



UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA

---

# COS 301 Software Engineering: Architectural Requirements Specifications and Design for the NavUP System

March 12, 2017

## **Group Ruby**

Matthew Perry - u13017030

Unarine Rambani - u14004489

Phuti Setoaba - u13032616

Mankgwanyane Tlaka - u14351872

Victor Twigge - u10376802

Bernard van Tonder - u15008992

GitHub Repository:

[https://github.com/perrymatthew/cos301\\_team\\_ruby](https://github.com/perrymatthew/cos301_team_ruby)

# Contents

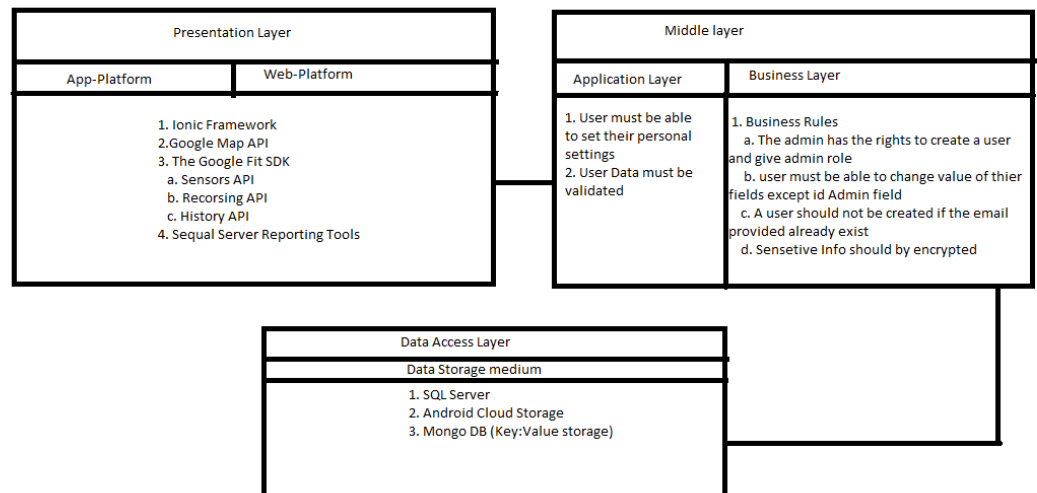
<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	N-Tier Architecture with Technologies . . . . .	3
1.2	Deployment Diagram . . . . .	4
<b>2</b>	<b>Requirements</b>	<b>5</b>
2.1	External Interface Requirements . . . . .	5
2.1.1	User Interface . . . . .	5
2.1.1.1	<b>Login Interface</b> . . . . .	5
2.1.1.2	<b>Search Interface</b> . . . . .	5
2.1.1.3	<b>Navigation Interface</b> . . . . .	5
2.1.1.4	<b>Administration Control Panel Interface</b> . . . . .	5
2.1.2	Hardware Interface . . . . .	5
2.1.3	Software Interface . . . . .	5
2.1.4	Communication Interface . . . . .	6
2.2	Performance Requirements . . . . .	6
2.2.1	<b>Search Feature</b> . . . . .	6
2.2.2	<b>Search Results</b> . . . . .	6
2.2.3	<b>Response Times</b> . . . . .	6
2.2.4	<b>Dependability</b> . . . . .	6
<b>3</b>	<b>Design Constraints</b>	<b>7</b>
<b>4</b>	<b>Software System Attributes</b>	<b>8</b>
4.1	Reliability . . . . .	8
4.2	Availability . . . . .	8
4.3	Security . . . . .	8
4.4	Maintainability . . . . .	8
4.5	Portability . . . . .	9
<b>5</b>	<b>Access Modules</b>	<b>10</b>
5.1	Use Case Diagram . . . . .	10
5.2	State Diagram . . . . .	10
5.3	Activity Diagram . . . . .	11
5.4	Class Diagram . . . . .	12
<b>6</b>	<b>GIS</b>	<b>13</b>
6.1	Use Case Diagram . . . . .	13
6.2	Sequence Diagram . . . . .	14
6.3	Activity Diagram . . . . .	15
6.4	Class Diagram . . . . .	16

<b>7</b>	<b>User Management</b>	<b>17</b>
7.1	Use Case Diagram . . . . .	17
7.2	State Diagram . . . . .	17
7.3	Sequence Diagram . . . . .	18
7.4	Activity Diagram . . . . .	19
7.5	Class Diagram . . . . .	20
<b>8</b>	<b>Fitness</b>	<b>21</b>
8.1	Use Case Diagram . . . . .	21
8.2	Sequence Diagram . . . . .	21
8.3	Activity Diagram . . . . .	22
8.4	Class Diagram . . . . .	22

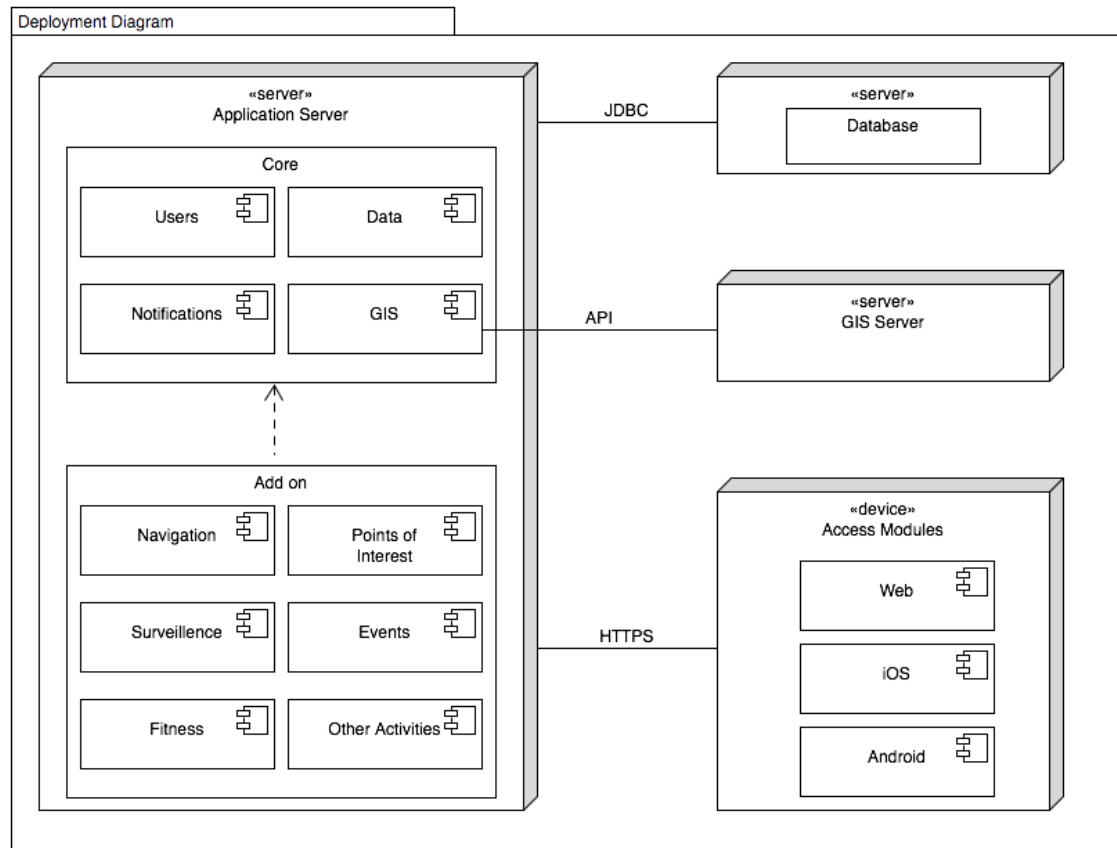
# 1 Introduction

This document describes the Architectural Requirements and the Design Plan for the NavUP on-campus navigation system. The rest of this document is structured as follows: Section 2 covers the External interface and Performance requirements, Section 3 shows what the Design constraints for this project are, Section 4 covers the Software system attributes and finally section 5-8 then delve into the design part of the document for the chosen modules which are, Access Modules, GIS, User Management and Fitness.

## 1.1 N-Tier Architecture with Technologies



## 1.2 Deployment Diagram



## **2 Requirements**

### **2.1 External Interface Requirements**

#### **2.1.1 User Interface**

##### **2.1.1.1 Login Interface**

The user will be presented with a login form with a username/email and a password field. The password field will hide input in order to ensure security. An error message will be displayed if the user inputs incorrect details for these fields. In the case that the user is not registered, a 'Register' option will be provided below the form; clicking on this link will present the user with a registration form.

##### **2.1.1.2 Search Interface**

An interactive search bar will be persisted at the top of the device screen. A user can click on the search bar and enter the name of a location to search for it.

##### **2.1.1.3 Navigation Interface**

Once a user's current and destination locations have been determined, the user will be required to press the 'Navigate' button in order to bring up the Navigation view that will render and display a view of the user's navigation session.

##### **2.1.1.4 Administration Control Panel Interface**

This interface will allow users with Administrator privileges to manage location and event information.

#### **2.1.2 Hardware Interface**

For the mobile applications, the user will need to will need an Android or iOS mobile device. The web portal does not have any hardware interfaces since it will not be developed for specific operating systems. GPS functionality will be handled by the device's built-in GPS application and connectivity to the server would be managed by the underlying operating system.

#### **2.1.3 Software Interface**

The mobile application will communicate with the GPS application in order to acquire spatial data; the application will also communicate with the database server to get information about different locations that are available on campus. The system will need either an operating system or a web browser in order to be operable.

#### **2.1.4 Communication Interface**

Communication between the different parts of the system will be handled by the underlying operating system. Communication with the server will be done using the HTTP/HTTPS protocol over the device's internet/WiFi connections

### **2.2 Performance Requirements**

#### **2.2.1 Search Feature**

The search feature should be prominent so that it is easily locatable for the user.

#### **2.2.2 Search Results**

Results from the search feature should be displayed in a list if more than one location is returned.

#### **2.2.3 Response Times**

- The mobile application and the web portal should be interactive and responsive.
- The system should respond to user input almost immediately.
- Searches must be completed within 2 seconds.
- Acquire the user's location with an accuracy within an 8m radius.
- Display the user's location on a map within 3 seconds after launching.
- Generate routes for users within 2 seconds.
- Update the user's location every second when navigating.

#### **2.2.4 Dependability**

- The system should give the user an indication when executing a process that will require the user to wait.
- The system should display a message if an error is encountered.

### 3 Design Constraints

- User and location information will be stored in a database that will be accessible via the system.
- The system should **always** be available.
- Users will be able to access the system from any mobile device that has Internet browser capabilities or from Android/iOS mobile devices that have the application installed
- Users must be registered and logged-in with correct details in order to perform functions
- The application should not take up more than 10MB(Megabytes) of the devices' disk space
- The GIS module must be limited to the Hatfield campus only
- Guest user's Functionalities must be limited, for all user functionality they must be a registered user
- Admin must only add locations that do not exist in the systems- i.e locations should not be duplicated



## 4 Software System Attributes

### 4.1 Reliability

- The system should generate accurate results for searches. A message should be displayed if a location cannot be found or the search cannot be performed.
- The system should determine user location and display level of accuracy for triangulation.
- The system should generate correct and reasonable routes when provided with two locations.
- Users with administration privileges should be able to create, update and remove locations.
- Notifications should be sent to all registered users via email or SMS.

### 4.2 Availability

- The system should always be available.
- An Internet connection should be available for the application to communicate with the database.
- The application should be connected to the built-in GPS application so that the user's location can be determined.
- Users should be notified if the system is not available.

### 4.3 Security

- Communication between the application and the user database should be encrypted.
- If a user attempts to log in with invalid credentials, they should not be given access and a failure message should be displayed.
- Only users with administration privileges should be able to perform functions that deal with the management of location objects and events.

### 4.4 Maintainability

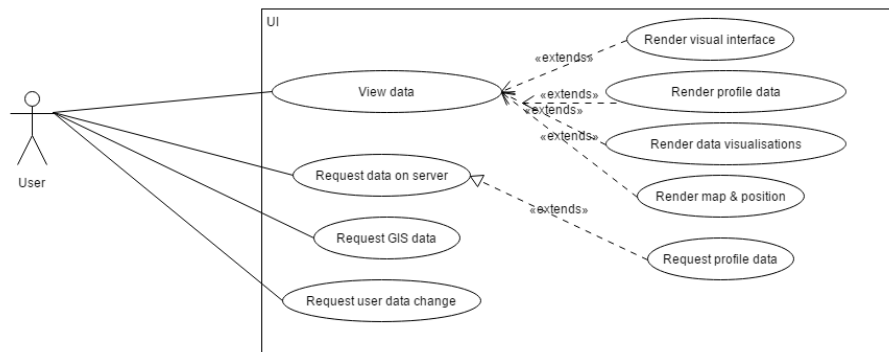
- The system should be implemented with as little coupling as possible, so that it is easy to add new functions and extend it.
- Tests must be built for each module in isolation, in order to determine their functionality.

## 4.5 Portability

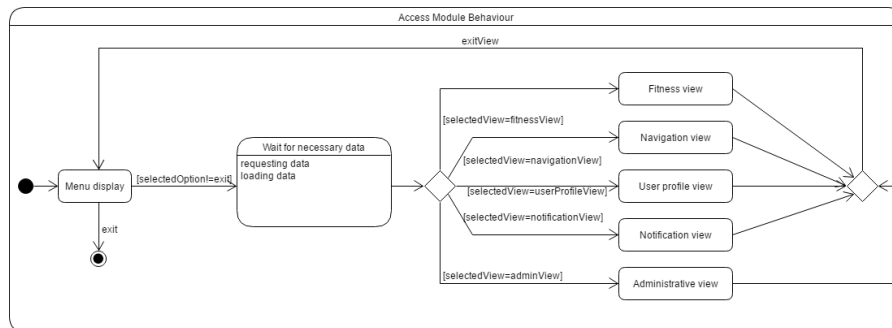
- The application should be portable with both Android and iOS platforms.

## 5 Access Modules

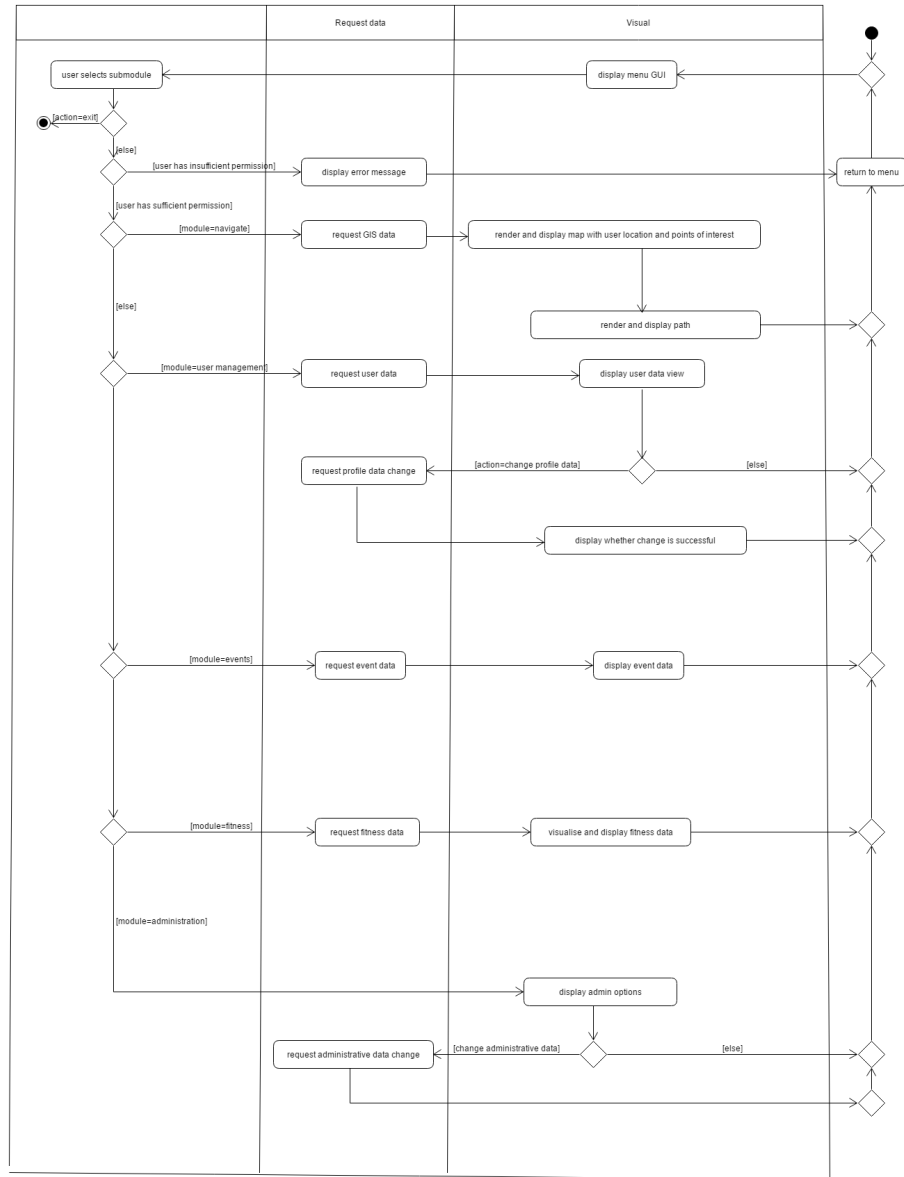
### 5.1 Use Case Diagram



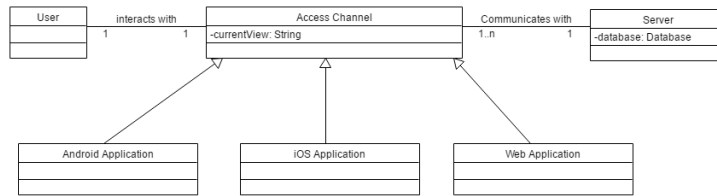
### 5.2 State Diagram



### 5.3 Activity Diagram

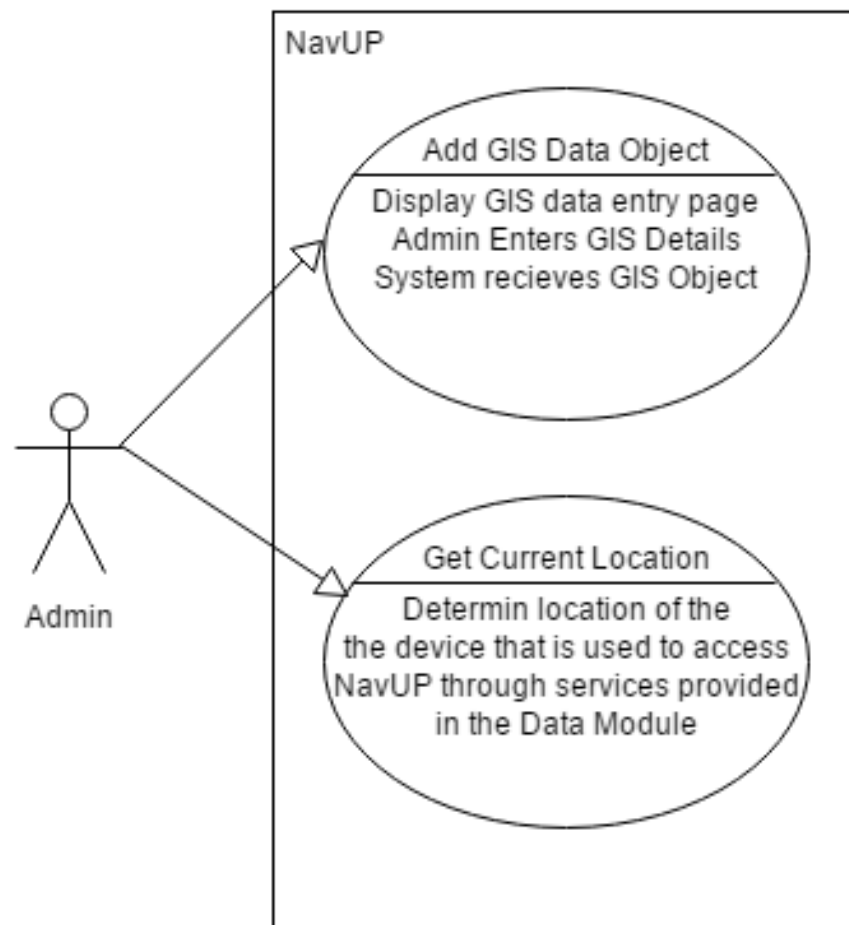


## 5.4 Class Diagram

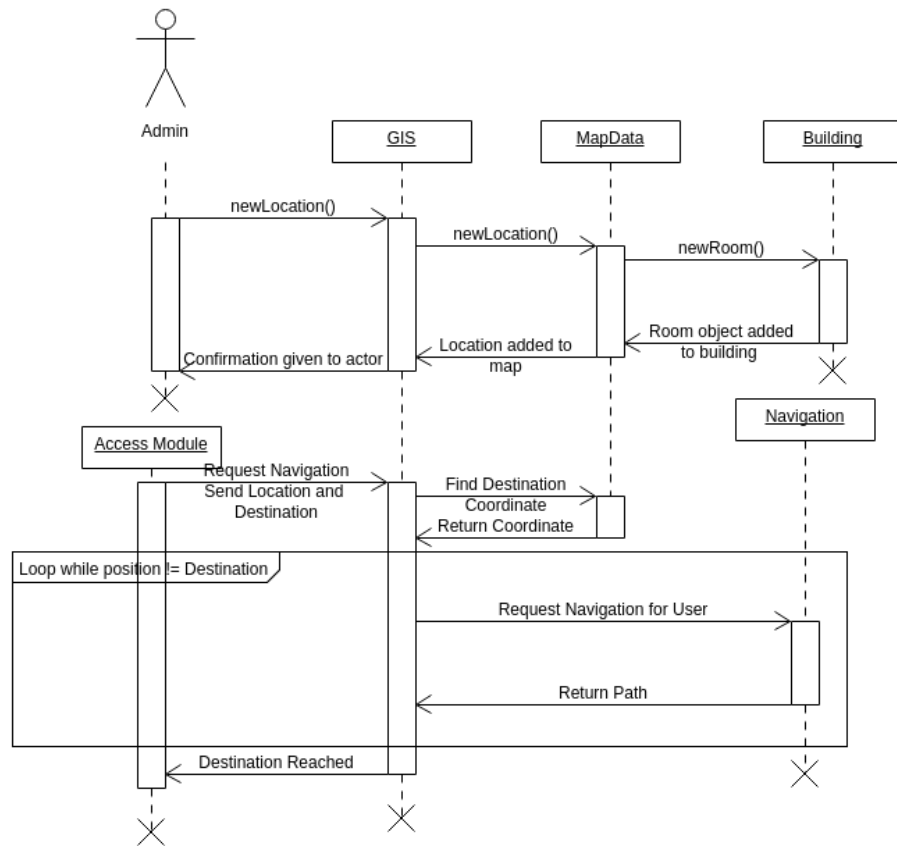


## 6 GIS

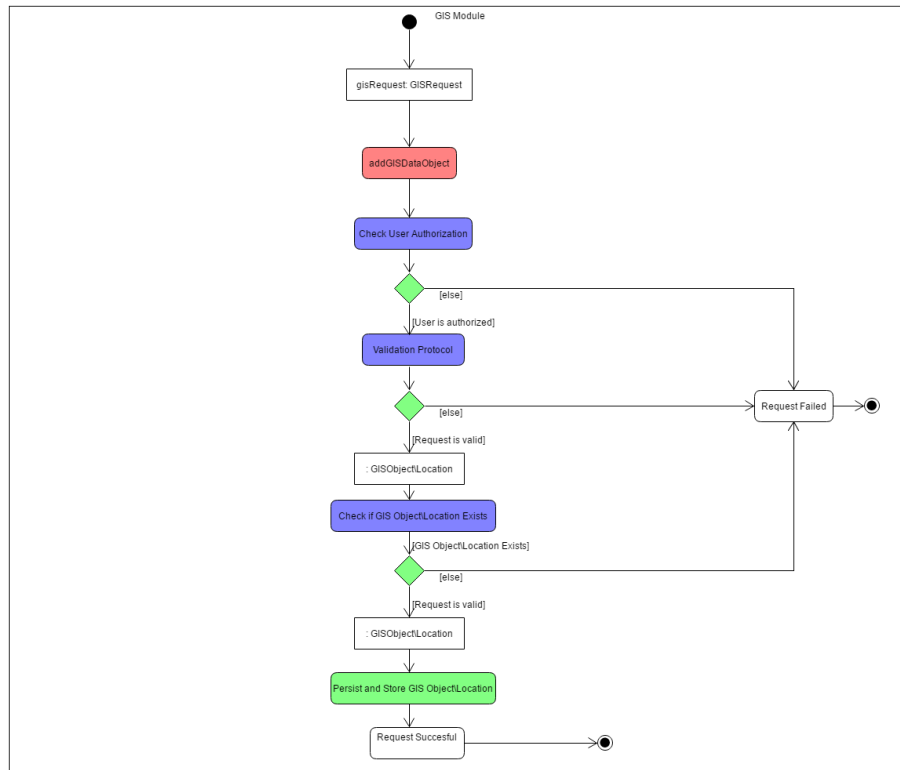
### 6.1 Use Case Diagram



## 6.2 Sequence Diagram

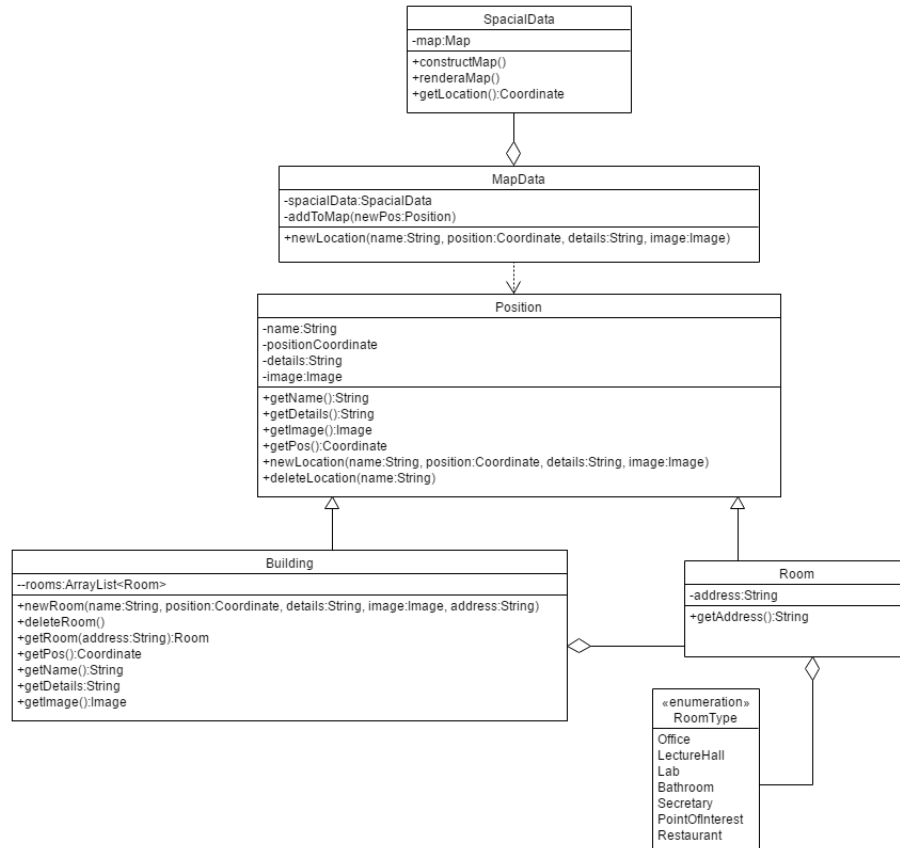


### 6.3 Activity Diagram



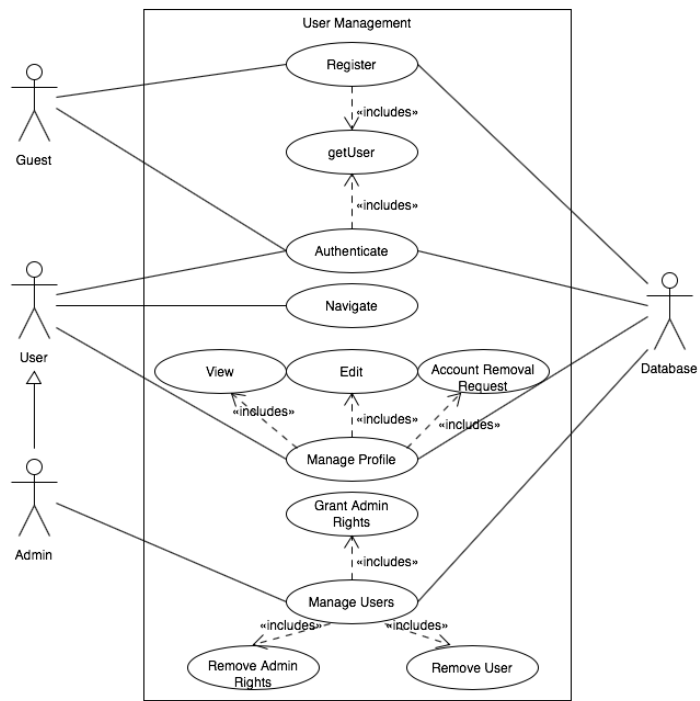


## 6.4 Class Diagram

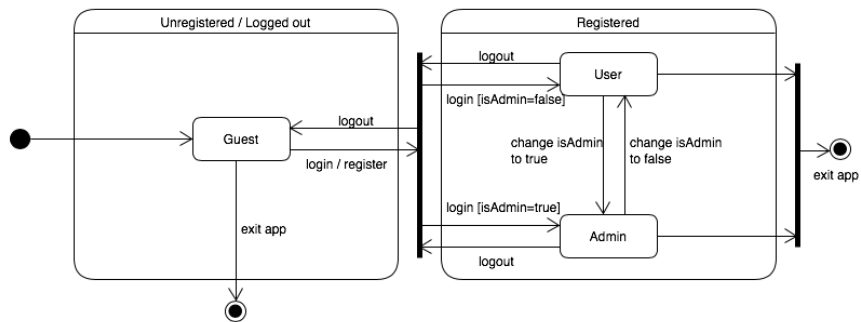


## 7 User Management

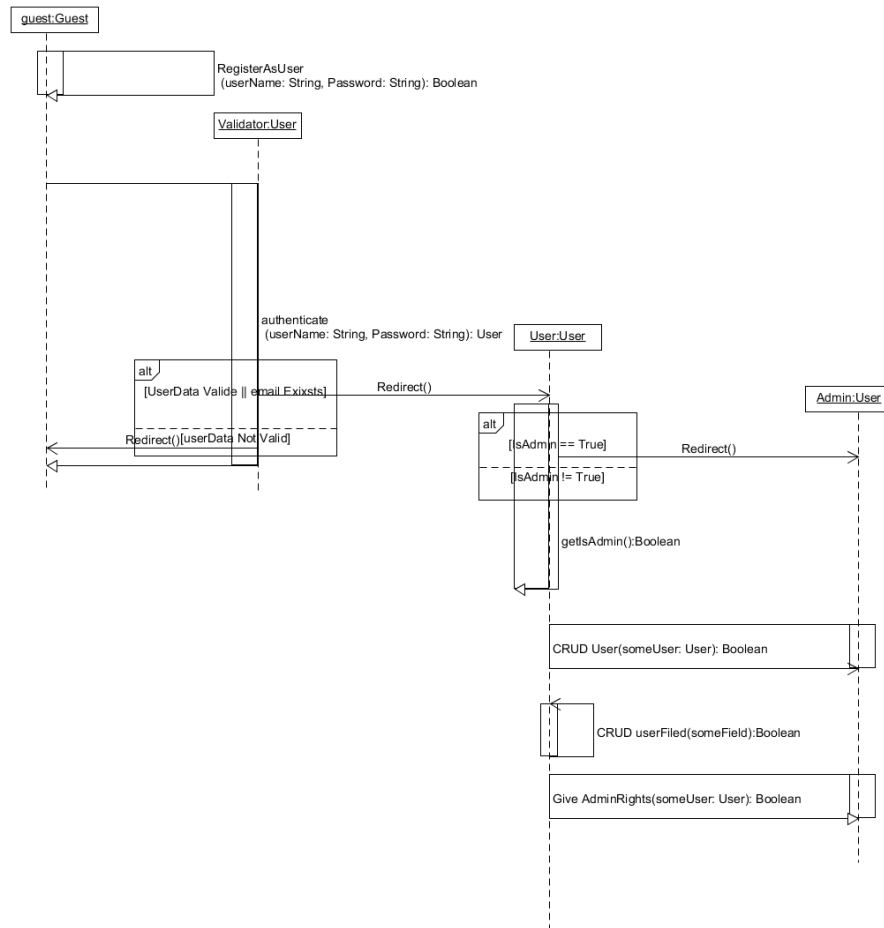
### 7.1 Use Case Diagram



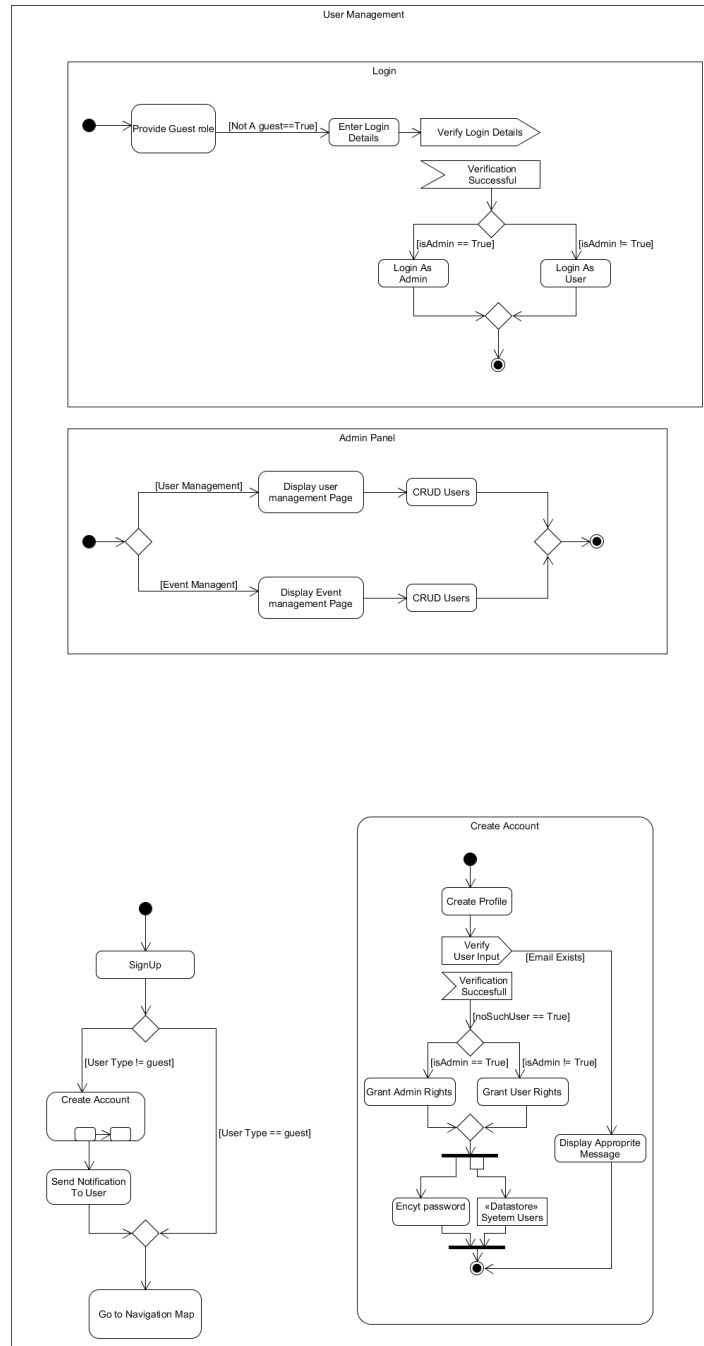
### 7.2 State Diagram



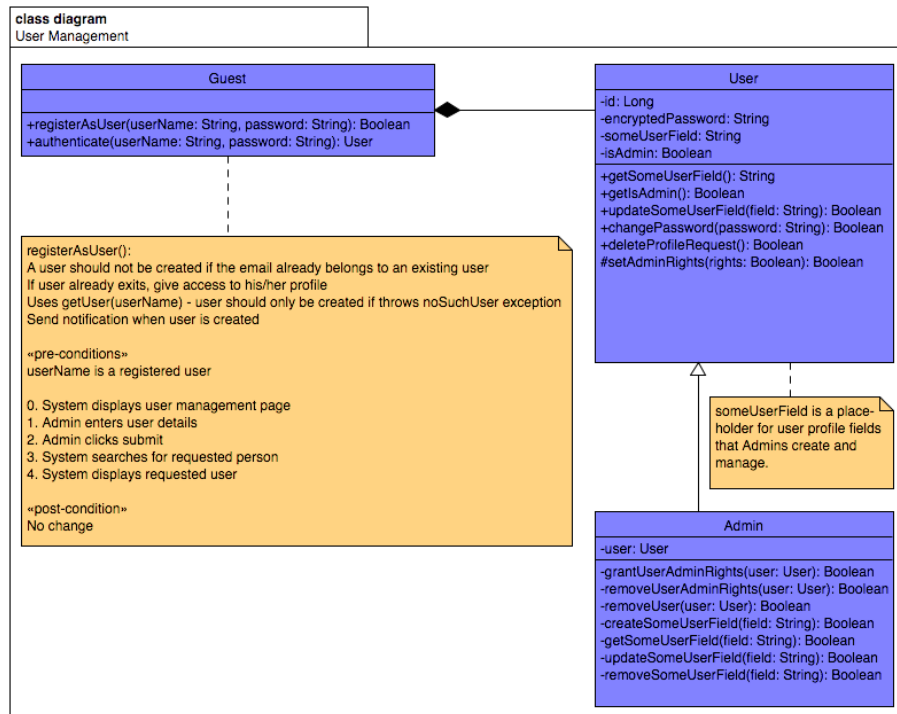
## 7.3 Sequence Diagram



## 7.4 Activity Diagram

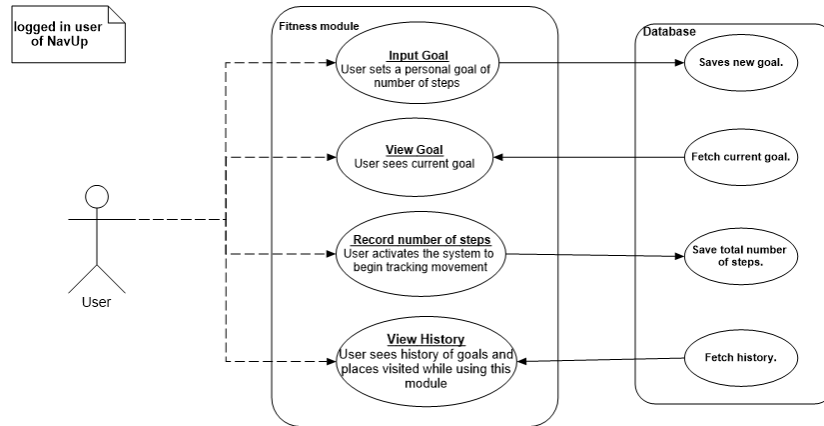


## 7.5 Class Diagram

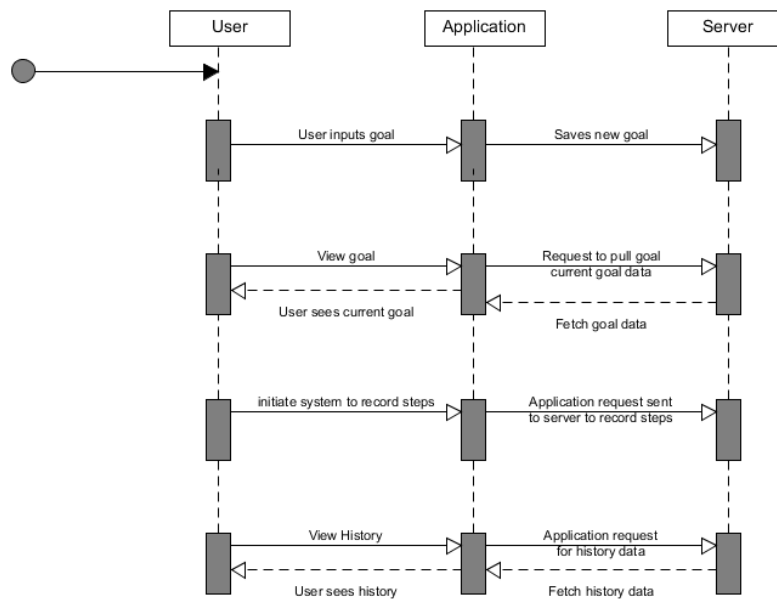


## 8 Fitness

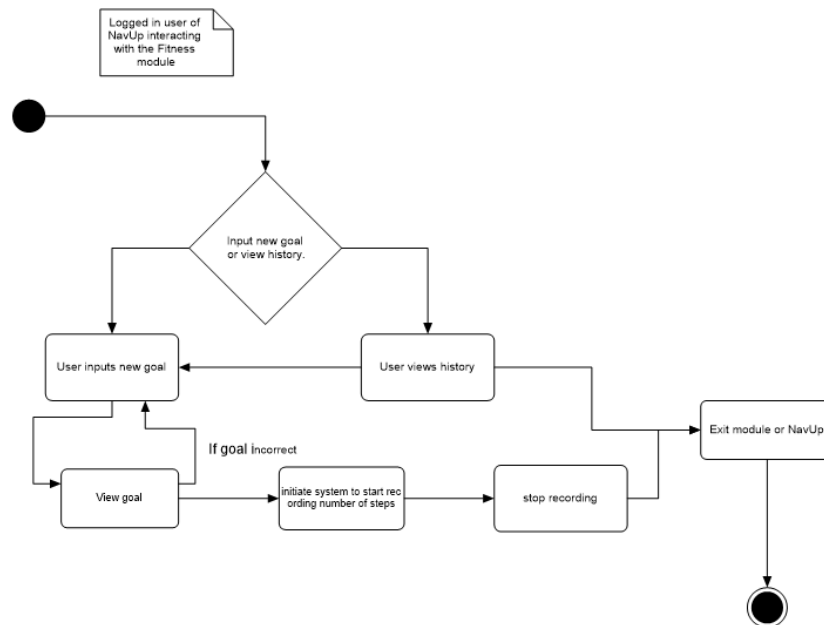
### 8.1 Use Case Diagram



### 8.2 Sequence Diagram



### 8.3 Activity Diagram



### 8.4 Class Diagram

