UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Faculty of Engineering, Built Environment and Information Technology

Khaki Round1

Team Khaki

Requirements Specification

Kulani Bamuza	15008402
Frederick Ehlers	11061112
Dimpho Mahoko	15175091
Katlego Mogokonyane	12134229
Maria Qumayo	29461775
Craig van Heerden	15029779
Linda Zwane	14199468

Stakeholders

Computer Science Department of University of Pretoria	Vreda Pieterse

Contents

1 Introduction

The introduction of the Software Requirements Specification provides an overview of the entire specification with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to define the problem in detail and provide the detailed requirements for NavUP.

1.1 Purpose

The purpose of this SRS document is to provide a detailed description of NavUP by collecting and analyzing the ideas that define the system. This document describes NavUPs user interface, External Interface, functional, and performance requirements. The document also describes the users of NavUp and its functions. The document helps developers of the NavUp system in software delivery lifecycle processes.

1.2 Scope

The product as mentioned before is called NavUP, nav being an abbreviation for navigation and UP is an acronym for University of Pretoria. The product should be available on all major mobile operating systems to ensure most users can use the product. The basic functionality of the product should be similar to the basic functionalities of navigation systems like Google Maps and Waze. It should be able to provide the user with their current location, it should be able to to search for locations and venues, it should be able to provide the user with navigation to a location or venue, and it should be able to save locations or venues.

The system must be able to provide the user their location outdoors as well as indoors. GPS will therefore not suffice because the GPS receiver will not be able to receive a signal indoors. The system will therefore only use Wi-Fi and crowdsourcing to determine the user's location.

The system should also have different levels of users, users with higher levels should be able to add new locations into the system. These locations can include points of interest, events and activities.

The system should be able to give the user information about how busy certain areas of the campus are. This can shown to the user visually through a heat map.

The system should also give users notifications based on their current location like points of interest. The system can learn what type of locations the user likes based on their previous locations and suggest them new locations to visit. The system should also record the user's movement data and reward them in a game like fashion.

The intended users are all students, staff and guests of the University of Pretoria. They should be able to navigate around campus, shown heat maps and get notifications based on their locations.

Table 1: Definitions, Acronyms, and Abbreviations

OS	Operating System:system software that manages computer hardware and software resources
IOS	an operating system used for mobile devices manufactured by Apple Inc.
IDE	integrated development environment,is a software application that provides facilities to computer programmers for software development.
MYSQL	is an open source relational database management system.
API	application programming interface,a set of functions and procedures that allow the creation of applications which access the features or data of an operating system or application.
Database	is a collection of information that is organized so that it can be easily accessed, managed and updated. Data is organized into rows, columns and tables, and it is indexed to make it easier to find relevant information.
Android SDK	Android software development kit.
Java	a general-purpose computer programming language designed to produce programs that will run on any computer system.
XML	a metalanguage which allows users to define their own customized markup languages, especially in order to display documents on the Internet.
AJAX	is a client-side script that communicates to and from a server/database without the need for a postback or a complete page refresh.
HTML	Hypertext Markup Language, a standardized system for tagging text files to achieve font, colour, graphic, and hyperlink effects on World Wide Web pages.
Javascript	is a high-level, dynamic, and interpreted programming language.

1.3 Definitions, Acronyms, and Abbreviations

1.4 Overview

The remaining sections of this document describes the context of the product, summary of the products functions, describes the characteristics of the users, outlines the restrictions of the solution space, lists the factors that affect the requirements, and it describes the software requirements including external interface requirements, functional requirements, and performance requirements. Section 2 provides an overview of the product. Section 3 provides a detailed description for each of the system interfaces, provides a detailed description of the products functionality, describes all the performance related capabilities of the product and outlines all the restrictions.

2 Overall Description

2.1 Product Perspective

As our scope given above details the requirements, the system will comprise of several inter-dependent modules viz, Heat map, Points of interest, Push information, Record user preference, Event driven, Navigation to locations/venues, Database, Search locations/venues and lastly (Graphical User Interphase)GUI. Aforementioned are the modules that will collectively form a system of inter-dependent sub-systems with low coupling and high cohesion. Each module will be elaborated in subsection(s) to follow and how they link up with each other to form a system.

2.1.1 Hardware interface

Minimum Requirements Hard drive space 56MB.

2.1.2 Software interface

Client on application The client must have Android 4.1 OS,IOS OS 7 and up installed on their phone/tablet, so they can run NavUP application on their phone

Database server We will use MYSQL to read and write information to our database.

Development end We will use Google Android SDK 1.0 to build and deploy the android application.Google Maps will make use of the Google Maps API for custom styling of maps.Our own map grid API to interact with google maps.IDE of choice with Java, XML, AJAX, Javascript, HTML,OS (windows, UNIX).

2.1.3 Communication interface

FTP(File Transfer Protocol)- The NavUp system will FTP to communicate with Google Maps. HTTP/HTTPS(Hypertext Transfer Protocol/ Hypertext Transfer Protocol over Secure Socket layer) to communicate with the user

2.2 Product Functions

The application will be able to track a users location. The user will search for a venue on campus by room number and be directed to that venue based on distance, time and crowd traffic. The app can also add a waypoint if the user wants to make a stop before the final destination. The application will also track the users steps and notify and reward them through a third party when they reach milestones. NavUP will also provide the ability for higher level users to push notifications to users if there are any events that user might be interested in. If the user isnt interested, they can block the notification resulting in the application learning their preferences.

2.2.1 Heat map module

This piece of module will be responsible for statistical computing of number of people/devices active at a certain or across Wi-Fi access point in either two forms (a) stationary participants and (b) on the move participants. The module upon implementation will visualize the concentration of connected devices to the Wi-Fi on GUI and thus tell us the little we want to know about pedestrian traffic.

2.2.2 Points of interest module

Module will retrieve a number of points of interest in the campus that will obviously need to be pre-loaded in the database and/or be loaded by users with special privileges to the database. The module upon implementation will take in a search of string type from a GUI entered by a user who will be searching for points of interest(s) in the campus and display the results if found to the GUI.

2.2.3 Push information module

The module will be designed in a way that it learns what the user prefers that can be either what the user likes searching or the location where the user spends most of his/her time at. This module upon implementation will predict what the user might be interested in based on the historical searches and locations and push that information to the user automatically, reason being for the user to be aware of something similar to what he/she might be interested in.

2.2.4 Record user preference module

This module will be recording top (10-15) frequently looked up (words, characters, numbers or strings) from the user. This module upon implementation will provide a dropdown like list of suggested words or phrases.

2.2.5 Event driven module

Interestingly this module will provide the user with Did you know or Alert pop ups when connected to the Wi-Fi giving a bit of information that the user wasnt aware of, or the user did not find from the database. This module upon implementation will make balloon pop ups in the GUI either giving the user tips and information with dates and alerts to keep users away from danger.

2.2.6 Navigation to locations/venues module

What this module does is to pinpoint the longitude and latitude of the user and wait for the final destination location to be entered by the user then that is when the user will be given directions on map reduced to campus scale or direction by text (i.e turn left.. turn right after few meters). This module upon implementation will make use of vector data points to draw navigation line on a map and the user has to follow or make a GUI window filled with text directions in sequence format.

2.2.7 Database module

The module will be mainly used to perform database queries like store, retrieve, delete and update. This module upon implementation will be in the backend accessing the database performing basic adding, retrieving, deleting and updating operations to the database.

2.2.8 Search locations/venues module

Module will be responsible for decoding character types from the GUI and retrieve matching strings from the database together with the information they hold or representing, and even go further to perform (80 percent) character matching mechanism for better search in case the user is not sure about the spelling.

2.2.9 Graphical User Interface (GUI) module

Module will handle display functionality, input functionality, and output functionality of all the information generated by above given modules. GUI makes it easier for a user to interact with the system in such a way that it will provide a display window for map, buttons to press, search box for a user to type in item(s) to be searched. Module upon implementation will be in graphical format, that supports various operating systems.

2.3 User characteristics

There are 4 user profiles available on the application. A guest user who is a user that isnt a frequent campus visitor. For instance a visitor to a staff member, someone coming for a conference or meeting. This user will only be able to access a limited amount of functionality like location services and step counting.

A registered user would typically be a staff member or a student. Someone whos on campus frequently and they will be able to access most of the functionality excluding higher level and third party functionality. So all the guest functionality with the added ability to add favourite venues and get rewards based on the amount of steps theyve taken relative to time. They would also receive notifications based on their preferences.

Third party user will be used for registered user reward givers like stores in and around campus. They will be able to indicate what rewards they have available, the system will reward at its own discretion and the third party will receive a way to validate which user gets what reward.

A higher level user would be an organization like a society, a political party, a faculty or another authorised party. They have Third party user rights with the added ability to add events to the pushed notifications and add new venues onto the app.

2.4 Constraints

The application will not be able to fully function without a connection to the internet Because it will need to extract data from the server. An offline user will only be able to use a limited number of things.

The smart device being used to access the application must have internet connectivity, a GPS System and WiFi.

The application is going to be constrained by the fact that it needs to access the devices GPS system. The fact that there are different devices with different manufacturers, the Interface between the different devices will differ.

2.5 Assumptions and dependencies

It is assumed that the applications will be used on either a smartphone or a smart tablet with location services, WiFi and mobile data enabled. The device is assumed to have enough memory to handle the application It is also assumed that the accelerometer hardware is installed on the device and well in function. It is assumed that your phone has enough battery to handle the application throughout the day The applications performance is dependent on that of the device. The accuracy of the step counter is dependent on the device accelerometer. The accuracy of the location tracking is dependent on the device GPS.

3 Specific Requirements

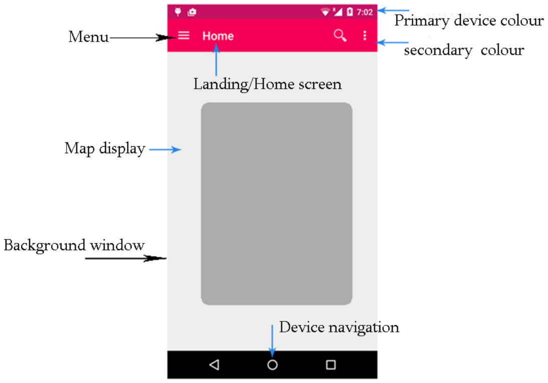
3.1 External Interface Requirements

3.1.1 User interfaces

The NavUp interface will be useful, easy to use and a delightful to interact with. This will enhance the user experience people will have while interacting with it. This will also ensure that value is given to the application.

This will be achieved by making use of Google's Material Design (GMD) the elements that have already been placed and created to be used. Each element is set to standards that have many dimensions, and includes different disciplines-such as interaction design, information architecture, visual design, usability and human-computer interaction.

The image below shows a prototype of a possible home screen.

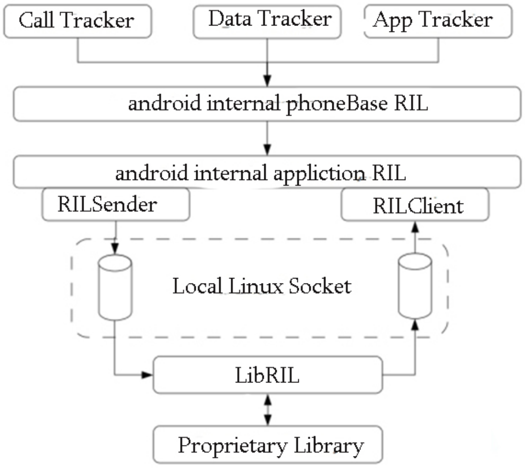


3.1.2 Hardware Interfaces

The NavUp hardware interface will primarily be dependent on the Users unique device. Either a Tablet, mobile phone, phablet.

3.1.3 Software Interfaces

The NavUp software interface will primarily be dependent on the Users device and operating system of that device (IOS, Android , Windows phone). But the general layout for android operating system is depicted in the image bellow:



3.2 Functional Requirements:

3.2.1 Functional Requirements List:

- FR1. User Registration - The system should allow the user to create an account using their email and a password. They should also have an option to sign in through another social media account.

- FR2. User Login - Given that the user was able to register the system should allow the user to login. After the first login the system should automatically log the user in.

- FR3. User Profile - The system should allow the user to CRUD their profile.

- FR4. User Password - If the user forgets their password the system should allow them to reset it. To reset their password there should be a security check or validation.

- FR5. Search - The system should allow the user should be able to search for locations, points of interest and events. The user should also be able to specify the type of location he is searching for, like restaurants or lecture halls.

- FR6. - Current Location - The system should provide the user with their current location. The system should also allow the user to share their current location with another user.

- FR7. Navigation to Location - The system should allow the user to select a location on the map or from the search results. Once the user has selected a location a route should be calculated based on the user's preferences. The user should then be navigated turn-by-turn to their destination. The user should also be given the directions in a list. If the user locks their phone the directions should be pushed as notifications.

- FR8. Heat Maps - The system should display a heat map of campus. While navigating this heat map should continuously be updated. The heat map data should be calculated based on the number of devices connected to Wi-Fi connection points. Data should also be collected via crowdsourcing.

- FR9. Location Information - The system should show the user information of their current location. This information could include history of a building, significance of a point of interest. If there are any activities taking place at that location in the near future the user should be able to see this as well. The system should allow higher level users should be able to CRUD location information. The system should also save a user's favourite locations.

- FR10. Location Based Notification - The system should push notifications to users based on their current location. The notification should be given based on the user's interests. The user should also have an option to block notifications for certain locations.

- FR11. Activities Rewards - The system should keep track of rewards. When a user completes an activity specified by a third party user, the user should be notified that they successfully completed the activity and that they are eligible to receive a reward. The system must also notify the third party who created the reward who has won it.

- FR12. Active Rewards - The system should keep track of the user's steps. The user should be rewarded with virtual badges for completing active activities. These virtual badges could be rewarded for the most number steps taken in a week the highest average steps taken in a week. The user should be able to view how far they are from receiving these badges.

- FR13. Events - The system should allow for third party users to create special events that users can participate in. These events include competitions, specials at stores and restaurants, and social events.

3.2.2 List of Use Cases:

- UC1. Search for Locations - The user enters a search query to search for a location.

- UC2. Search for Locations with a filter - The user chooses a filter then enters his search query.

- UC3. Show Current Location - The user should be shown their current location on a map.

- UC4. Share Location - The user should be able to share their location with another user.

- UC5. Calculate Route Without Traffic - The user should be able to specify for their route to be calculated without traffic.

- UC6. Calculate Quickest Route - The user should be able to specify their route to be the quickest possible route.

- UC7. Navigate to Destination - The user should be given navigation instructions to get to their location.

- UC8. Save Location as Favourite - The user should be able to save their current or a searched location as their favourite.

- UC9. Show Favourites - The User should be shown a list of their favourite places.

- UC10. Registration - When the user opens the app he should be given the choice to register if he wants to.

- UC11. Login - If the user has a account the system should allow the user to login.

- UC12. Reopen Application after closing - The application must auto-login the user. It should also continue navigation if there was a navigation in progress.

- UC13. Navigate when application is closed - If the application is still open in the background the user will receive push notifications for directions.

- UC14. Password Reset - The user forgets their password and submits a reset request.

- UC15. Show Traffic - If the user has enabled heat maps the user will be shown a heat map of campus on the map.

- UC16. User walks past location he will be interested in - The user must receive a notification that they might be interested in this location.

- UC17. Block Notification - The user has the option to permanently block a notifications from a location when he gets a notification from that location.

- UC18. View Location Information - The user opens a locations information page. The locations information will be displayed on this page.

- UC19. CRUD Location Information - Higher level users will have access to CRUD locations.

- UC20. User Gets Reward - The user completes an activity which makes him eligible for a reward. The user receives a notification informing them that they have received a reward. System notifies the third party who is giving the reward.

- UC21. User Walks Required Steps - The user receives a notification informing them that they have achieved a goal and that they have received a badge for it.
- UC22. CRUD Events - A third party users have access to CRUD special events that users will be interested in.
- UC23. Create New Event - Third party user creates new special event. Users that might be interested will be notified immediately.

3.2.3 Use Case Diagrams

Refer to next page for use case diagrams

3.2.4 Actor system interaction

Table 2: My caption

UC11:Login	
	App displays home page
If user clicks login link	App displays login page.
If user clicks continue link without logging in	TUCCW navigation.
TUCEW user sees welcome page	

UC1:Search for locations	
Actor: User	System: Location service
	Location service displays map with the following options
TUCBW user either signs in or continues without signing in	App displays map with the following options
	Search for location.
	Perform filtered search for location
	Show current location
	Share location
	Add location to favourites
	Show favourite locations
	View location details
TUCEW one of the following cases UC1. Search for location.	
UC2. Perform filtered search for location	
UC3. Show current location	
UC4. Share location	
UC8. Add location to favourites	
UC9. Show favourite locations	
UC18. View location details	

UC7:Navigate to destination	
Actor: User	System: Navigation
	Navigation displays navigation page with the following options
TUCBW User sees navigation page	
	Calculate route without traffic.
	Calculate quickest route
TUCEW one of the following cases UC5.Calculate route without traffic.	
UC6.Calculate quickest route.	

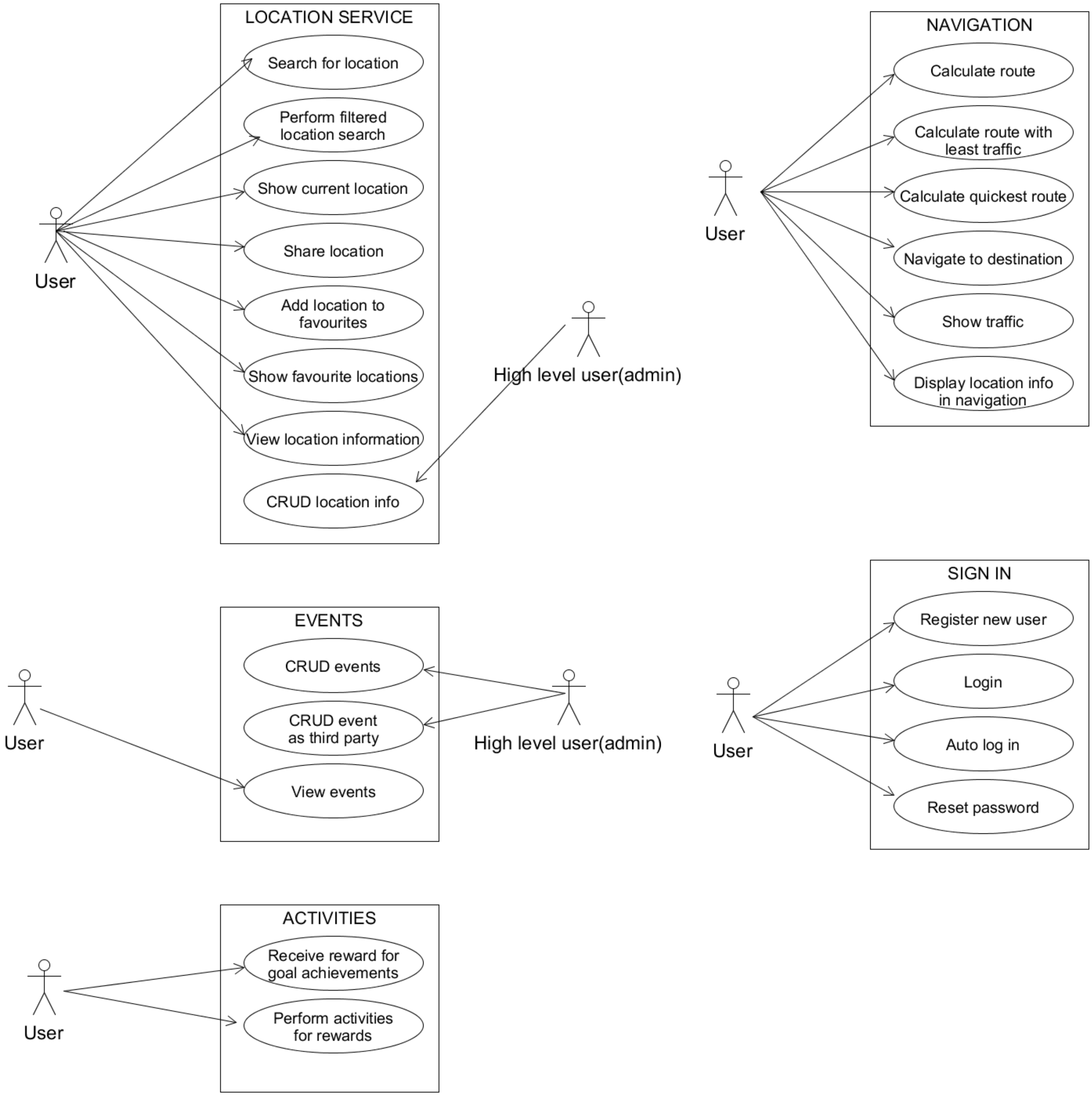


Figure 1: Use case diagram

UC5:Calculate route without traffic	
Actor: User	System: Navigation
	User has selected destination
TUCBW User clicks ”Calculate route” button after selecting destination	
	App displays home page
If user clicks calculate route without traffic button	App determines and displays route on map with least traffic.

UC6:Calculate quickest route	
Actor: User	System: Navigation
	User has selected destination
TUCBW User clicks ”Calculate route” button after selecting destination	
	App displays home page
If user clicks calculate quickest route button	App determines and displays quickest route.

UC3:Show current location	
Actor: User	System: Location service
	Map has finished loading
TUCBW user clicks ”show current location” button	Location service determines user location
	User location and data shown on map

UC4:Share location	
Actor: User	System: Location service
	User is at/selects point of interest
TUCBW	
If user clicks share button	App shares current location with other app users.

UC2:Perform filtered search for location	
Actor: User	System: Location service
TUCBW search bar is clicked	Filter options are displayed
If user clicks on filter button	App performs search based on filters chosen by user.
TUCEW user sees search results	

UC8:Save location to favourites	
Actor: User	System: Location service
	User is at/selects point of interest
TUCBW ”Save location button is pressed”	
If user clicks save location button	App adds location to list
TUCEW confirmation dialog appears	
	updated favourites list displayed

UC9:Show favourites	
Actor: User	System: Location Service
TUCBW ”Show favourites button is clicked”	App fetches list of favourites from server
TUCEW favourites page displayed	

UC10:Registration	
Actor: User	System: Login
TUCBW	
	App displays home page
If user clicks login link	App displays login page.
If user clicks continue link without logging in	TUCCW navigation.
TUCEW user sees home page	
	User directed to home page

UC12:Reopen app after closing	
Actor: User	System: App
	App currently minimized/in use
TUCBW App is reopened	User signed in automatically
	App displays home page
TUCEW user sees home page	

UC13:Navigation when app is closed	
Actor: User	System: Navigation
	App running in background
TUCBW app minimized while navigation in process	
TUCEW app is reopened/arrival at destination	

UC14:Password reset	
Actor: User	System: Login
TUCBW User clicks ”forgot password” button	App displays password retrieval page
	App displays home page
TUCEW user sees welcome page	

UC15:Show traffic	
Actor: User	System: Navigation
	Map already loaded
TUCBW ”Show traffic option is selected”	fetch traffc data and display as heat map
TUCEW user sees traffic data on map	

UC16:User walks past location he will be interested in	
Actor: User	System: App
TUCBW User walking past point of interest	App displays location
TUCEW User walking away from point of interest	

UC17:Block notifications	
Actor: User	System: Notification
	User receives notification
TUCBW User selects ”block notifications” option	App dismisses notification
TUCEW Particular location added to ”blocked” list	

UC18:View location information	
Actor: User	System: App
TUCBW user clicks location information link	
TUCEW Location information displayed on screen	

UC19:CRUD location information	
Actor: High level user	System: App
	Display location information management page
TUCBW Location information management page is loaded	
TUCEW High level user CRUD location information	

UC20:User gets reward	
Actor: User	System: Activities
TUCBW User achieves set goal eligible for reward	System grants user eligibility to collect reward
TUCEW user collects reward/waits to collect later	

UC16:User walks required steps	
Actor: User	System: Activities
TUCBW User’s steps walked reach set number	
	App displays banner message
If user takes no action wait	App displays login page.
Click banner or open app to retrieve reward	Rewards page displayed.
TUCEW banner message disappears after set time	

UC22:CRUD events	
Actor: User	System: App
	Event management page diaplayed
TUCBW Third party user sees events management page	
TUCEW Third party user CRUD events	

UC23:Create event as third party	
Actor: User	System: Events
TUCBW ”Create event” button is clicked	Page displayed to capture details
TUCEW Event added to location details	

3.2.5 Traceability Matrix

Requirement	Priority	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14	UC15	UC16	UC17	UC18	UC19	UC20	UC21	UC22	UC23
FR1	3										X													
FR2	**3**											**X**	**X**										**X**	**X**
FR3	**3**								**X**	**X**	**X**	**X**			**X**									
FR4	**3**														**X**									
FR5	**5**	**X**	**X**																					
FR6	**6**			**X**	**X**				**X**								**X**		**X**					
FR7	**6**					**X**	**X**	**X**					**X**	**X**										
FR8	**4**					**X**										**X**								
FR9	**5**									**X**										**X**	**X**			
FR10	**4**																**X**	**X**						
FR11	**3**																				**X**			
FR12	**1**																					**X**		
FR13	**3**																						**X**	**X**
UC Priority		**5**	**5**	**6**	**3**	**5**	**5**	**5**	**6**	**3**	**3**	**4**	**3**	**3**	**4**	**5**	**4**	**2**	**6**	**6**	**3**	**2**	**4**	**4**

3.3 Performance Requirements

1. Low battery consumption

    - Since the application uses predominantly Wifi will be priority to minimize impact on the battery of the phone since battery life is high concern of the users.
    - Numerous calls to various functions when navigating the user from point A to point B, thus optimising this functionality will be a great concern since this can become a high impact on the battery, draining it rather fast.
    - With the added assumption that the user will grant permission to allow us transmit the device's location in attempt to help the application to navigate the user to locations with no signal for GPS but have WiFi accessibility.
    - Heat map generation will require additional request sent from devices to the routers if this is constantly updated it will start impacting the battery heavily.

2. Fast real-time navigation updates

    - Important aspect of navigating a user is the accuracy of the process. Thus making use of available resource effectively can create an accurate representation and navigation the user to the desired location. The resources that will be used is the Heat Map, other user locations, WiFi and GPS when available.

3. Accurate Step Counting

    - Awards will be awarded based on steps taken thus the counting must use as much of the resources at its deposal to be as accurate as possible to distinguish between actual walking and people bouncing their legs when seated or swinging their phone around to attempt to fool the system.

4. Accurate location tracking

    - This is the main promise of the application. Everything will be off and wrong if the location is inaccurate. Thus this performance requirement will be most important.

3.4 Design Constraints

The database has to be able to handle more than 30000 users. The applications is supposed to be designed in such a way that it can work with WiFi, GPS and Mobile networks. It should switch between WiFi as necessary.

The application will need hard drive space on the device. The lack thereof will result in an inability to download. Based on research the application should need between 100 - 200 MB of harddrive space.

Memory usage will be high because the application will always be running in order to track the users steps unless the user decides otherwise. The data should not be more than 20MB.

3.5 Software System Attributes

1. Reliability

   - Our system makes a lot of assumptions in terms of certain resources that will be available along with permission to use them. Such as User permission to access their GPS/Location/WiFi/Accelerometer.
   - Our system relies on availability of WiFi across the campus of the University of Pretoria, that campus has power and or the generators work if there is no power.

2. Security

   - There are four levels of Users with each having similar and different rights. The differences must be sure to authenticate the user trying to execute the function. If the user is not authorised to do so they must be denied access or if they are authorised they must be allowed to continue.

3. Availability

   - The availability is non existent if the user denies all of the permissions. If we do not have access to the device's GPS/Location/WiFi/Accelerometer none of the application's functionality will be able to work.
   - Limited availability overall but the main functionality will be achieved with permission to use the user's Location and the user's WiFi.
   - Most accurate functionality of the main functions can be achieved with permission to use Location/WiFi and GPS of the device and use the user's permission to send anonymous pings to generate heat maps and use other devices' location to navigate the user.