disability.
3. Postcondition: Client device recieves and displays information regarding the optimal navigational route for the special needs user.
4. Actor-System interaction model:

Figure 4: Use Case Diagram - UC4 Provide special routes for users with disabilities

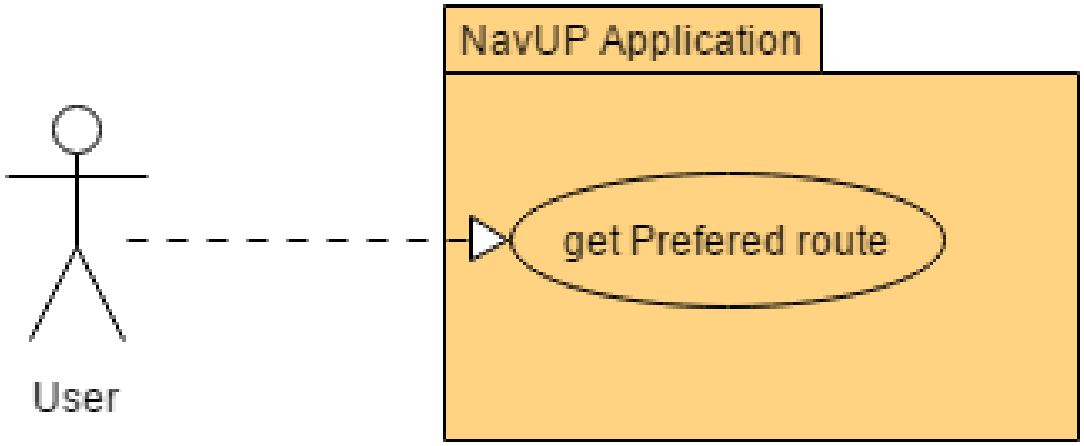

Table 4: UC4 - Provide special routes for users with disabilities

Actor: User	**System: NavUP**
	0. The NavUP system displays the main window
1. The user clicks the Routes button on the main menu	2. The system prompts the user for a destination
3. The user enters a destination	4. The system prompts the user for the type of route they require
5. The user selects the "special needs" option	6. The system displays the route and directions to the destination.

Use Case for REQ3: Provide the fastest route to a destination

1. Use Case ID: UC5
2. Precondition: The user running the application and requires the fastest route from one location to the next.
3. Postcondition: Client device recieves and displays information regarding the optimal navigational route.
4. Actor-System interaction model:

Figure 5: Use Case Diagram - UC5 Providethe fastest route to a destination

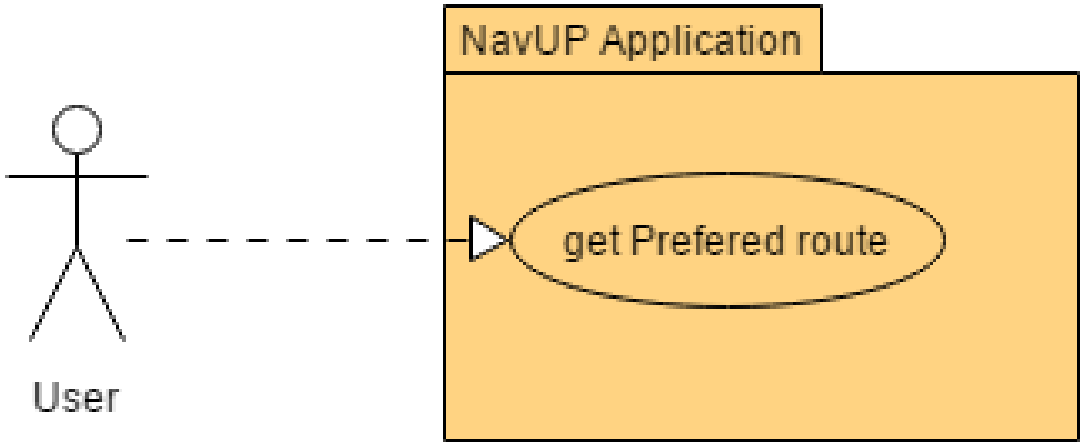

Table 5: UC5 - Provide the fastest route to a destination

Actor: User	**System: NavUP**
	0. The NavUP system displays the main window
1. The user clicks the Routes button on the main menu	2. The system prompts the user for a destination
3. The user enters a destination	4. The system prompts the user for the type of route they require
5. The user selects the "fastest path" option	6. The system displays the route and directions to the destination.

Use Case for REQ5: The user logs into the application

1. Use Case ID: UC6
2. Precondition: The user wishes to log into the application.
3. Postcondition: Client device logs the user into the system and displays the user's login page.
4. Actor-System interaction model:

Figure 6: Use Case Diagram - UC6 The user logs into the application

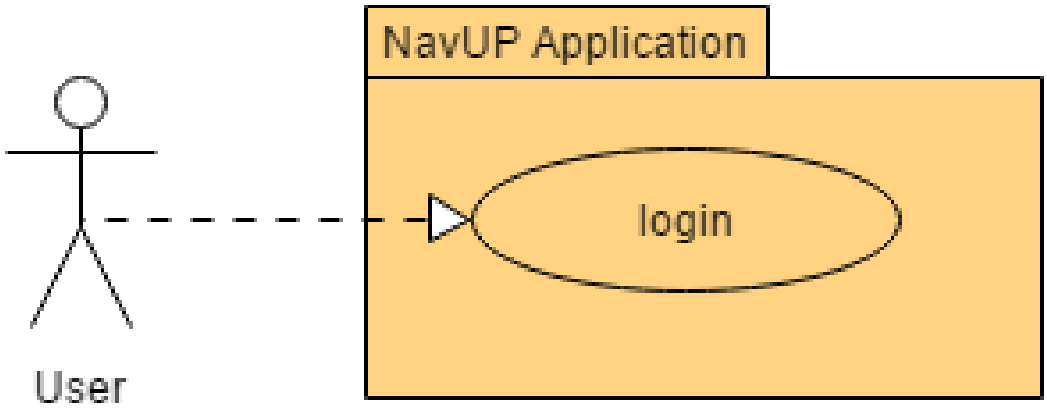

Table 6: UC6 - The user logs into the application

Actor: User	**System: NavUP**
	0. The NavUP system displays the main window
1. The user clicks the Login button on the main menu	2. The system prompts the user to enter the required login details
3. The user enters their login information	4. The system logs the user into the application and displays the user's user page

Use Case for REQ6: Allow users the ability to save a location(drop a pin)

1. Use Case ID: UC7
2. Precondition: The user is logged in and wishes to save their location.
3. Postcondition: Client device displays storage of a location by showing an icon over the specified position.
4. Actor-System interaction model:

Figure 7: Use Case Diagram - UC7 Allow users the ability to save a location(drop a pin)

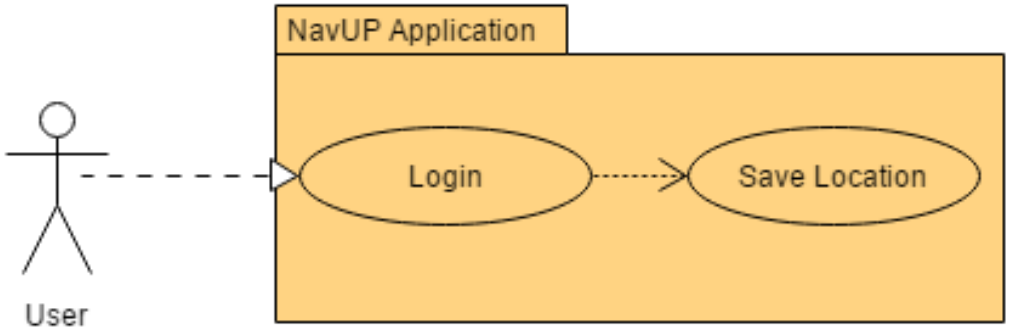

Table 7: UC7 - The user saves a location

Actor: User	**System: NavUP**
	0. The NavUP system displays the user's page.
1. The user clicks the Save Location button	2. The system prompts the user to enter a description for the saved location
3. The user enters a description for the saved location and confirms the action	4. The system saves the location and displays an icon over the location

Specific Requirements

External Interface Requirements

User Interfaces

The mobile application will interface with the supported input and output features of the host's operating system. Inputs include text that the user will enter for login or searching a venue. Outputs include the type of fonts to display text or graphics to show images or draw the map.

Hardware Interfaces

Since neither the mobile application nor the web portal have any designated hardware, it does not have any direct hardware interfaces. The WiFi software in the mobile phone manages the built-in WiFi and the hardware connection to the database server is managed by the underlying operating system on the mobile phone and the web server.

Software Interfaces

The final prototype will likely make use of the OpenCV API for machine vision, while consulting a standard PostgreSQL database system for persistent data such as GIS information. Some of these queries are triggered by interaction with the front-end user interface such as viewing the map and finding the shortest route, while others are automated such as fiducial marker identification.

Communication Interfaces

The communication between the different parts of the system are important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for both the mobile application and the back-end of the system.

Performance Requirements

Performance

- Offline activities should have a response time of +/- 2 seconds (instantaneous) when responding to an activity, while online activities such as calculating routes should have a response time of +/- 2-4 seconds so that the users have an uninterrupted experience.
- It should also allow the integration of a variety of services.

Reliability

- It should be able to handle +/- 50 000 users concurrently (simultaneously) when implemented into a suitable production environment.
- The application should be reliable, in that it will provide the fastest route every time without fail and complete all other computations successfully.
- All activities should be completed with a 0.1% error allowance.
- The application should provide accurate locations in a constantly changing environment.

Security

- Data transmission should be securely transmitted without unauthorized access, or loss of information.

Design Constraints

- Security - The users personal information and current location should not be accessible to the public.
- Accuracy - The users location should be found whether the user is indoors or not. The location of the user should also be found in terms of which floor they are on in the building.
- Performance - The system should be able to handle a large amount of users making use of the software at the same time, the system should also use resources on the client device efficiently.
- Reliability - The application should still operate when one or more data access points are no longer available.
- Accessibility - The application should be easily accessible to everyone that requires it.
- Usability - The applications interface should be easy to use and understand.
- Size - The application should not require too much memory in order to operate.
- Users - The nature of some users enforces constraints such as conforming to the WCAG for accessibility issues.
- Sensors - A built in digital camera, GPS and magnetometer is needed, all of which is standard in most smart phones.
- Platform - the system should be accessible on Android devices.
- Modularity - The system should be modular, thus allowing for high cohesion with low coupling and reducing the dependencies within the system.
- Aesthetics - The system's interfaces should be aesthetically pleasing and follow the UP branding guidelines.
- Fault tolerance - if unavoidable malfunctions occur,the design should be constrained such that the system can recover without loss of data or damage.

Software System Attributes

Reliability

- Any information that is stored on the database must remain correct when being transferred to the user interface.
- The services offered by the system should be available to users except for when the system is undergoing maintenance.
- The system should reply to user requests in the shortest time interval possible.
- The system must be fault tolerant, it needs to maintain a certain level of performance and offer other services that are not affected by this fault to the users.
- In the event of a fault the system must be able to recover within the shortest time period possible and recover any data that may have been lost.
- The system should be able to respond appropriately if it receives bad input data from the user.

Scalability

- The system must be able to cater for increases in the work load, for example large number of users or activities at any given time, without impacting the performance of the system.
- If the system does not cater for increases in workload it should at least provide the ability to be readily enlarged.

Maintainability

- The system must be designed in a modular fashion that provides high cohesion and loose coupling, this will allow parts of the system to be easily maintained without affecting the rest of the system.
- Maintenance should be able to be carried out by different maintenance teams, therefore the system must be easy to learn and understand.

Integrability

- Since we are following a modular design, components of the system that are separately developed should work correctly together.
- Follow coding standards specified by the client to allow for easy integration and employ continuous integration in our design process.

Usability

- The system must be easy to learn.
- System must cater for user mistakes, by providing the user with the undo or roll back options.
- The user interface must be easy to use and must be intuitive.
- The system should display options in a logical manner.
- Incorporate widgets and icons that the target users may be familiar with.
- The user manual should have a detailed description of the system.
- A help option must be provided to the users.