UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

NavUp Architectural Design

March 12, 2017

Team Java

Dail Nathan Jonker - u13325630
Kulani Bamuza - u15008402
Yashvir Brijlal - u14387744
Bayanda Chakuma - u11283212
Xolo Dandashe - u14245681

Contents

1 Specific Requirements

1.1 External Interface Requirements

1.1.1 User Interface

The UI(user interface) of NavUp will be designed to be user friendly. This is required to make the mobile application easy to use for the user. This will in turn reduce the time it takes the user to get accustomed to the mobile application and it's functionality. The UI will offer the base functionality to both a guest and logged-in user. A guest user will be able to open the application and use the navigation system of NavUP like any other user. It will display the various points of interest and activites as well for guests when they visit campus. Activities that might interest a guest to campus will be highlighted on the map itself. A user will be able to easily log in or register to gain access to further functionality of the system. The UI will provide the user with the necessary tools and information to navigate in and around campus grounds. The user will require valid login details to be able to log in to the system. Certain users will have administrative rights allowing them to edit data of the NavUP system. These users will be adding and removing data as required.

1.1.2 Hardware Interface

The hardware required will have to be in the form of a smartphone/tablet. The system will be designed with the idea of it being mobile. The system will thus be developed on a mobile platform for it to be downloaded as an application to be used on a smartphone or tablet. This will allow the system to be used on the go. Most students are on the move meaning they need hardware that can track and navigate them in real time. The wifi hotspots will be a key feuture when GIS is not being used.

1.1.3 Software Interface

The mobile application itself will be designed to function on the Android and IOS mobile platforms. These platforms have touchscreen capabilities, allowing the user to use the system as any other application on their smartphone. The built in software of each platform allows the system to be compacked, users will be able to easily navigate and search various locations by using the on-board keyboard of the smartphone. GIS and wifi will allow the software to locate there position and navigate a route for students aswell as provide relevent information on the location they are at.

1.1.4 Communications Interface

As stated in the SRS document, the system will be using GIS navigation. It will also make use of wifi-signal strength to determine a user's position depending on the various wifi hotspots on campus. A user will make use of these technologies

in various ways, but they need to be connected to the system in some form to gain access to it, this includes mobile data and wifi. Seeing as campus has wifi available to guests and students/staff, this allows for easier communication between the smartphone app and the system itself. This will also help save mobile data for the users when navigating or searching for locations on campus. The system will also communicate to the user in the form of notifications on the application itself and via email when certain tasks are done by the user.

1.2 Performance Requirements

The mobile version of NavUp needs to funtion like any other mobile application. There should not be any indication of delay when moving between interfaces and using the functionality of the application. The only delay that can be acceptible is the connection to the system or determining a user's location. This can be caused by a user's smartphone that has a weak signal/connection to the system. This can also happen when connection is lost or interrupted. The application itself needs to opperate smoothly on any smartphone device that can support it. This means that it needs to be designed with old smartphones in mind, allowing all smartpfone users to use it without any delay. Position needs to be calculated as fast as possible including navigating a path for a user. The end point will be static information saved in the database, this needs to be fetched as quick as possible for the route to be calculated. Unnecessary delay may be frustrating to the user.

1.3 Design Constraints

The mobile application cannot be too large in size. Users sometimes have limited space left on their smartphone device, this will then require the application to be as small as possible while stil maintaining propper functionality. The static data will be stored on a database along with any user information on a server which will reduce the size drastically. The map itself will be the largest part of the application, so care needs to be taken of what detail and size the map will consist of. The remainder of the apllication must take care not to overstep the size boundry(This will be set at a later stage, most likely in implementation).

1.4 Software System Attributes

1.4.1 Realiability

The system itself will never crash or hang, excluding an operating system error. The system will provide alternatives when connection has been lost or when there is a delay when connecting. The system will continue to try and re-connect when connection has been lost.

1.4.2 Availability

The system will be available to all users that have access to a smartphone or tablet. A web based version will also be available to administrative users at all times.

1.4.3 Security

The user's information will be kept confidential and will not be used for other purposes than the system itself. Only administrative users will have access to this information. Information like passwords will be encrypted before it is saved on the database.

1.4.4 Maintainabilty

The code of the system will be modular to allow for future modifications to improve or add on to features. The code itself will be documented and authorship will be added for refrencing purposes. The system itself and the database will be maintained and monitored by the administrative users.

1.4.5 Interoperability

User devices will communicate with the system, exchanging information required for the functionality of the system. User devices will use this information to display relevant and required data to the user and in turn provide feedback to the system when updates are made.

2 UML Diagrams

In this section of the document the various diagrams of the sub-systems will be illustrated. The main diagrams will include a Use Case and Class diagrams. State, Sequence and Activity diagrams may be included to expand on non-trivial functionality.

2.1 Users

The following Diagrams include a Use Case and Class diagram for the User sub-system:

Figure 1: User Use Case Diagram

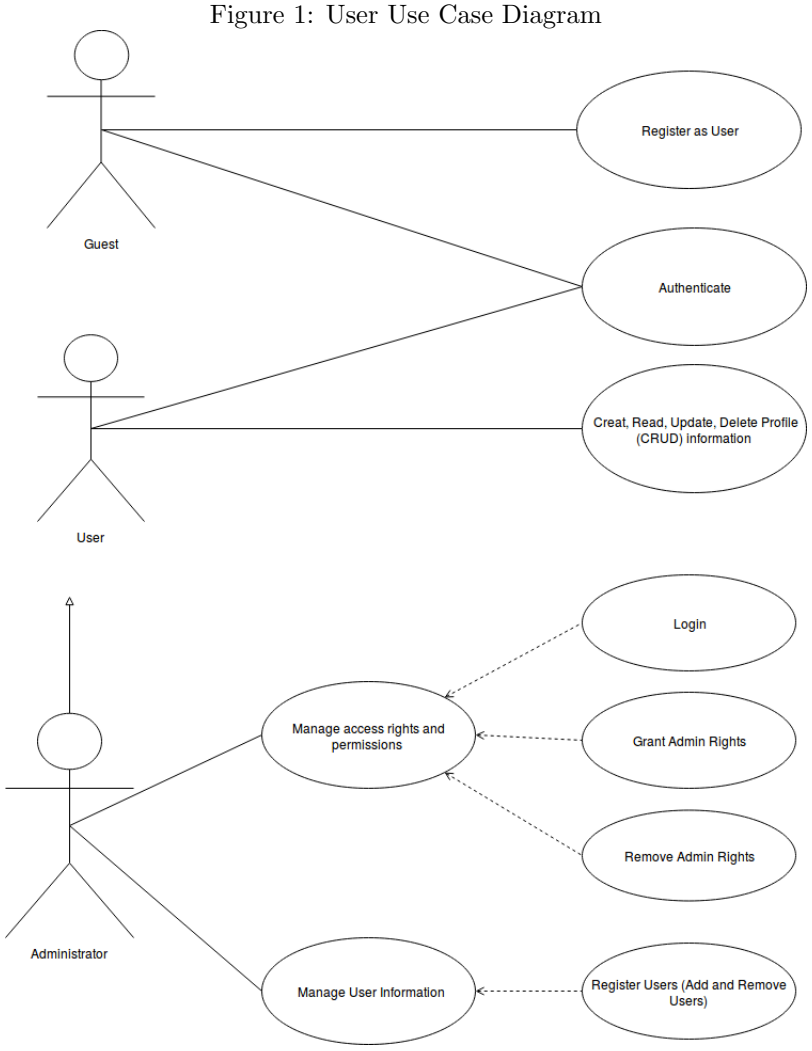
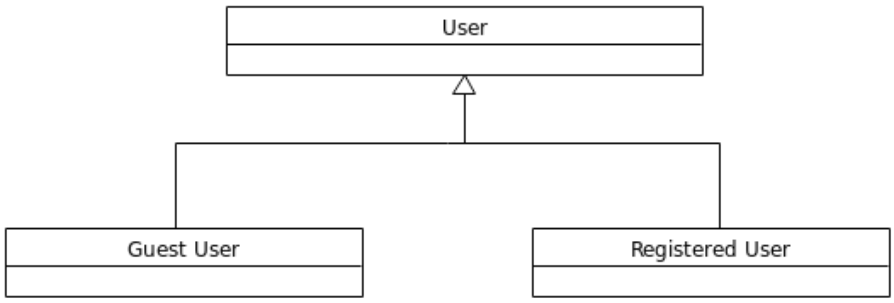


Figure 2: User Class Diagram



2.2 Navigation

The following Diagrams include a Use Case, State and Class diagram for the Navigation sub-system:

Figure 3: Navigation Use Case Diagram

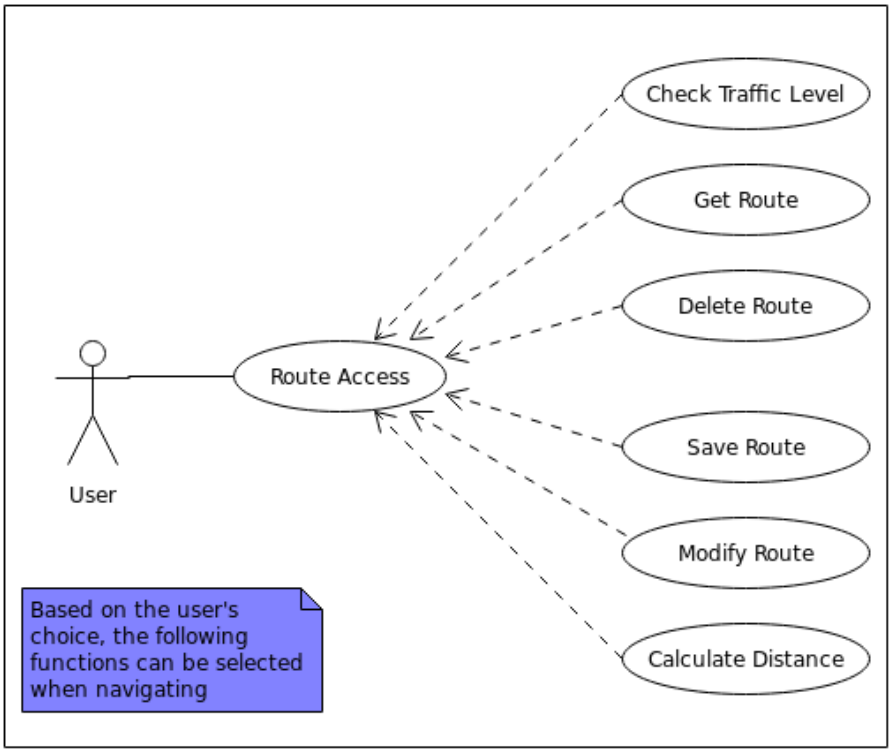


Figure 4: Navigation Class Diagram

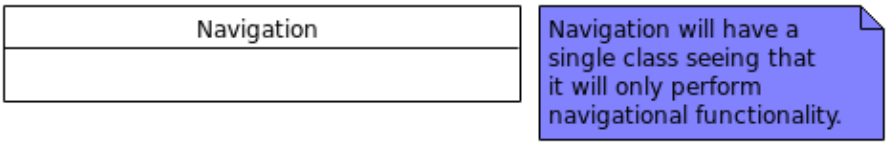
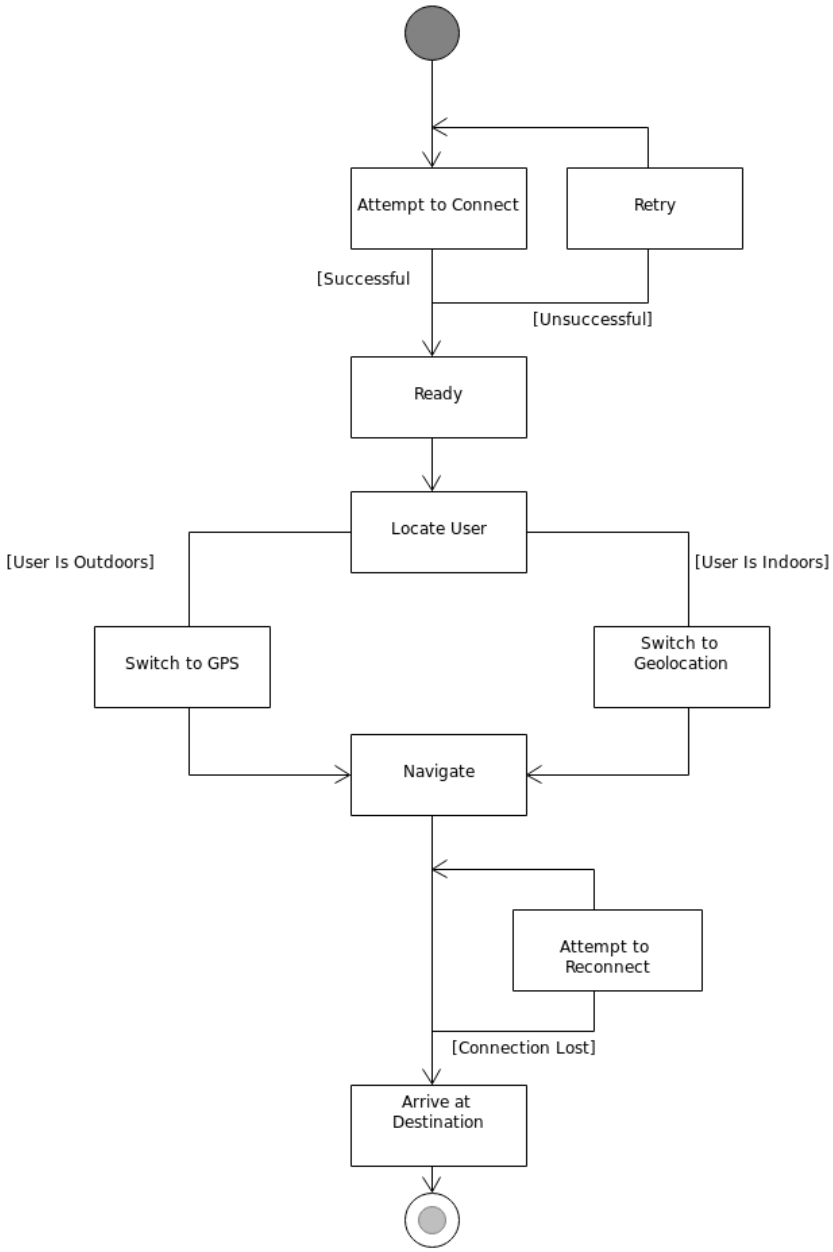


Figure 5: Navigation State Diagram



2.3 Notification

The following Diagrams include a Use Case, Sequence and Class diagram for the Notification sub-system:

Figure 6: Notification Use Case Diagram

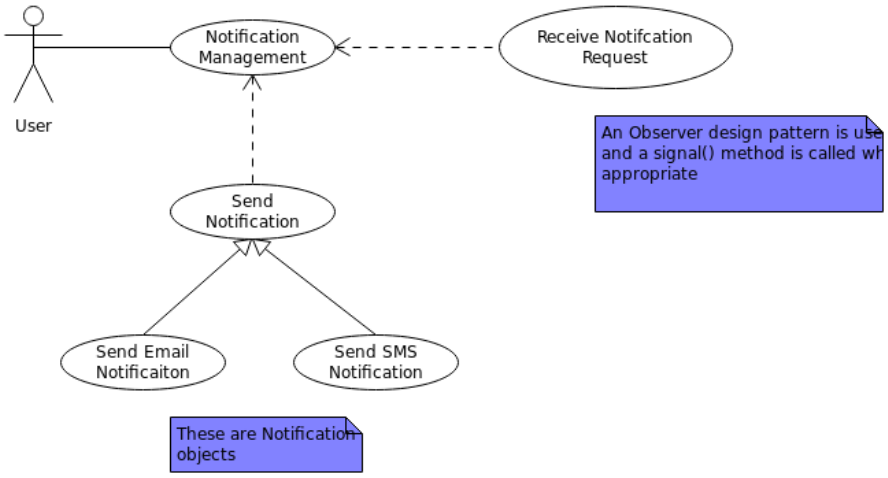


Figure 7: Notification Class Diagram

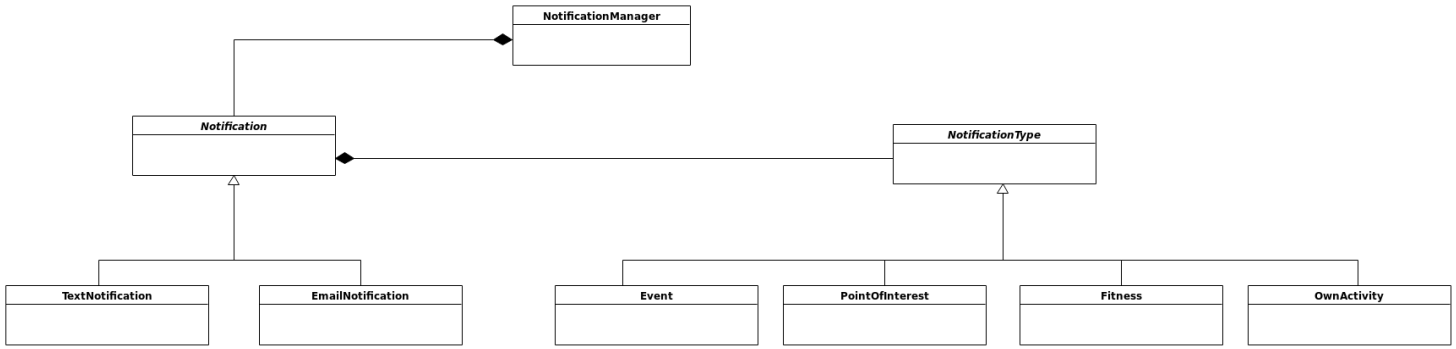
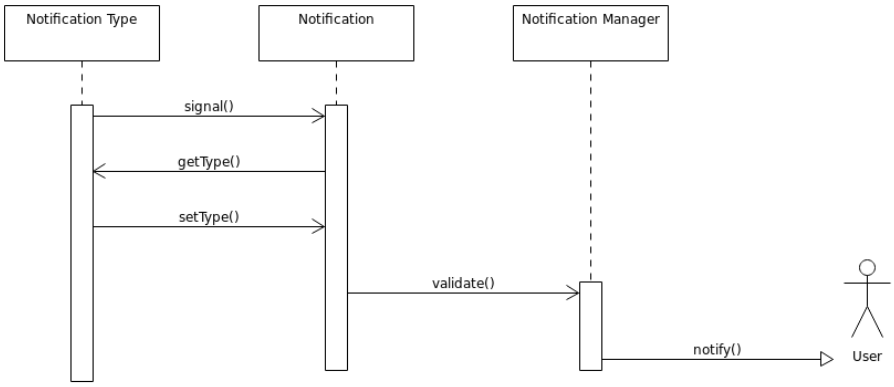


Figure 8: Notification Sequence Diagram



2.4 Points of Interest

The following Diagrams include a Use Case and Class diagram for the Points of Interest sub-system:

Figure 9: Points of Interest Use Case Diagram

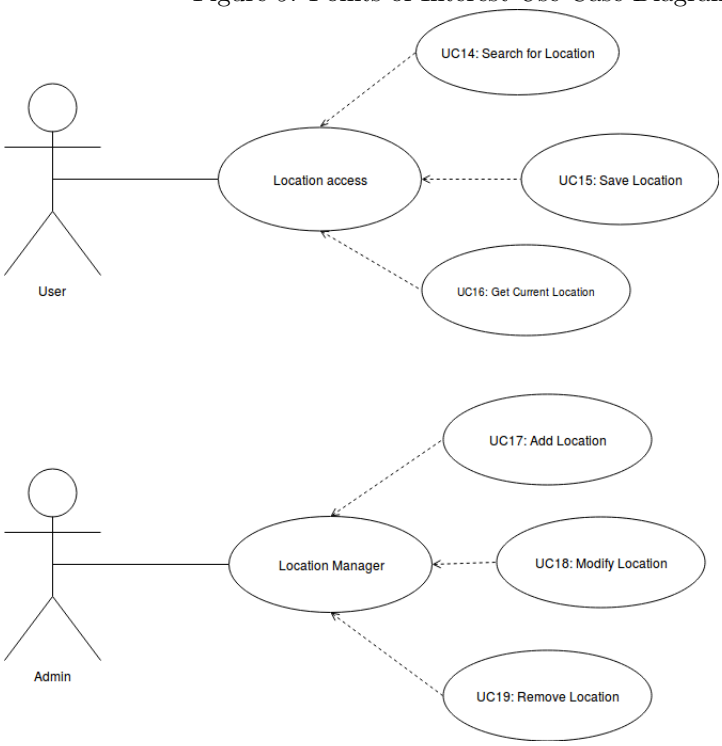
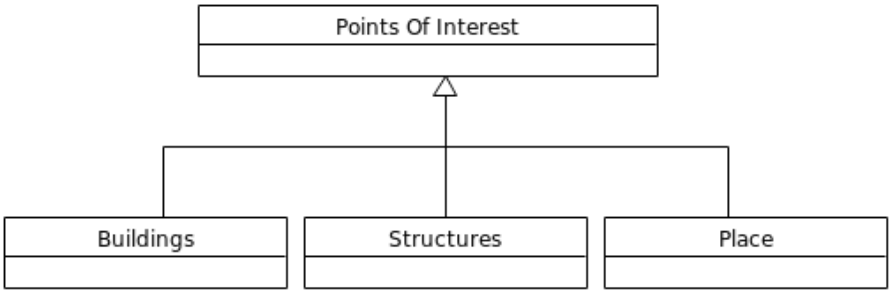


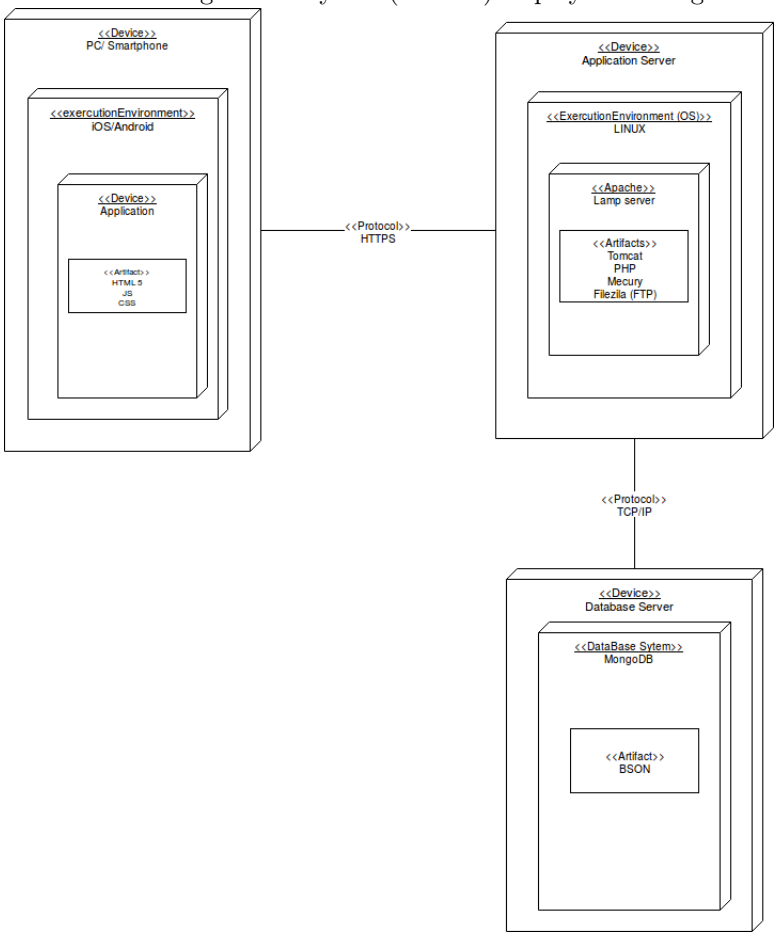
Figure 10: Points of Interest Class Diagram



3 Deployment Diagram

The following Deployment diagram will illustrate the system as a whole, visualizing hardware, platforms and the software that runs on it:

Figure 11: System(NavUP) Deployment Diagram



4 Design Pattern

The following will explain the various design patterns chosen:

Figure 12: Singleton Design Pattern

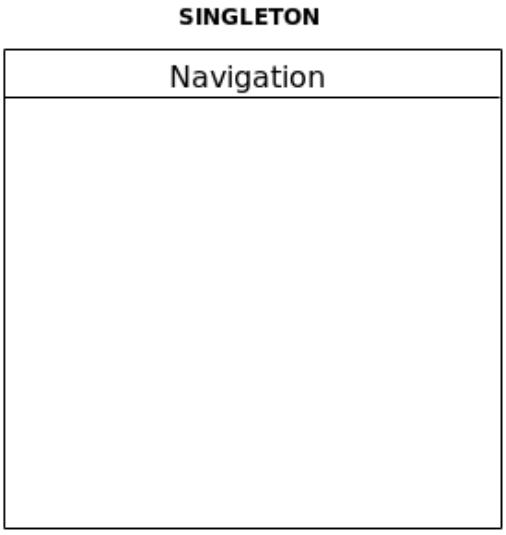


Figure 13: Facade Design Pattern

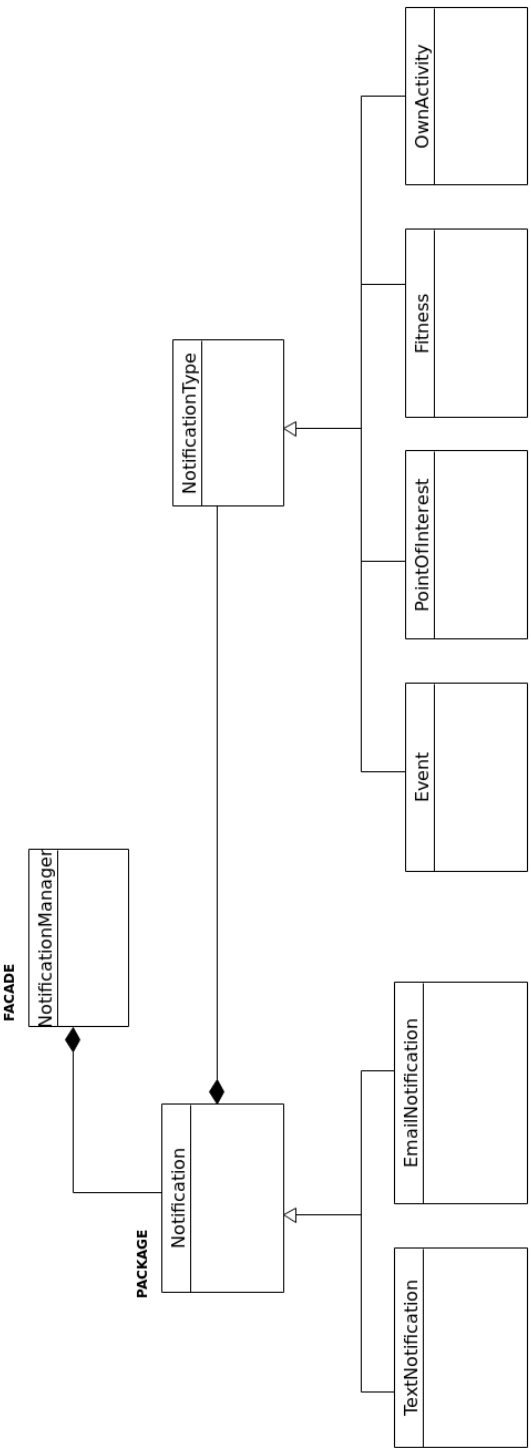
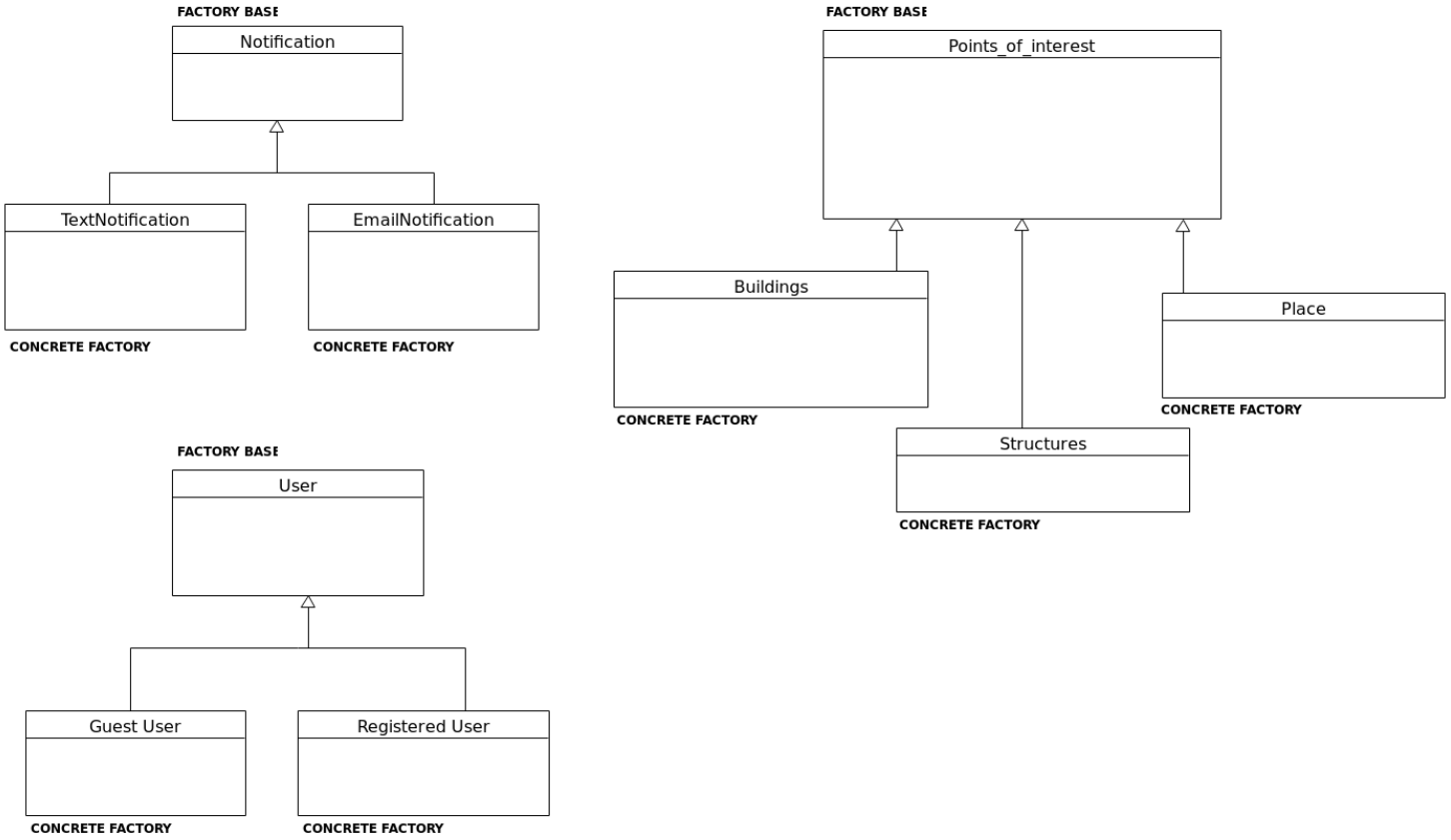


Figure 14: Factory Design Pattern



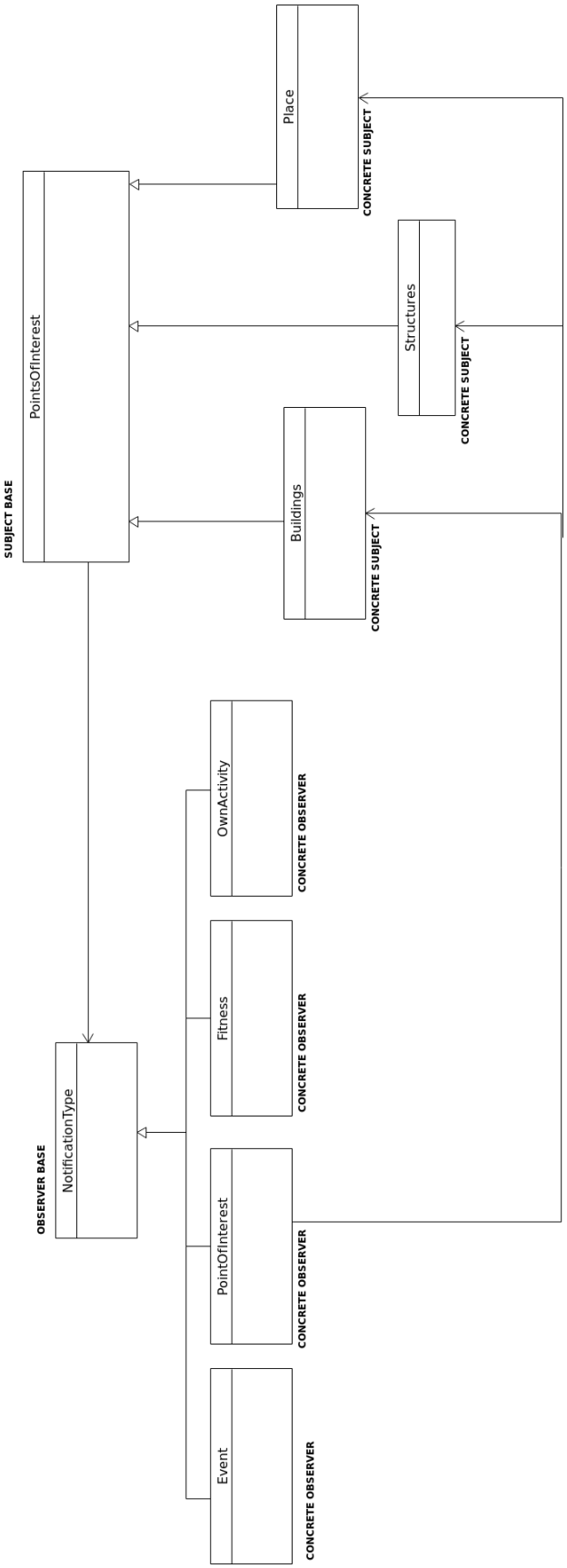


Figure 15: Observer Design Pattern

5 Technologies

The application makes use of various technologies in order to implement its functionality. These technologies include, but are not limited to, back end programming languages, web design technologies, web development technologies, navigation technologies, networking technologies and database technologies. The technologies used are based on the implementation designed and specified within this document.

5.1 Platform

The two most popular mobile platforms by far are Apple and Android. As such, the system will be implemented on these platforms as to cater to the vast majority within the user base.

- iOS
- Android

5.2 Application Programming

Parts of the system's backend and certain functionality is implemented on the application and not in the largely web based interface. This requires platform-specific technologies. These technologies were chosen as they are the standard programming language used in their respective environments.

- Swift (iOS)
- Java (Android)

5.3 Interface

The system's interface will be largely web based. This is due to the fact that there needs to be real time updates within the interface, constant data retrieval and a persistent internet connection. Additionally all web technologies are designed in such a way that they can easily integrate with one another thus the system can make use of the various strengths of a particular technology yet not being bound to its limitations. Additionally, within these technologies some are used for web design, and others web development. The application's interface is a combination of these two categories:

5.3.1 Web Design

- HTML
- CSS
- Bootstrap
- XML

5.3.2 Web Development

- PHP
- Javascript

5.4 Database

Database and data storage is an integral part of the system. The application heavily makes use of persisted data, be it user information or point of interest information etc. Thus a large database with multiple tables needs to be maintained. Given the system requires fast data retrieval, a technology that offers high performance at scale is used.

- MongoDB

5.5 Networking

The ability to communicate with the various routers around campus is of utmost importance. The applications requires networking capabiilties in order to ping routers and deduce the user's location. In order to do this a technology is employed to scan the surrounding routers and ping them.

- Nmap
- OAuth2 tokens item

5.6 Navigation

NavUP is essentially a navigation application. Specialized technologies are used for this purpose. The application utilizes these technologies in order to obtain locations, calculate routes and distances, find locations etc.

- GPS
- Google Maps API