

---

---

# NavUP Architectural Requirements Specifications and Design

---

---

Compiled By

Andries Jacobus du Plooy - u15226183

Joseph Phutjane Letsoalo - u15043844

Mathapelo Matabane - u12206522

Munyaradzi Mpofu - u15071830

Ritesh Doolabh - u15075754

Midhun John Cheriyan - u17308632

2017

TEAM FORTRAN

### **Abstract**

This documentation covers all the design requirements for the NavUP system, this includes System's External Requirements Performance Requirements, Technology choices and Design Constraints, to name a few. Furthermore, the documentation brings to mind the appropriate design patterns to implement to construct the system.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Purpose . . . . .	4
1.2	Scope . . . . .	4
<b>2</b>	<b>References</b>	<b>4</b>
<b>3</b>	<b>External Interface Requirements</b>	<b>5</b>
3.1	User Interface . . . . .	5
3.2	Software Interface . . . . .	5
3.3	Hardware Interface . . . . .	5
3.4	Communication Interface . . . . .	5
3.5	System Interface . . . . .	6
3.6	Memory . . . . .	6
<b>4</b>	<b>System Features - Modules</b>	<b>7</b>
4.1	User . . . . .	7
4.1.1	Scope . . . . .	7
4.1.2	Design details . . . . .	7
4.2	Notifications . . . . .	7
4.2.1	Scope . . . . .	7
4.2.2	Design details . . . . .	9
4.3	Points Of Interest . . . . .	12
4.3.1	Scope . . . . .	12
4.3.2	Design details . . . . .	12
4.4	Events . . . . .	12
4.4.1	Scope . . . . .	12
4.4.2	Design details . . . . .	12
<b>5</b>	<b>Other Non-functional Requirements</b>	<b>13</b>
5.1	Performance Requirements . . . . .	13
5.2	Design Constraints . . . . .	13
5.3	Software System Attributes . . . . .	13
5.3.1	Reliability . . . . .	13
5.3.2	Availability . . . . .	13
5.3.3	Security . . . . .	13
5.3.4	Maintainability . . . . .	13
5.4	Technology choices . . . . .	13
<b>6</b>	<b>Appendix A - Glossary of Terms and Definitions</b>	<b>14</b>

# 1 Introduction

Information about the activities, points of interest and venues on campus is not readily available to employees, students and visitors to the UP campus. This is especially a problem at the beginning of the academic year when thousands of new students come to campus to attend regular lectures and on open days when large numbers of visitors have to find their way to exhibitions and service points.

## 1.1 Purpose

NavUP is aimed at providing access to a wealth of information about campus activities, points of interest and venues via the campus WiFi network. The information should be accessible using smart devices. Users should be able to use the information to navigate on campus and be informed about activities and facts of their interest.

## 1.2 Scope

The core of the system is a navigation tool which is enriched with functionality receiving information of interest.

# 2 References

Goldman, J.L., Abraham, G. and Song, I. (1998) Generating Software Requirements Specification (IEEE Std. 830 1998) document with Use Cases. Available at: <http://www.pages.drexel.edu/~sga72/docs/SRSwithUseCases.pdf> (Accessed: 24 February 2017).

Singh, E., Pieterse, V., Omeleze, S., Theunissen, M. (2017) 2017 Class Project NavUP. Available at: <http://www.cs.up.ac.za/courses/COS301/NavUPRequirementsSpecification.pdf> (Accessed: 27 February 2017).

## **3 External Interface Requirements**

### **3.1 User Interface**

The subsystem will allow for notifications to be pushed to registered users in the form of E-mails. The E-mails will have a descriptive subject associated with it and will be accessible from any device that allows a form of e-mail service. The subsystem will allow for notifications to be pushed to registered users in the form of SMS's. the SMS's will only have a senders name which will indicate the type of notification and the SMS's content will then go on to expand on the notifications description. The subsystem will interact with users mostly on a cellular device and a colour screen will be required to get the most out of the application in terms of the E-mail service, SMS's on the other hand are a vital part of most cellular devices and in most cases no extra functionality is required.

The system must be able to help the user locate buildings by the use of display, navigate and select functions over the system's built GUI. The system shall provide help options to the user when locating a certain POI.

### **3.2 Software Interface**

The application will require an e-mail and SMS application in order to receive notifications of this type. The subsystem will need to have access to a database of users such that they can be added to the user lists for the particular type of notification/s of their preference. These lists will then be referenced when the notifications are being sent to the users. A programmatic interface for operations such as adding locations, modifying locations, deleting locations is required.

### **3.3 Hardware Interface**

With the use of most cellular devices, a color screen will be required for the use of e-mails and the device must in all cases be able to display information in a way that is understandable by the user. The device must have the appropriate hardware in place to allow a form of data connection between itself and the Internet. The system needs a database of all the WI-FI hot spots around the campus to map out the location of each Point of Interest. Moreover, the system must update information pertaining to points of interest.

### **3.4 Communication Interface**

The subsystem will need to communicate with any other subsystem that wishes to create and send out notifications to the users. The subsystem will need to communicate with an automated E-Mail service to send out the mass e-mails to the users. Each notifying module will have access to its own instance of the notification subsystem to prevent any notifications from different notifying

modules causing any conflict, and this also allows for no waiting period between creating and sending of notifications. They will be delivered in real time. The subsystem will need to communicate with an automated SMS service to send out the mass SMS's to the users. Each notifying module will have access to its own instance of the notification subsystem to prevent any notifications from different notifying modules causing any conflict, and this also allows for no waiting period between creating and sending of notifications. They will be delivered in real time. The users will need to have the appropriate communications interfaces available to allow them to receive notifications. This would include a working cellphone SIM card with a registered cellphone number and/or a working e-mail account that is accessible on a device with a data connection of some form.

### **3.5 System Interface**

### **3.6 Memory**

## **4 System Features - Modules**

### **4.1 User**

#### **4.1.1 Scope**

The user management module is responsible for maintaining information about the registered users of the system, including the authority levels of each user. Administrators can manage information about venues and activities whilst users who have signed up may request services from the various modules and persist private information related to particular services.

No information is stored for the guest user. When accessing the service, the user assumes the guest role without being logged in. The guest user may use public services and may register or log in. When a user is registered, the fields shown in the domain model are stored for the user using a unique automatically assigned ID. The user provides all other fields. The password should be stored in encrypted format. The user may change the value of any of the fields of his own record except the value of the isAdmin field. The only difference between a User and an Admin is the value of the isAdmin field. Only Admin users may change the value of the isAdmin field of other users. There are three types of users: Admin, User and Guest.

#### **4.1.2 Design details**

### **4.2 Notifications**

#### **4.2.1 Scope**

The notifications module provides notifications to system users regarding particular system updates that a user would like to be notified about through some medium external to the application.

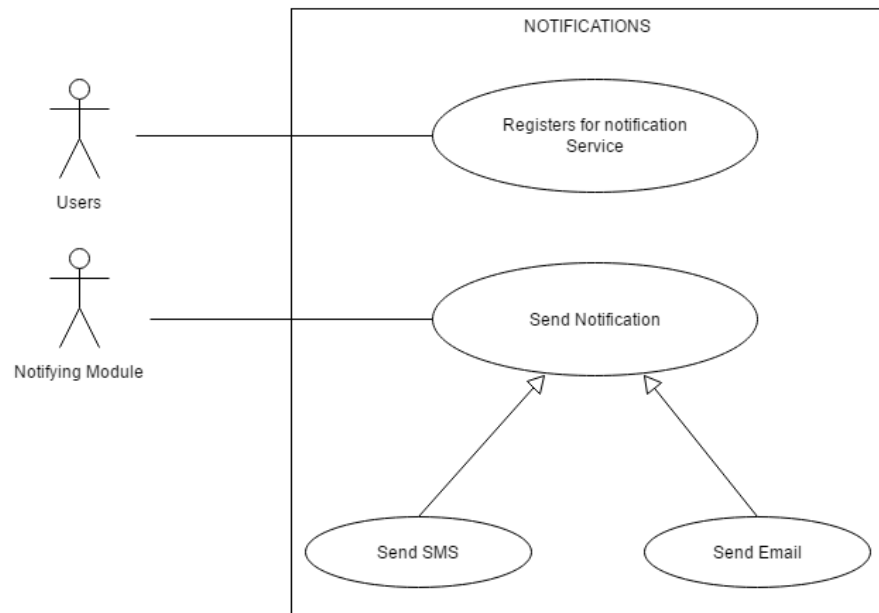


Figure 1: Notification Use Case Diagram



## 4.2.2 Design details

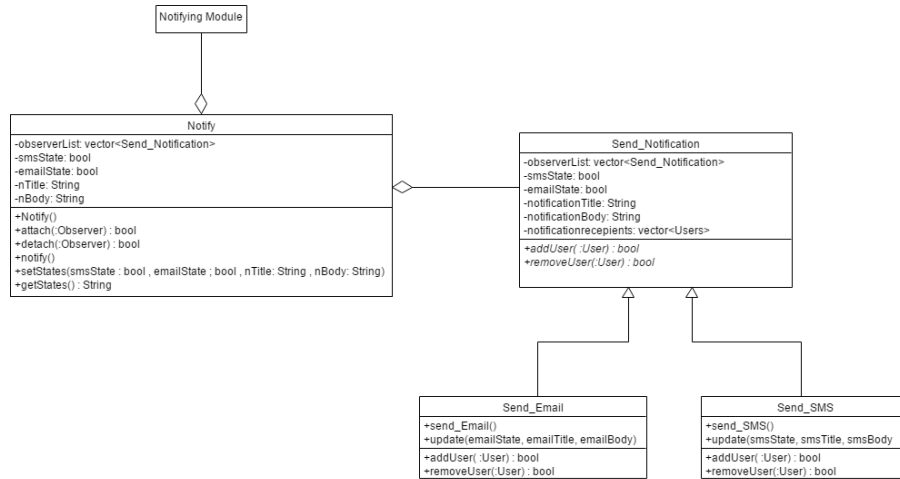


Figure 2: Notification Class Diagram

For the notifications subsystem the observer design pattern came to light. This pattern allows objects to change state and all of its dependents follow suit. This allows a notification state to be used that can update whether or not a notification needs to be sent out. In this case the push method of observer was used. Also evident is the template design pattern that uses the concrete observers to redefine the way in which the notifications are sent. Notifying modules are also given the flexibility to decide which medium(s) are used for the notification.

There are methods in the Send Email and Send SMS classes that allow users to be added to the user lists for the notifications to either or both of these mediums. There is also provision for more notification mediums to be added in the future and these can be attached to the Notify class. Should a notification medium be unavailable it can be detached from the Notify class until it is available. The Notify class can be called from any other module that would like to request a notification to be sent out to users which allows for modularity.

Notify: Subject

Send Notification: Observer(Observer) / Abstract Method(Template)

Send Email: Concrete Observer1(Observer) /Concrete Method(Template)

Send SMS: Concrete Observer2(Observer) / Concrete Method(Template)

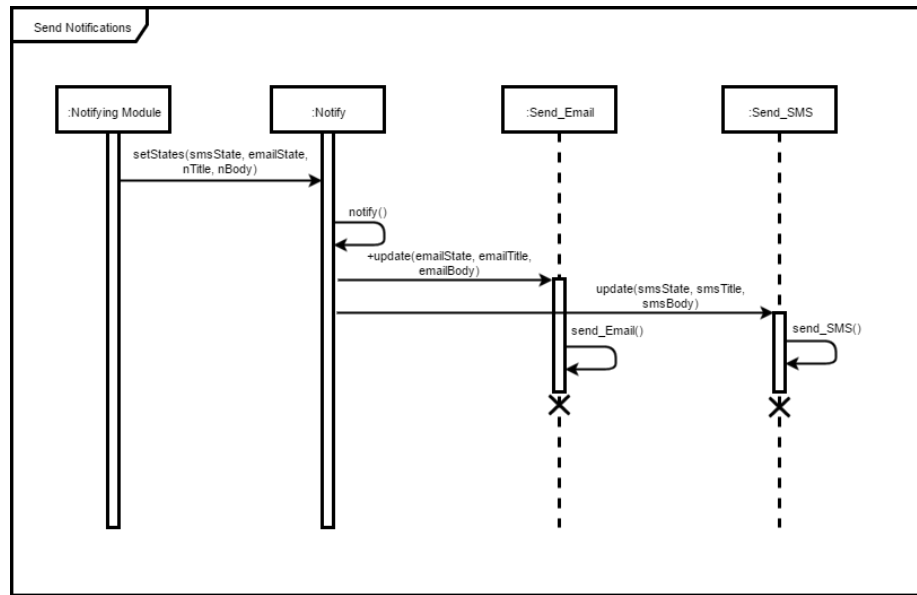


Figure 3: Notification Sequence Diagram

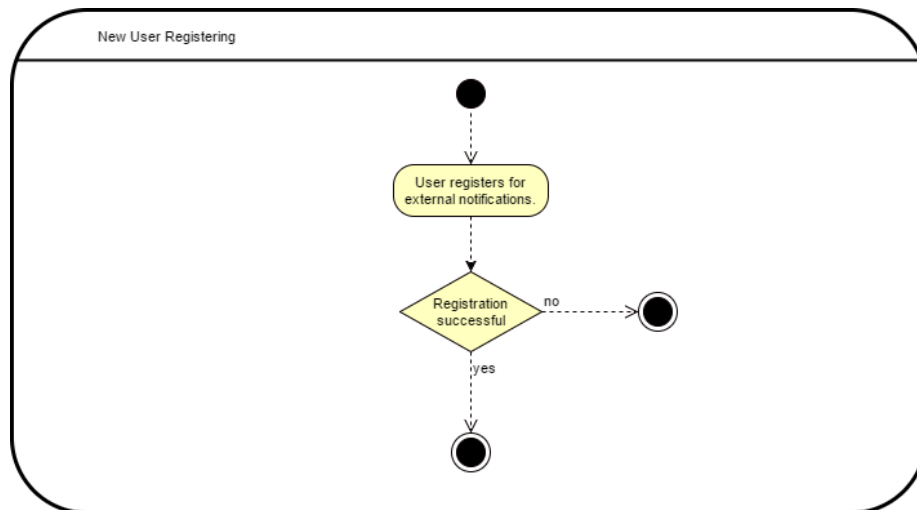


Figure 4: Add User Activity Diagram

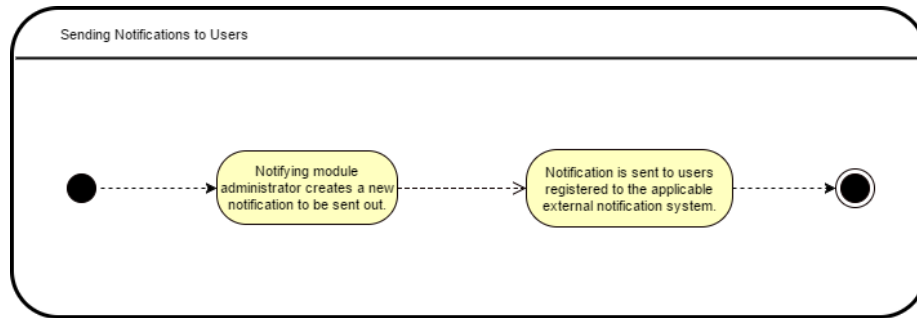


Figure 5: Send Notification Activity Diagram

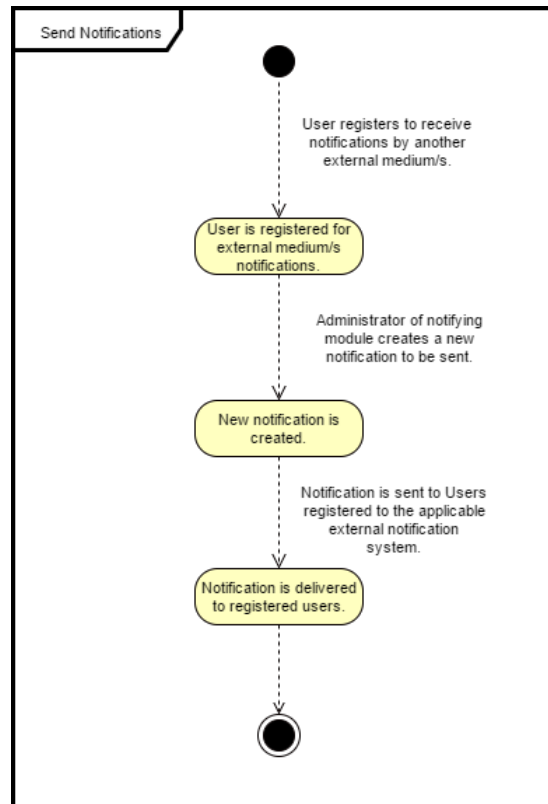


Figure 6: Notification State Diagram

### 4.3 Points Of Interest

#### 4.3.1 Scope

#### 4.3.2 Design details

### 4.4 Events

#### 4.4.1 Scope

#### 4.4.2 Design details

## **5 Other Non-functional Requirements**

### **5.1 Performance Requirements**

The notifications will have to be pushed out in a timely manner such that the time between creation and time of sending of notifications is minimal.

The system will also have to handle sending bulk e-mails and SMS's from multiple notifying modules in a manner that will not cripple the system or put the system at risk.

The system must accommodate any number of users at any given time period. All page requests must be processed by the system in no more than the average response time of the Internet connection.

The system must display confirmation of user log ins, as well as display favorite Points of Interest relative to the user profile.

### **5.2 Design Constraints**

Notifications should not be long such that loading times are affected(e-mail) or that partitioning of messages occurs(SMS). Notifications should be pushed out within a given time frame. Notifications should conform to the rules of the medium of transport. No unnecessary data/information should be sent in these notifications.

### **5.3 Software System Attributes**

#### **5.3.1 Reliability**

The notifications subsystem is crucial in informing users of possible information. With this it is of utmost importance that all notifications sent are received by the registered users.

#### **5.3.2 Availability**

The service must be available to all users and administrators of notifying modules at all times.

#### **5.3.3 Security**

Notifications and communications must be secure such that no malicious activity occurs on the system and users are not conveyed wrong information.

#### **5.3.4 Maintainability**

The amount of notification mediums is increasing and to accommodate this new forms of communication mediums must be easily appended to the system.

### **5.4 Technology choices**

## **6 Appendix A - Glossary of Terms and Definitions**