Department of Computer Science

COS 301 - Mini Project

Architectural Specification

Tshepo Malesela	14211582
Mfana Masimula	12077713
Lebelo Tlou	15209190
Wandera Henry	17253129
Mandla Mhlongo	29630135

March 13, 2017

Contents

Architectural Requirement Specification

Team VbScript github repository link

For further references see Github March 13, 2017

0.1 Architectural Scope

- The System should be accessible at anytime while the user is on campus.
- The server that host the system and the database is always on.
- The ability to navigate a given destination on campus
- The system must know your current location.
- The system must be able to to give ...
- The system need to be able to host all the users on campus.
- Administrative users have more access to the system have CRUD capabilities on events.
- Each event on the system must have a destination and multiple ways of reaching that destination through the navigation.
- The system must have the ability to notify users of current event.
- Users have the ability to set a route based on user constraints such as routes with less pedestrian traffic.

0.2 External Interface Requirements

User Interface

- Standards for ...

  Fonts - Verdana
  Images - square, ratio 1:1
  Buttons labels - Button(HTML input element) , LinkButton and ImageButton
  Color schemes - UP Colors , white , blue and red
  Field tabbing sequence - row by row

- Standard buttons, functions and navigation links that will appear on every screen.

  - Help button
  - Search button
  - Users: Login / Logout link
  - Navigation: always display users current location.
  - Events: button to navigate to current events on campus.

- Shortcut keys.

  Mobile Devices (Android / iOS)

  - Volume UP = Open textbox to search New direction
  - Volume DOWN = View events on campus

  Desktop Computer

  - Ctrl + F = Open textbox to search New direction
  - Ctrl + E = View events on campus
  - Ctrl + Z = logout / logOff
  - Ctrl + Alt + H = Help

- Message dispaly conventions
  - Dialog box: Error message , usage/help message
  - Notification icon:
    - Display on login, if user have not subscribe to recieve notifications either via SMS or email.
    - Alert user if any new events have been added or is about to take place.
  - Status bar: Dispaly users current location at all times.

- Accommodation for visually impaired users.

  1.) Allowances For Enlarged Text: Stylesheet with large font size whose layout does not break when text-only zoom is enabled in browser.

  2.) Use Keyboard Shortcuts to Aid Navigation: Possibly being able to nevigate the whole application with the use of only shortcut keys.

  3.) Contrast is Key: Bold text on low contrust items and user ability to highlight text with mouse (quick trick to increase contrast and to aid visual focus).

Hardware Interface

- Supported Device type
  - WiFi supported divice
  - Precise location (GPS and network-based) support

- Data and control interaction between the software and hardware

- Communication protocol
  - - JSON (JavaScript Object Notation)
  - Ajax (Asynchronous JavaScript and XML)

Software Interface

- Databases:
  - Databases: MongoDB
  - Operating Systems:

- Operating Systems:
  Mobile devices:
  - Android 4.2 (Jelly Bean) or later
  - iOS 8 or later

  Desktop:
  - accessible through Web browsers (Google chrome or Mozilla Firefox) @www.nav.up.ac.za

Communicatons Interface

0.3 Performance Requirements

Response Time

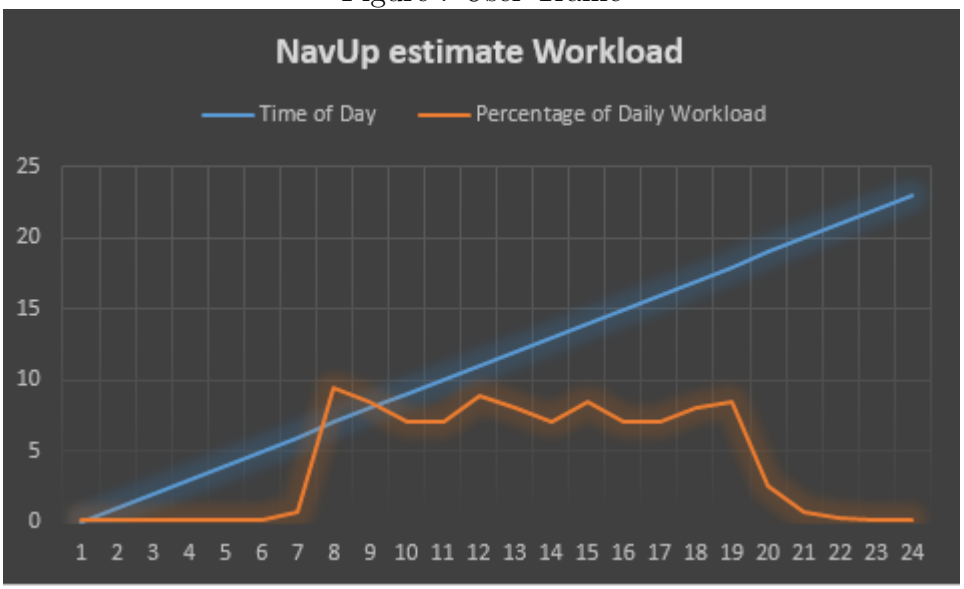
Support 90000 Guests, users and admin.
Users consisting of students, lectures, and university employees.
The system should be able to withhold the following performance requirement:
- 95 percent of all the time (excluding Route Navigation response) the system should respond within 5 seconds.
- Less than 10 seconds response time for normal expected route navigation during more than 60 percent of the time as depicted by conditions on campus.
- Atmost 20 seconds allocated for Route Navigation response time, at which Error messages for system failure are issured eg "Navigation taking longer then expected"

Workload

Table 1: User scenario statistics.

Scenario	Daily Total	Pages	Think Time
New user/Guest	500	Register, Login	1 min
Looking for fun	2500	Events, Navigation	25 seconds
Touring Campus	150	Points of interest, Navigation	25 seconds
Lost on Campus	4500	Navigation	15 seconds

Figure : User Traffic



Platform

Not specific. Open to all systems that have WiFi support integration with functional requirement to enable GPS (Global Positioning System) operations. This include standard navigation features such as good graphics card for mapping directions in any format 2 or 3 dimensonal.

0.4 Design Constraints

- The system will only use the Wifi network.
- The system should be as small as possible, preferably less than 150MB.
- The system should be user friendly and have a pleasant user interface.
- The system will be accessible on smart devices mainly using IOS and Android platforms.
- The system should only perform updates at night when the system work load is presumably significantly less.

0.5 Software System Attributes

Maintainability

The system should be simple and easily understandable so as to readily facilitate upgrading of the software when necessary. This will be achieved by using the latest available software at our disposal. A modular design will allow parts of the system to be maintained with affecting the whole system.

Usability

A short tutorial must be provided to the user showing the capabilities of the system and how it must be used. This must be in straight forward, easy to learn manner. The state in which the application is closed must be saved. The system must provide suggestions to the user aswell as a help option should the user get lost. The system should allow the user to choose from a list of languages to use for interaction(voice command could added at a later stage). Deploy a friendly game-like interface that'll cater for all age groups.

Scalability

The system should still be able to perform optimally even under huge work load and traffic. If the system reaches a bottleneck, it should disable "nice to have" feature and only provide the core basic functionality.

Reliability

The system should be able to check for common requests(prospective student looking for the CSC) and load the directly from the server therefore minimising the traffic to the server. The system should continuously perform back-ups in the event that the system crashes or the generators don't kick in quick enough when a power failure occurs. The system should provide suggestions close to the user input if the input searched for is not found. If there are multiple paths to a destinations the system must let the user know about this but the shortest path must be the one suggested to the user.

Availability

The system should be available in times where most users are presumed to be on campus(07h00-18h00) aswell as big days on the University calender, UP birthday, Open Day etc.

0.6 UML Design

Domain Model

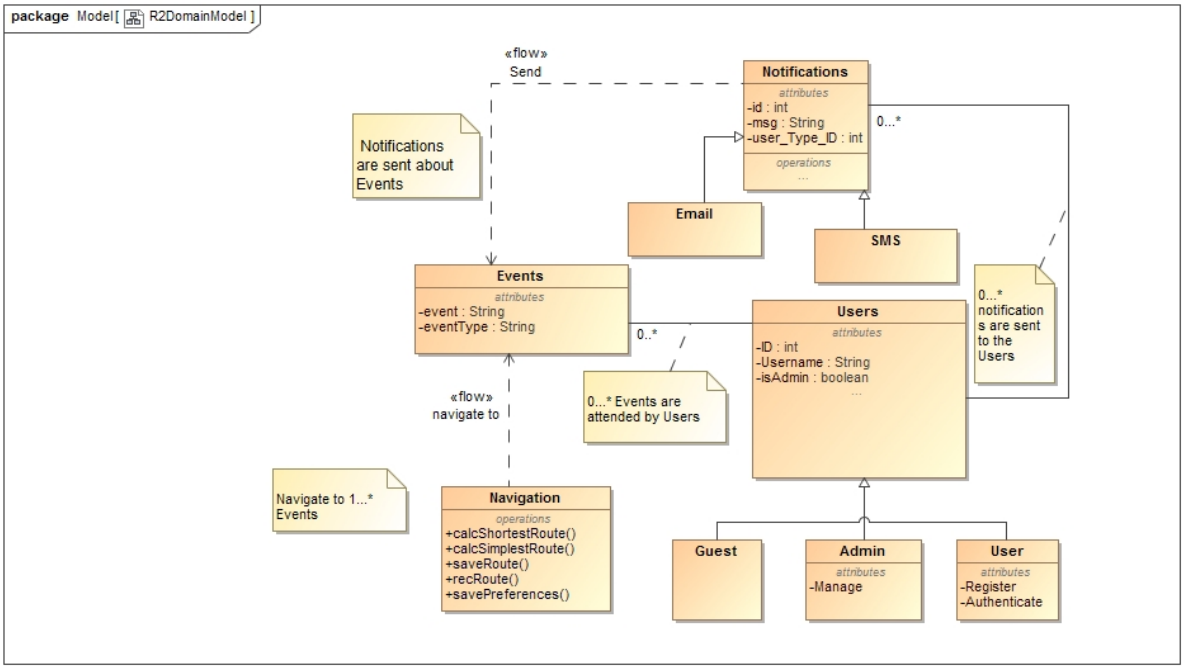


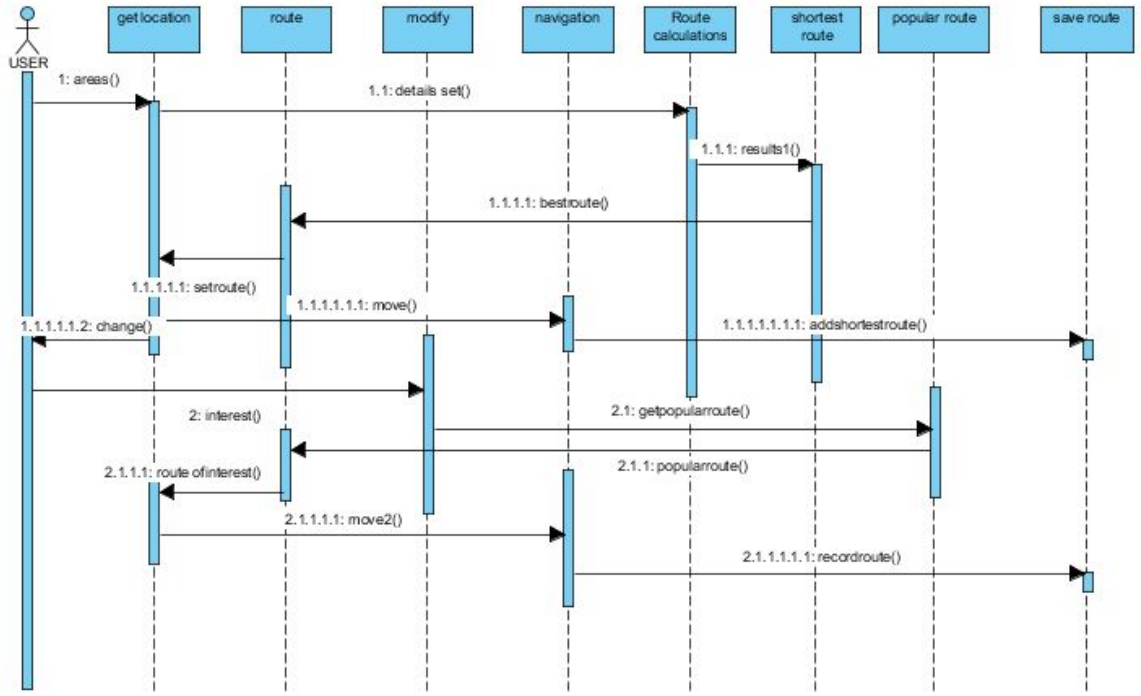
Figure 1: Domain Model Class Diagram

Package Diagrams

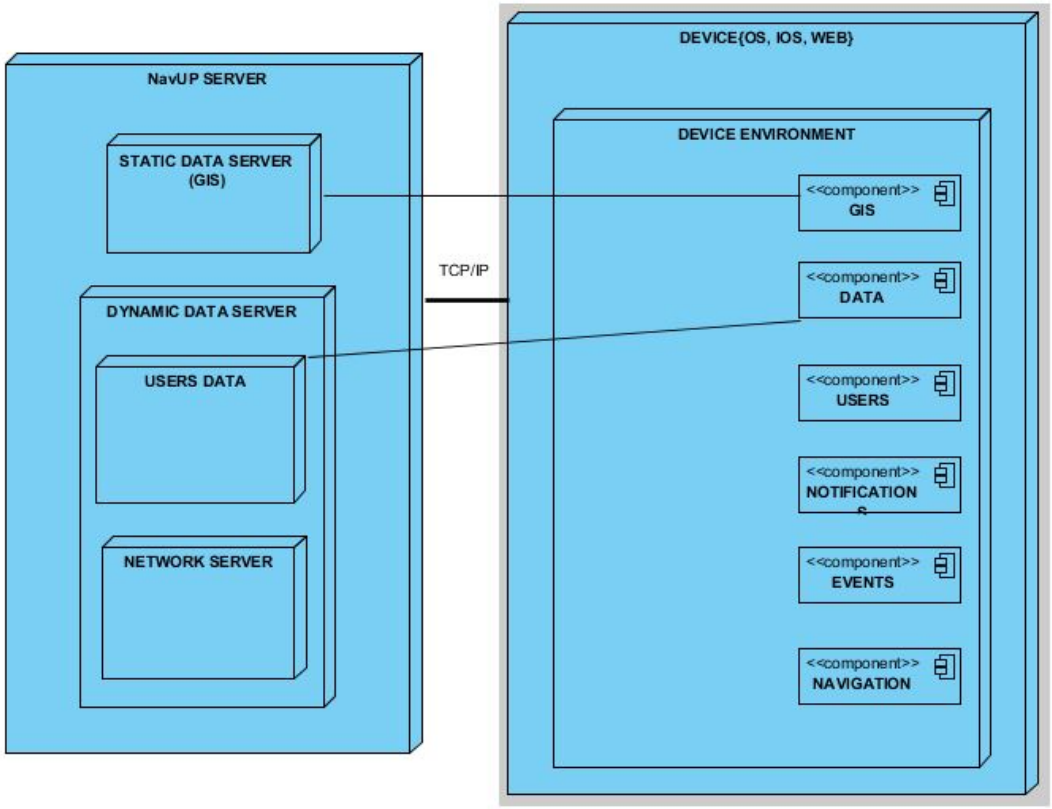
Events

Sequence diagram

Navigation Module



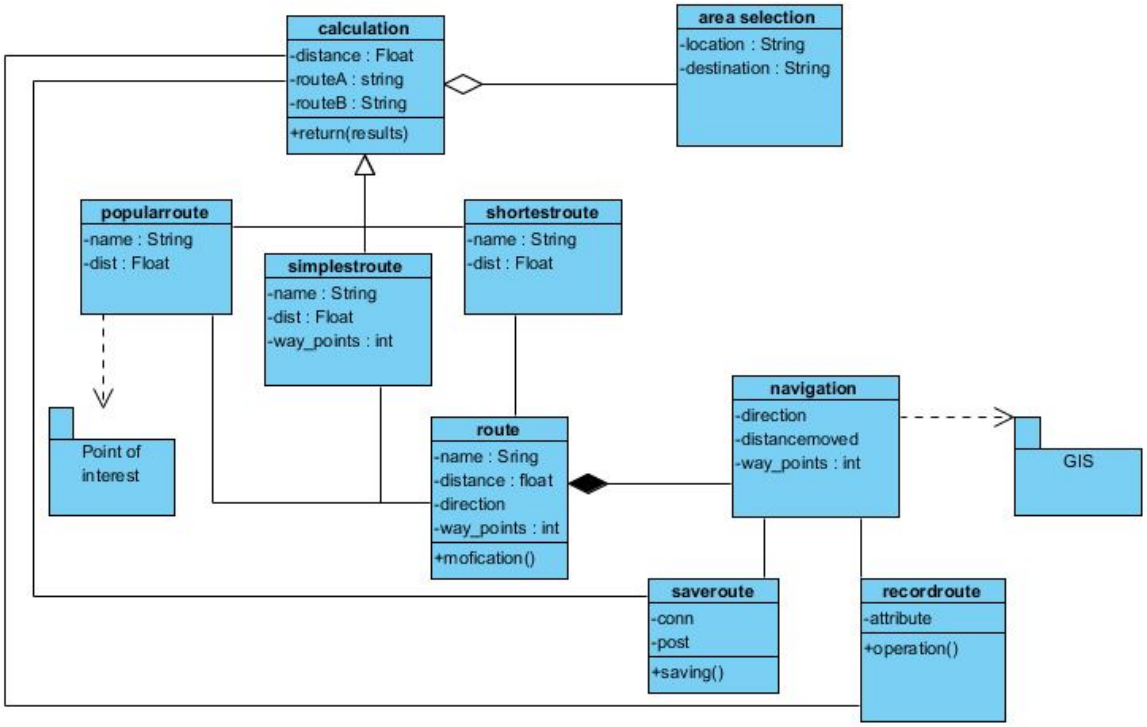
Deployment Diagram



The GIS works with the Static data server for storing information about locations and this information is available for other modules like the navigation module. The static data server holds information about users and their preferences.

Class diagrams

Navigation Module



Events

Notification Module

Users Module

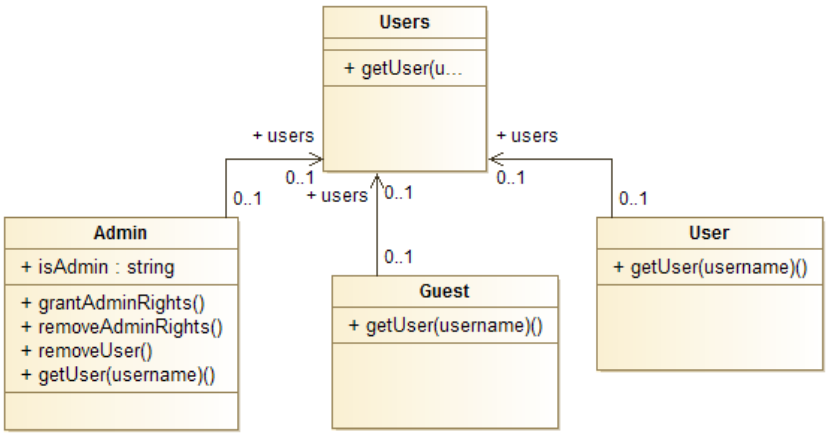
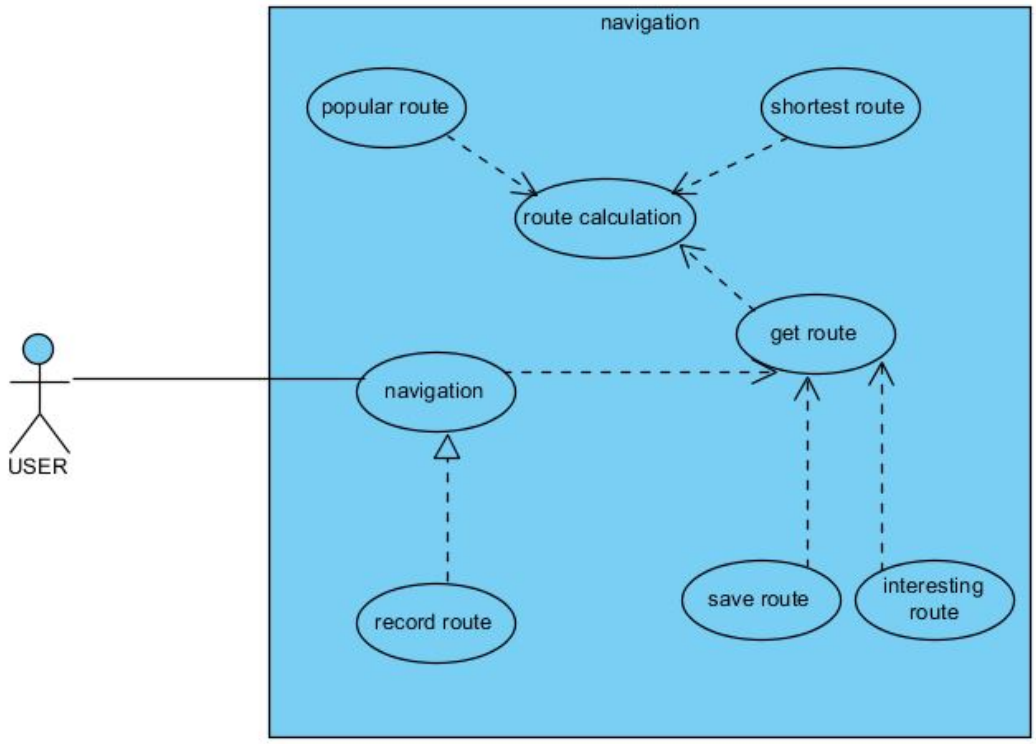


Figure 2: Users Class diagram

Use Case Diagrams

Navigation Module



Events
Notification Module
Users Module

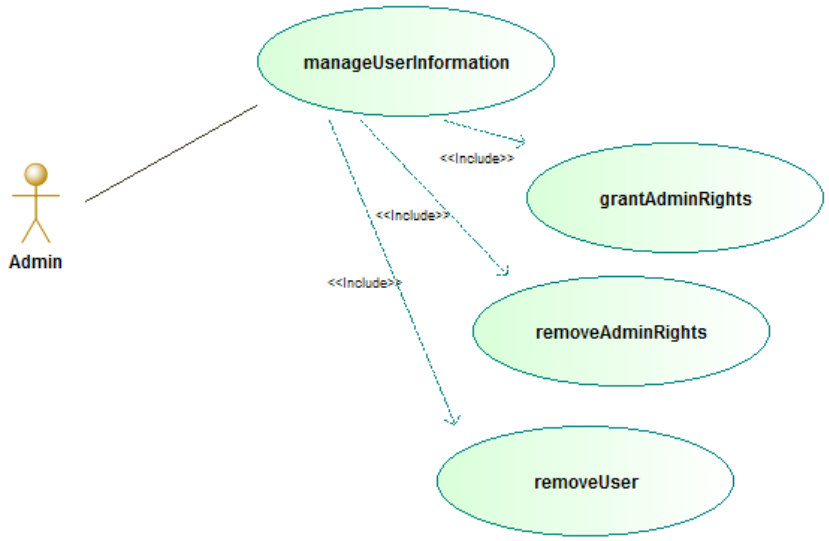
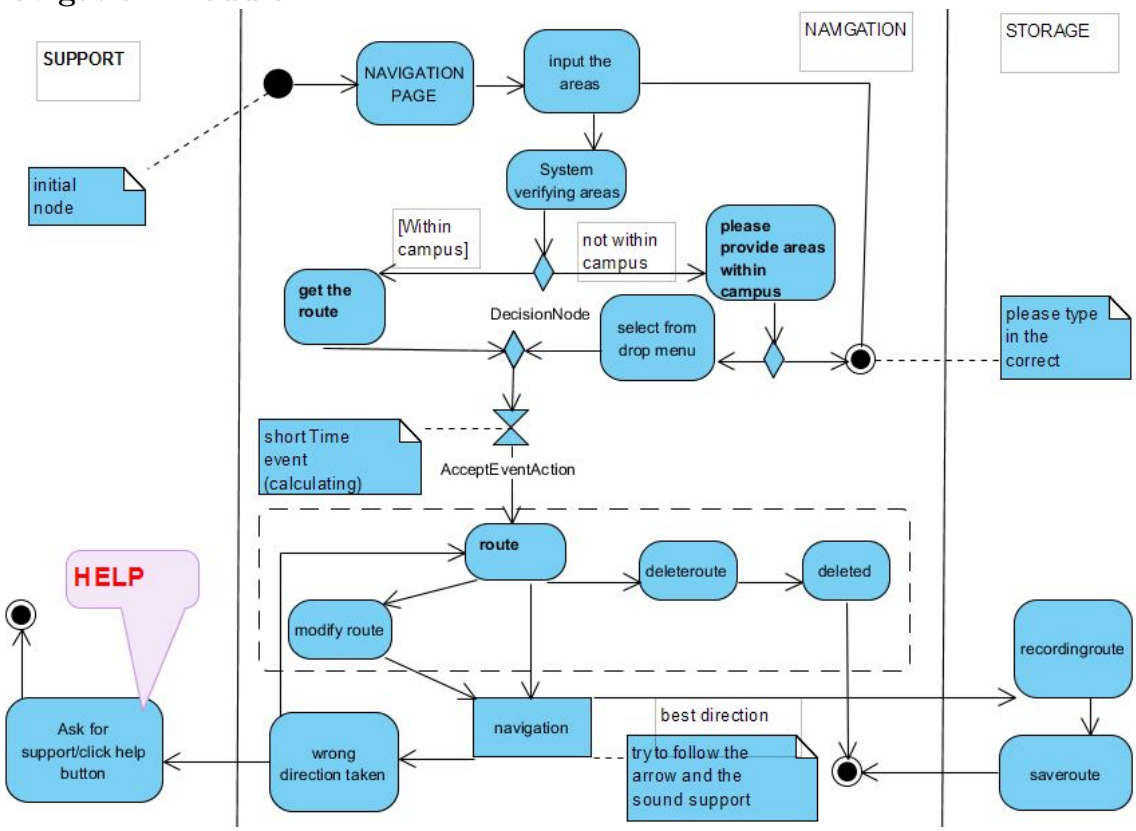


Figure 3: Users Use case diagram

Activity Diagrams

Navigation Module



Events

Notification Module

Users Module

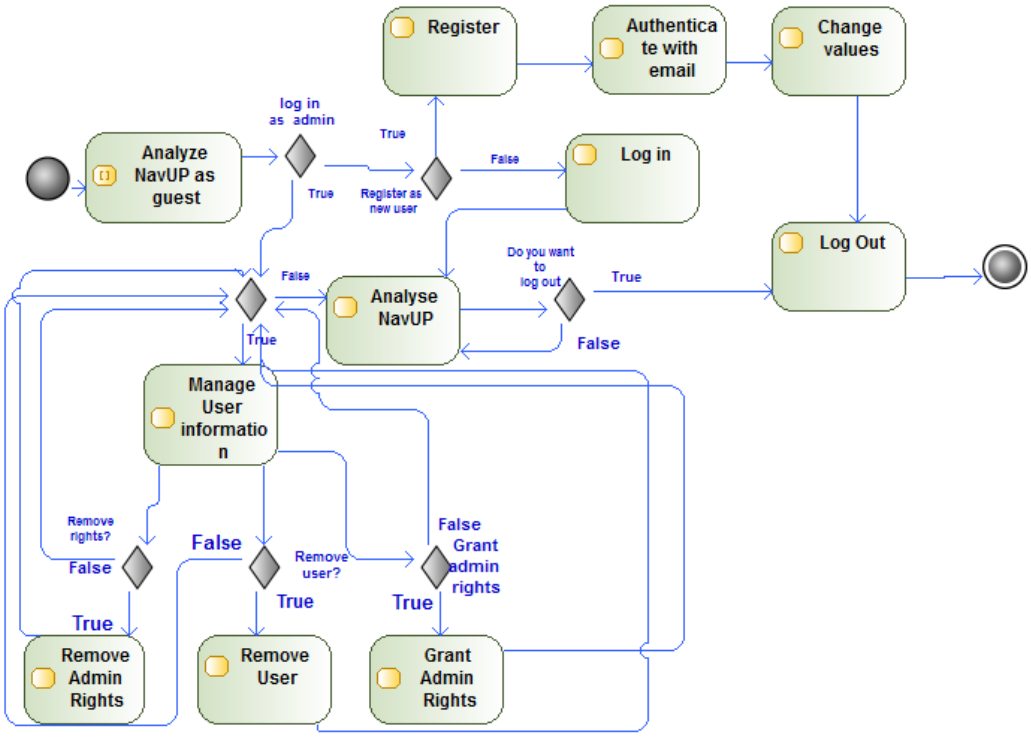


Figure 4: Users Activity diagram

0.7 Design Patterns

User Module
For the User module a Template Method willl be used because since there are 3 classes of different users. In the abstract class User defines the abstract primitive operations that appear as steps in the template method. The concrete classes are Admin, Guest and User which define the primitive operations.

- Abstract Class : Users.
- Concrete Class : Admin, Guest and User.

0.8 Technology

Choosing the correct technologies is very important for this project because there may be more than one way of performing a task but choosing the correct method can have other benefits such as better performance. For this project the technologies have been devided into three categories, The Mobile technologies, The Server-Side Back End technologies, The Website Front-End technologies.

Server-Side Back End Technologies

- CodeIgniter to provide a MVC framework to help with the development.
- MySQL which will be used to store the system data that will be queried later on.
- MongoDB for the system database and willbe used corrospondingly with the MYSQL
- PHP will be the server script language to be used in this project.
- Linux operating system on the server.
- Apache for the web server

For this project requires reliability, security and performance and the solution of using LAMP (Linux, Apache, MySQL, PHP) which is a certified system with less weakness than more recent back end packages. LAMP is also the ideal choice whhen you look at which languages the students a familiar with.

Website Front-End Technologies

- HTML 5 to define the stucture and content of the website.
- CSS3 for the styling of the website and to provide animation where needed.
- Javascript to provide the website with functionality. The javascript will be used to process the return data from server.
- JQuery will be used with the javascript to provide functionality.

Mobile Application Technologies

Android Application Technologies

- Android studio will be the platform for development.
- Java EE will be used as the programming language with the correct API's.
- RoboGuice to provide dependancy injection framework for android.
- GSON for converting Java objects to JSON representation and vice versa.

A native android application because native android applications provide some of the fastest performance and user interface. Building a native android application will help with how devices may display and content placement. If we are indeed building a native android application, Java will be the primary programming language.

iOS Application Technologies

- Xcode 8/9 as the platform of development.
- Swift3 which is the programming language used to develop Apple applications.
- Objective C will be used inline with the Swift for developing apple applications.
- MapKit framework for providing a map we can change and all the map functionality.
- OpenGL ES GL Kit for the rendering of the systems interface, this will help with the graphics and look of the application.

These form the standard procedure for iOS application development.

0.9 Quality and Feasability Design