NavUp Requirements specification

February 21, 2017

Contents

1 Introduction

This section gives a description and an overview of all the information in the SRS document. The purpose of the system will be provided including additional information of the system.

1.1 Purpose

This document serves to describe the requirements of the NavUP system. This includes the overall features of the system, functionality, interfaces, constraints and integration. The purpose of this document is intended as a communication tool between developers and client to reach consensus on what the client's requirements are for the system. After requirements elicitation and final draft of the SRS is complete, the client should sign off on the document if the client agrees with the specification. After sign off, this document shall be used as a reference material for developers when creating the system.

1.2 Scope

NavUP is a project proposed by the University of Pretoria's department of Computer Science. The objective of the system is to provide the user with an interface from which they can select a location(building or classroom). The system will then calculate the fastest route for the user based on user specified restrictions. The system will also provide information on points of interest on campus. The system will be available to staff, students and visitors to the University of Pretoria.

The system will provide users with an easy way to traverse campus and get information on the various points of interest. This will include historical buildings, events and activities. Users will be able to set restrictions on the navigation part of the system, for example, avoiding pedestrian traffic and accessibility for disabled people. There will also be an option to save locations and routes for future use by the user.

The system will determine the users location both indoors and outdoors through various ways like wifi signal strength, gps location and crowd sourcing. This will allow users to connect to the system and provide them with the necessary information to navigate campus. This is beneficial to new students and visitors that do not know the layout of campus yet. This will also provide users with fast routes to their location when time is of the essence by avoiding pedestrian traffic and activities/events.

1.3 Definitions, Acronyms and Abbreviations

Term	**Definition**
NavUP	The navigation system that is proposed in this document.
User	The people that will be using the system.
Guest User	The people that will use the system, but will only have limited functunality.
Admin/Administrator	Person given permission for managing and controling the system.
Location	A precise point on a map.
Venue	A building name or a room within the building itself
Points of interest	Various locations that may interest users to visit them.
Events and Activities	Various events/activities that may take place on the campus.
Restrictions	User selected preferences when using the navigation system NavUP.
Network	A system consisting of various interconected computers and hardware.
Heatmaps	An indication (usually in color) of a congested or populated area.
Wifi	A facility allowing computers, smartphones, or other devices to connect to the Internet or communicate with one another wirelessly within a particular area.
Wifi hotspots	Areas where users will be able to connect to the network wirelessly.
GPS	GPS, which stands for Global Positioning System, is a radio navigation system that allows land, sea, and airborne users to determine their exact location, velocity, and time 24 hours a day, in all weather conditions, anywhere in the world.
UI/User Interface	The method of which users will interact with the system.
GIS	A geographic information system (or GIS) is a system designed to capture, store, manipulate, analyze, manage, and present spatial or geographic data.
Android	An open-source operating system used for smartphones and tablet computers.
iOS	An operating system used for mobile devices manufactured by Apple Inc.

1.4 References

Bibliography: Kung, D.C. (2013) Object-oriented software engineering: An agile unified methodology. 2013th edn. New York: McGraw Hill Higher Education. In-line Citation: (Kung, 2013)

1.5 Overview

1.5 Overview

The remainder of the document consists of two more chapters, an appendixes and a index.

The second chapter provides the overview description of the system which gives a more explicit structure of the NavUP system's functionality in terms of the product's perspective, functions, user characteristics, constraints, assumptions and dependencies.

The third chapter provides the specific software requirements the system consists of which is the external interface, functional, performance requirements.This chapter also specifies the design constraints, software system attributes and other requirements.

The appendixes provides the subsidiary matter at the end of the document together with the index which is a list of names or words which were mentioned in the document with reference to their page number.

2 Overall Description

2.1 Product Perspective

2.1.1 System Interfaces

2.1.2 User Interfaces

2.1.3 Hardware Interfaces

2.1.4 Software Interfaces

2.1.5 Communications Interfaces

2.1.6 Memory

2.1.7 Operations

2.1.8 Site Adaptation Requirements

2.2 Product Functions

Products function

The products function enables the user to enter their personal information and interests to be used in the functionality of the system.The system enables user's to find a route based on their selected destination, preferences and interests using they personal profile. The mobile functionality is mainly for the user's while the web-based side is more for the administration purposes.

For the mobile functionality the system should allow priviledged managers which may specify the location and movement activities therefore gaining access to individuals data records.The user's details will be used in tracking the user for various activities.Their personal interests are to be used in new information provided in the event of passing a point of interest, event taking place in the system or route provided to personal preferences.The result of the route should be viewed in the form of a map.The user's location , designated destination, pedestrian traffic and new information should be also visible whilst moving towards the designated area.

For the web-based functionality the system should be allow for overall data analysis.Data analysis performed should include user, perfomance and statistical analysis which aid in maintaining and devloping the system at large.

2.3 User Characteristics

The bulk of users will consist of mobile users. Mobile users will consist of guests with the age group ranging from primary school children to students, alumini and lecturers. Basic competence with smartphones is assumed in addition to navigation skills with map based apps such as google maps.

The web based side of the system will consist of administrators, who should be able to ammend, capture and delete information from the database, thus some knowledge of database management is required. The web based interface will also serve researchers who will be able to analyse data. It is assumed that the researchers have the appropriate skills to conduct their research, this includes domain specific knowledge and basic skills in querying a database.

Lastly the system aims to cater for the disabled. The disabilities which will be accomodated have not yet been fully clarified. But the same level of competence is expected from the disabled as from the general mobile user population to make use of the navigation features. This system is mostly geared toward the physically disabled in order to find the most accomodating routes to classes and buildings.

2.4 Constraints

No constraints have been voiced by the client

2.5 Assumptions and Dependencies

Factors that could affect the requirements include compliance with the Protection of Private Information Act of 2013. The POPI act is applicable since user location data gets captured and assigned to a persistent ID and then stored in a database. This means that compliance requires certain additions to functional requirements such as asking the data-subject for consent when the app is opened for the very first time and allowing withdrawal of consent and destruction of identifiable data within a reasonable time. It could also potentially restrict certain functional requirements such as surveilance and analysis.

For the system to function as expected, we must assume that there will be sufficient wifi and GPS coverage and the sensors, when integrated, will provide sufficient accuracy to serve its purpuse. We also assume that map data will be available with a sufficient degree of accuracy and completeness to satisfy the requirements of the system.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interface

The user interface for NavUP application will be specifically designed with users in mind, allowing them to have the best and easiest user interface possible. The aim is to make the user interface easy to navigate so that users spend less time trying to figure out how to use it.

We will offer a menu service to allow users of different levels to access their respective sections with ease, these sections would be Student, Lecturer and Guest. If the user chooses Student or Lecturer then possible registration and login screens may be necessary. There will also be a menu service after each user has selected their particular section so as to allow the user to choose their current use of the application.

3.1.2 Hardware Interface

The hardware interface to support the NavUP application would require a smart-phone on which the application will be downloaded and run on. The smart-phone would have to meet the standards of the operating system that the application is going to be developed on,as well as a colour screen with a resolution to match that specified later.

3.1.3 Software Interface

The NavUP application would be designed to run on Android and IOS devices, this is to be confirmed at a later stage. Most devices with this platform feature touchscreens that would allow the application to run at its best. Information can be entered using the on-screen keyboard and the pre-existing touch and drag approach could be used to navigate the map interface.

3.1.4 Communications Interface

The application would mainly use WiFi connections, GPS system and cellphone towers to triangulate and estimate the position/s of the user/s. Information would also have to be collected from the user/s, to allow this, any form of data communications would be allowed, although in essence this would be done mainly through a WiFi or cellphone data connection.

3.2 Functional Requirements

Serial:F1
Abstract: Some abstract
Description: the longer description
Pre-condition: Any preconditions
Post-condition: Any postconditions

3.2.1 Navigation

Base User requirements
The base user requirements outlines the base/foundation functionality of the system. The inner workings of the system and it's subsystems will be expanded later in the document. Your typical base user will be able to do the following:

Serial:F2
Abstract: View the map of campus
Description: The user will be able to view the map through the use of the UI. The map itself will be saved on the system enabling access to it when connected or disconnected. It will contain all locations and venues located on campus itself.
Pre-condition: TODO
Post-condition: TODO

Actor system interaction model: View the map of campus

Actor: Mobile User	System:NavUp
	0. The mobile app displays the main menu with the map and map info occupying most of the screen at the default position.
1. The user scrolls around the map	2. The app updates the map the map and map info that's in view, pulling information from the remote repository if needed and if possible.

Serial:F3
Abstract: Search for a location/venue on campus to be displayed on the map
Description: The user needs to be able to search for a location/venue. This will involve a search bar (possible auto-complete functionality) where the user will enter the name of the location/venue. The mobile device will then search through it's database and indicate on the map when the location/venue has been found or display an appropriate prompt when the location/venue is not found.
Pre-condition: TODO

Post-condition: TODO

Actor system interaction model: Search for a location/venue on campus to be displayed on the map

Actor: Mobile User	System:NavUp
0. The user clicks on the magnification button	1. The mobile app displays a menu to type in the location and venue.
2. The user types in the location or venue	3. Using information from cache and the remote repository, the app tries to perform autocomplete based on what the user is typing.
4. The user chooses the autocomplete suggestion or types the search term out.	5. The app generate a list of candidate locations/venues that match its internal cache or the remote repository.

Serial:F4
Abstract: Find the fastest route between two points
Description:

1. The user will be able to set the start point in the following ways:
   (a) By clicking on the map to indicate the start point or typing the name of the location into the search bar as described in another functional requirement. This should work for both online and offline.
   (b) The start point is determined by integrating information from various sensors such as wifi and cellphone tower triangulation, gps, accelerometer and gyroscope. The functionality may work for both online and offline if possible.
2. The user will be able to set the end point by clicking on the map to indicate the start point or typing the name of the location into the search bar as described in another functional requirement. This should work for both online and offline.
3. After setting the start and endpoint the Mobile device must calculate the fastest path between these two points and provide the directions to the user. If the phone is offline it must make use of cached data to make an informed choice, if it is online, it must consult the remote repository for the most up to date congestion data.

Pre-condition: TODO
Post-condition: TODO

Actor system interaction model:Find the fastest route between two points

Actor: Mobile User	System:NavUp
0. The user clicks on the navigation button	1. The mobile app displays a menu to type in the starting point and end point.
2. The user selects the starting point by making a mark on the map or setting it as current location of the mobile device or searching for a place on the map	3. If the mobile device is online, the remote server sends up to date congestion data to the mobile device. If the mobile device is offline, it uses cached data. In both cases the mobile phone calculate the fastest path using the congestion data, then displays the fastest path to the user.

Serial:F5
Abstract: View various points of interest
Description: When selecting a certain building or location the mobile device should display information where applicable for the user to review. When offline the mobile device will use whatever it has in cache, if it is online it may consult the remote repository for the most up to date information.
Pre-condition: TODO
Post-condition: TODO

Serial:F6
Abstract: View any current events or activities happening on campus
Description: The mobile device should display various events and activities happening on campus on the current day as well as upcoming events and activities. When offline it should attempt to display activities and events that have not expired that resides in cache. When online, it should retrieve the most up to date list of events and activities from the remote repository.
Pre-condition: TODO
Post-condition: TODO

Serial:F7
Abstract: Users should be able to view any congestion/ high traffic in areas
Description: When using the application in a map view of the campus, users should be able to view the amount of pedestian traffic in surrounding areas. In addition to this any congestion in areas will also be indicated, this could be in the form of pedestrians or any other kind.
Pre-condition: TODO
Post-condition: TODO

Serial:F8
Abstract: Notifications and alerts should be pushed to users
Description: The application will allow important notifications and alerts to be given to users in a mass instance. In emergency situations the application

could be used to give users directions to assembly places or something of the like.
Pre-condition: TODO
Post-condition: TODO

Disabled User requirements

Logged-in User requirements
The logged in user will have certain privileges over and above those of the base and guest users. There are two types of logged in users, namely students and lecturers who would log into the system using some form of user-name and password. The typical logged in user will be able to do the following:
Serial:F9
Abstract: Find places of interest based on saved user preferences
Description: The users will be given the option to populate a list of preferred preferences and places of interest. The system will then generate and display a list, when needed, of the places of interest and various events that are happening currently. This list that is provided to the user is unique and tailored to them.
Pre-condition: TODO
Post-condition: TODO

Serial:F10
Abstract: View a list of upcoming events in places of interest
Description: Using the saved list of preferences and places of interest that the user has provided, the system will automatically generate a list of the upcoming events that are taking place at these places and the user will be able to view them in a time-line manner and set reminders for these events.
Pre-condition: TODO
Post-condition: TODO

Serial:F11
Abstract: View a history of past places visited
Description: Users will be able to view a history of the past events that they have attended. This will include the type of event that took place and the location that the event was at.
Pre-condition: TODO
Post-condition: TODO

Serial:F12
Abstract: View a history of a users step count
Description: The system will will track and record the step-count of each user, this history will be viewable by the user according to the time-line that the user will be able to specify.

Pre-condition: TODO
Post-condition: TODO

Guest User requirements
In regards to the Guest Users, they will have the same requirements of a Base User (See Base User Requirements). They won't have an associative ID linking them to the database seeing as they will not be logged in. As soon as they do so they will be seen as a Logged In User. This means they will not recieve any extra functionality from the system except for the base functionality.

3.2.2 Agent Requirements

Agent User Requirements
The agent is a metaphor for the system itself that will have privileges, such that it will have access to all the users' information and analyse this, after which meaningful data will be generated from there and be used to form the real-time notifications that will be pushed to other users. The agent will be able to do the following:

Serial:F13
Abstract: Analyse and integrate the information sent from users
Description: The agent will have access to different forms of data that the users will continuously send to it and the agent organises this data so that meaningful information can be extrapolated and used. This data will be in the form of location coordinates, stagnant users, step counters,user preferences and places of interest time taken on a specific route and possibly more. This data will then be analysed by the agent and real-time feedback will be given back to the users in terms of fastest routes, traffic congestion, places they may be interested in, nearby events, surveillance information and possibly more.
Pre-condition: TODO
Post-condition: TODO

Serial:F14
Abstract: Privilege management and authentication
Description: The agent will be able to distinguish between the different types of users and convey the relevant information to them as well as differentiate the type of access rights that each type of user has on the data, for example mobile users will only have read access to congestion data, agents will have read and write data and administrators will have read and write access on the data.
Pre-condition: TODO
Post-condition: TODO

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software System Attributes

3.6 Other Requirements