

# NavUP System

Bayanda Chakuma u11283212  
Keegan Ferrett u15132847  
Thomas Honiball u15348751  
Keamogetswe Matsobe u16087004  
Marin Peroski u13242475  
Lucian Sargeant u15225560  
John van Schalkwyk u14307317

February 24, 2017

## **Abstract**

Submitted in partial fulfillment of the requirements of COS301  
Software Engineering

## **1 Introduction**

Navigating through the University of Pretoria can be quite daunting for those who have never visited the institution before and for the first years alike. Lectures are spread across various buildings that contain a multitude of lecture halls. It definitely takes some time for a first year to become accustomed to the large ground space and especially when one's timetable is packed back to back with lectures. Technology can be used yet again to simplify life and lessen the tediousness of learning where locations are on campus. NavUP is a mobile application that will help with just that.

### **1.1 Purpose**

The purpose of this document is to provide a high level requirements specification of the NavUP navigational system. It will explain the basic purposes and features of the system as well as detailing the user interactions via modeling diagrams. This document is intended for stakeholders and developers of the system.

### **1.2 Scope of the Project**

The system will be a mobile application for use by all students and visitors of the university. The system will be designed to provide faster navigational routes around the University of Pretoria Campus grounds by selecting shortest routes and accounting for pedestrian traffic density. By providing an intuitive interface and tools to automate the tasks of predictive route plotting the system will satisfy the users navigational needs while remaining user friendly. Specifically this system is designed to allow users to select a location to navigate to. The software will facilitate navigation by analysing crowd gathered

locations from other application users along with wifi access points and signal strength to pinpoint the users location on campus grounds. This data is then used to calculate pedestrian traffic around campus and calculate the fastest route to the destination. The system also provides predictive location suggestions by gathering usage data if the user allows it too.

### 1.3 Glossary

Term	Definition
Crowdsourcing	The use of signals from multiple application users in order to gather datasets efficiently.
Heat Map	A representation of the density of people in a specified area.
Mobile Application	A software system developed to run specifically on mobile devices with sufficient software capability.
NavUP	A navigational system proposed within this System Requirements Specification document. An abbreviation of Navigation University of Pretoria.
Offline	A state indicating a device lacks connection to the internet.
Online	A state indicating a device has connection to the internet.
Profiling	Collecting data related to a person in order to form predictive formulae based on their behavioural patterns and routines
Pedestrian	An individual that travels without the aid of a vehicle along pathways
Reward	Incentive provided upon reaching a set goal
Software Requirements Specification	A document describing the requirements and constraints of a system. For example, this document.
Stakeholder	Any person with an interest in the project that is not a developer.
User	Any individual using the mobile application in accordance with its purpose.

### 1.4 References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.

### 1.5 Overview

The next chapter of this document will give an overview of the products functionality. It details informal requirements for the system and establishes a context for the technical requirements specification.

The third chapter of this document is the Requirements Specification. It is written primarily for the developers and it will describe the various details of the product in technical terms.

These sections are intended for different audiences, but both refer to the same mobile application.

## **2 Overall Description**

### **2.1 Product Perspective**

The system consists of the End Users which can be students, lecturers and visitors of the University of Pretoria, all of which function as the same actor in the system, faculties and organizations on campus, which submit activities to the system. Thus the system consists of a total of 2 types of actors.

## 2.2 Product Functions

### 2.2.1 UC1 - Open Application Use Case

Diagram:

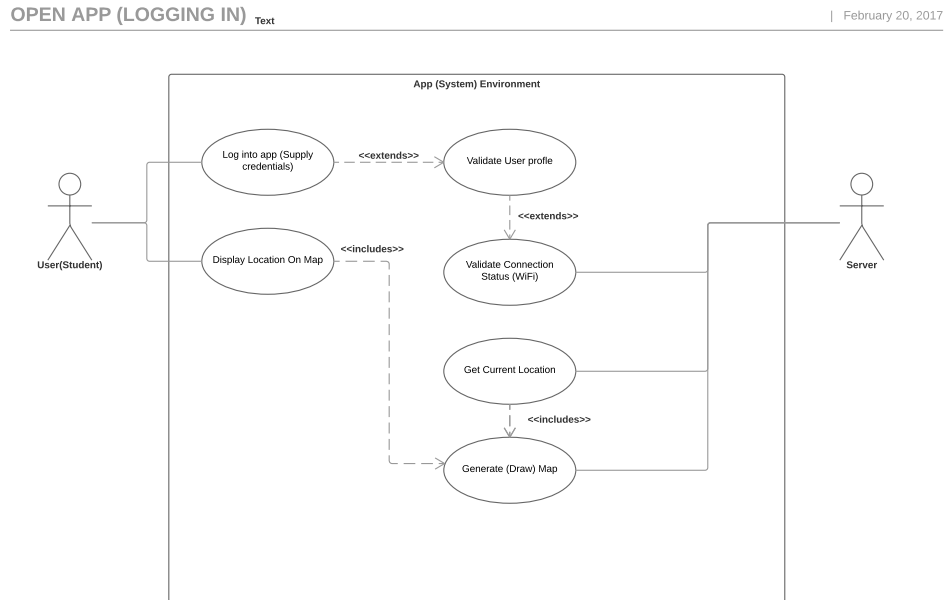


Figure 1: UC1 - Open Application Use Case Model

#### Brief Description

User opens the app and is presented with a map of campus honed in on their current location as well as a search bar at the top to indicate where they would like to go.

#### Initial Step-By-Step Description

1. Map of the university is drawn. Location of user is gotten using GPS and/or WiFi and displayed on the map.

## 2.2.2 UC2 - Location Search Use Case

Diagram:

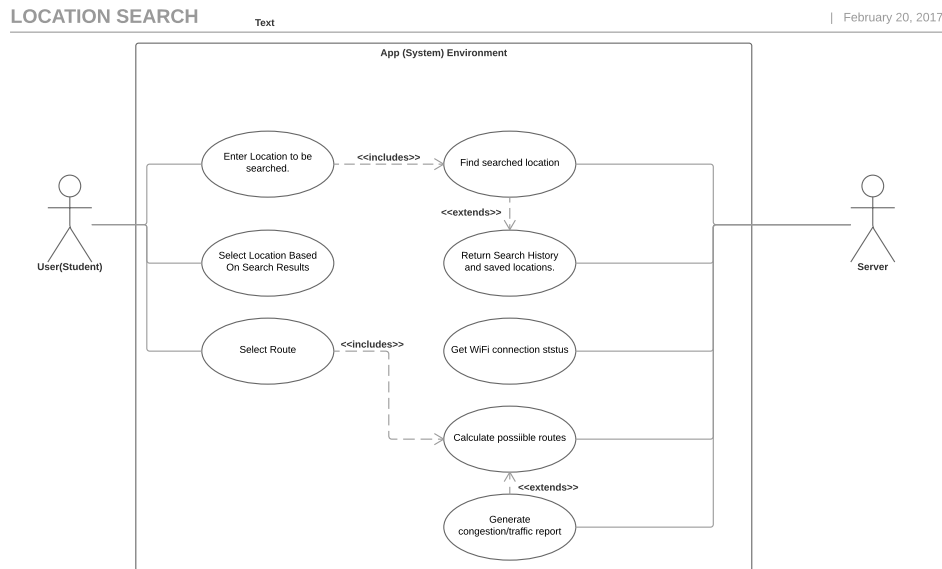


Figure 2: UC2-Location Search Use Case Model

### Brief Description

User either types in the name of the location or selects it on the map by placing a “pin”. The user is then presented with routes to the destination if the destination exists.

### Initial Step-By-Step Description

1. System displays drop down list of recently visited locations and suggested locations, as well as a search bar.
2. User either selects a choice from the drop down list or searches for another location.
3. System searches for location in database. If location is not found, report back to user and end user case, else, display location on university map and provide user with further options.

### 2.2.3 UC3 - User Input Use Case

#### Brief Description

The user can add information that describes what he or she is interested in with regards to activities or pastime events so that points of interest can be highlighted to them as they walk through campus.

#### Initial Step-By-Step Description

1. System displays form with options of what the user may be interested in.
2. User selects options from the form and then submits the results.
3. System analyzes the given information and come up with points of interest to display to the user when the user is deciding on a destination.

### 2.2.4 UC4 - User Timetable Use Case

#### Diagram:

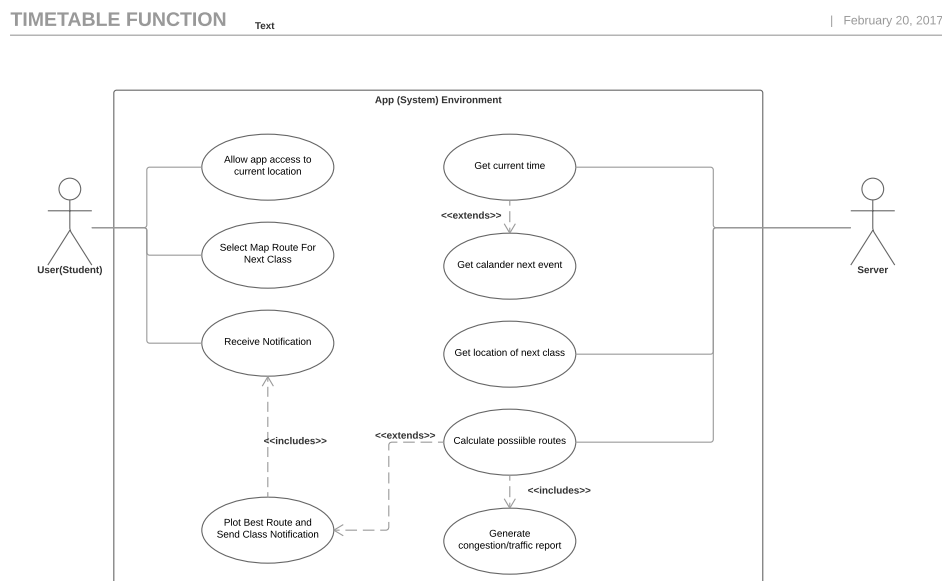


Figure 3: UC4 - User Timetable Use Case Model

#### Brief Description

The timetable can also be added by the user so that the route can automatically be determined based on his or her schedule.

#### Initial Step-By-Step Description

1. User selects option to add timetable

2. System displays a form to allow the user to submit a new class to their timetable.
3. User enters information about the class, such as module code, location, and time and then submits the information.
4. System adds class to user database and then refreshes the form for a new class.
5. User and system repeat steps 2 and 3 until all classes have been entered.
6. System can now provide a shortcut option when the user is searching for the location of their classes.

### **2.2.5 UC5 - Optimal Route and Traveling Use Case**

#### **Brief Description**

The user has requested a route to a destination and is presented with multiple options based on campus congestion. These route options will provide estimated time of arrival for each one allowing the user to make an informed decision. The optimal route will be shown and map changed based on whether the user is indoors or outdoors.

#### **Initial Step-By-Step Description**

Before this Use Case can be initiated the User must have requested a destination as per UC2

1. System displays multiple routes on the university map along with additional information about the route.
2. User selects a given route to the destination.
3. System displays chosen route, instructions and estimated time to reach selected destination.
4. User begins along route.
5. System refreshes user location and reflect the location on the university map. Update estimated time based on user speed.
6. System relays next instruction.
7. User follows an instruction given by the application.
8. Repeat steps 5-7 until UC-8 occurs or navigation is aborted.

### **2.2.6 UC6 - Activities Use Case**

#### **Brief Description**

User selects a route and the estimated number of steps is added to their user profile in a game like fashion leading to possible rewards awarded by third party members.

#### **Initial Step-By-Step Description**

User must have initiated and completed UC2 or have initiated UC5 up to step 3

1. System adds estimated number of steps to the user profile. If user is eligible, system presents a reward to the user.

2. If reward is presented, the user can collect/store the reward.
3. System records the state of the user's reward (stored/ used).

### **2.2.7 UC7 - User Based Augmented Reality Use Case**

#### **Brief Description**

As the user travels along their route and passes certain landmarks they will be presented with certain information about these landmarks. For example if they pass the music department they will be given the option to see upcoming performances. This will be based on the interests that they have specified on their profiles.

#### **Initial Step-By-Step Description**

Before this Use Case can be initiated the user must have accepted the terms and conditions related to data usage and must be near a landmark

1. Systems tracks user location as they are on route to their destination. If user gets near a possible point of interest, notify them and provide information about the landmark.
2. User acknowledges the notification. If events are available at the landmark the user can select any to be added to a calendar.
3. If any event was selected system adds it to the users calendar.

### **2.2.8 UC8 - End Point Use Case**

#### **Brief Description**

Once the user reaches their destination navigation will stop until the user enters in a new location or the location is pulled from their timetable information.

#### **Initial Step-By-Step Description**

This Use Case cannot be initiated unless UC5 has completed immediately prior

1. System realises user has reached the destination when users location is close enough to the location of the entered destination. Displays information about the trip, such as duration, distance travelled, average speed, etc.
2. User acknowledges that they have arrived at the destination.
3. System UI resets and awaits a new destination from the user.

## **3 Specific Requirements**

### **3.1 Functionality**

#### **3.1.1 Functional**

1. R1 - Basic User Profiling without violating user's personal information
  - The system shall allow users to create a profile and set credentials
  - The system shall authenticate user credentials to view profile



- The system shall allow the user to update profile
  - The system shall display frequently searched destinations by the user in the profile
  - The system shall allow the user to register for push notifications
2. R2 - Indoor Functionality
- The system shall provide a map of the current floor the user is on.
  - The system shall provide the user's location within the floor.
  - The system shall allow the user to specify a classroom on the floor.
3. R3 - Outdoor Functionality
- The System shall provide the location of the user on campus
  - The system shall provide building identification
4. R4 -Heat Map Signatures of Crowds
- The system shall allow the user to view possible traffic and crowded congested areas
5. R5 -Detailed map of the university (both indoor maps and outdoor maps)
- The system shall allow the user to view a detailed map of the University
6. R7 -Ability to direct user to specific class rooms and buildings
- The system shall enable the user to enter search text (destination) on the screen
  - The system shall allow the user to select preferred route to destination
  - The system shall notify the user when there is no matching search result
7. R8 -Should use quickest path (least congested)
- The system shall calculate the optimum path to get to the destination
8. R9 -Provide information on any local landmarks or events
- The system shall notify the user of any activities happening on campus
9. R10 -Use WiFi as well as GPS to determine location
- Full functionality and performance is determined by the use of Wi-Fi and GPS
  - Response time for loading the NavUP will be determined by the internet strength
10. R11 -Be able to determine height location in buildings
- The system shall enable the user to view their current floor-level height in buildings
11. R14 - Provide a tour around campus
- The system shall provide a basic tour of the campus grounds, covering all the important buildings and landmarks.

## 3.2 Non-Functional

1. R6 -Accurate directional navigation both indoors and outdoors
  - The system will allow the user to set destination
  - The system shall allow the use of icons and tool bars
  - The system shall direct the user to the correct destination
2. R12 -When not connected to network have basic offline functionality
  - The system will enable the user to view profile
  - The system will allow the user to receive push notifications
3. R13 -Third party reward system for performing certain actions

## 3.3 Performance

The NavUp App has to be downloaded from Playstore or iTunes to run on smartphone devices. The devices have to be connected to Tuks Wi-fi for the user to access and use the application. The Wi-Fi internet connection strength will determine the efficiency of the app.

## 3.4 Design Constraints

### 3.4.1 Navigation Application Development

The application is to be built using a webpage based on IDE (Intergrated Developmet Environment)

### 3.4.2 Navigation Application

1. The user is required to have basic computer skills to understand and use the App
2. The App must allow the user access it easily without violating users personal information
3. Response time for functions of the App should not take long

## 3.5 Software System Attributes

- Reliability
  - NavUP App should be reliable in terms of returning accurate results.
- Availability
  - NavUP App should always be available when used, exceptions are when there is a network failure.
  - For the App to be available it should always be connected to Tuks Wi-Fi.
  - App should be connected to GPS to get the user's location, calculate the destination and determine shorted route.

- Security
  - Users input information should not be easily identifiable if the system is compromised.
  - The System should limit user details to essentials needed for the app to work.
- Maintainability
  - The code of the Application should be written in a way to accommodate the future implementation functions.
  - Test Environment should be built to test different functions of the application.
- Portability
  - NavUp App should be portable with Android and iOS.

### **3.6 Other Requirements**

1. R2 -Indoor functionality
  - (a) Detect User's Location
  - (b) Find routes
2. R3 -Outdoor functionality

## 4 Requirements Traceability Matrix

Table 1: Traceability Matrix

<b>Requirement</b>	<b>Use Cases</b>							
	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8
R1			X	X		X	X	
R2	X	X		X	X		X	X
R3	X	X		X	X		X	X
R4					X			X
R5	X	X		X	X			X
R6					X			X
R7				X	X			X
R8					X			X
R9			X				X	
R10	X				X		X	X
R11	X	X			X			X
R12		X						
R13						X		

## 5 User-System Modelling

### UC1 - Open Application

Actor	System
	0. Map of the university is drawn. Location of user is gotten using GPS and/or WiFi and displayed on the map.

### UC2 - Location Search

Actor	System
	0. Drop down list of recently visited locations and suggested locations, as well as a search bar.
1. User either selects a choice from the drop down list or searches for another location.	2. Searches for location in database. If location is not found, report back to user and end user case, else, display location on university map and provide user with further options.

### UC3 - User Input

Actor	System
	0. Displays a form to allow the user to submit a new class to their timetable.
1. Enters information about the class, such as module code, location, and time and then submits the information.	2. Adds class to user database and then refreshes the form for a new class.
3. Repeats process until all classes have been entered.	4. Can now provide a shortcut option when the user is searching for the location of their classes.

#### UC4 - User Timetable

Actor	System
	0. Displays a form to allow the user to submit a new class to their timetable.
1. Enters information about the class, such as module code, location, and time and then submits the information.	2. Adds class to user database and then refreshes the form for a new class.
3. Repeats process until all classes have been entered.	4. Can now provide a shortcut option when the user is searching for the location of their classes.

#### UC5 - Optimal Route and Traveling

Actor	System
	0. System displays multiple routes on the university map along with additional information about the route.
1. User selects a given route to the destination.	2. Displays chosen route, instructions and estimated time to reach selected destination.
3. Begins along route.	4. Refresh user location and reflect the location on the university map. Update estimated time based on user speed.
5. Follows an instruction given by the application.	6. Refresh user information and display next instruction.

#### UC6 - Activities

Actor	System
	0. Estimated number of steps is added to the user profile. If user is eligible, system presents a reward to the user.
1. If reward is presented, the user can collect/store the reward.	2. Record the state of the user's reward (stored/ used).

#### UC7 - User Based Augmented Reality

Actor	System
	0. Tracks user location as the are on route to their destination. If user gets near a possible point of interest, notify them and provide information about the landmark.
1. Acknowledges the notification. If events are available at the landmark the user can select any to be added to a calendar.	2. If any event was selected added it to the users calendar.

#### UC8 - End Point

Actor	System
	0. Realises user has reached the destination when users location is close enough to the location of the entered destination. Displays information about the trip, such as duration, distance travelled, average speed, etc.
1. User acknowledges that they have arrived at the destination.	2. UI resets and awaits a new destination from the user.

## 6 Logical Module Grouping

Functional Cluster	Functional Description	System Requirements	Functional Subsystem Identified
Destination Information	User input destination through application, detailed description on how to get to that destination with provided map	R2, R3, R5, R6, R7, R8, R10, R11, R12	Accurate mapping of the campus and the interiors of its buildings. Exit and entry points Labelled
Location Services	Use multiple campus Wi-Fi connections to determine current location for directions to be provided to destination	R2, R3, R5, R6, R7, R8, R10, R11, R14	Wi-Fi connections along campus to determine current position and course needed to take to reach end destination
User Profiling	User entries of timetable or hobbies and activities that the user is interested in	R1, R9	Information provided by user
Activities Log	Third parties that rewards user for completing tasks or events based on fitness goals.	R13, R14	All third party events and user interactions
Optimal Route	Using heat maps to avoid crowds and congested areas .Determining easiest way to reach the destination from user input	R2, R3, R4, R5, R6, R7, R8, R10, R11	Wi-Fi connections along with heat map updates to determine shortest route