UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Faculty of Engineering, Built Environment and Information Technology

PYTHON ROUND2

Architectural Design

Dewald de Jager	15188800
Johan du Plooy	12070794
Juan du Preez	15189016
Marthinus Hermann	15081479
Maria Qumayo	29461775
Orisha Orrie	13025199
Azhar Patel	15052592

STAKEHOLDERS

Computer Science Department
of University of Pretoria

Vreda Pieterse

Contents

1 Introduction

1.1 Purpose

1.2 Scope

2 External Interface Requirements

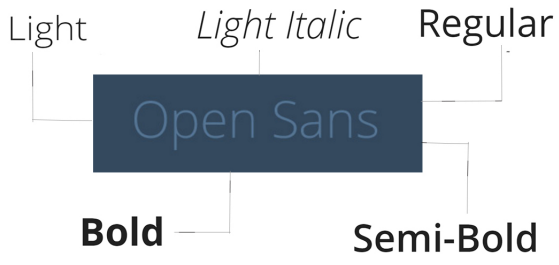
2.1 User Interface

NavUp's user interface is clear, simple to understand, concise, familiar, responsive, attractive and easy to use. It is designed to the standard guides specified by googles materiel design on mobile devices. This ensures that the user has a stress free experience when interacting with it. The application is effective, efficient resulting in a satisfied user.

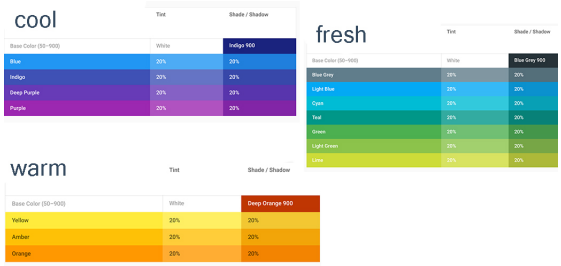
References to GUI standards and product family style guides that are to be followed by NavUP: As previously mentioned google material design standards will be adhered to while designing NavUPs graphical user interface. Mainly because we expect the NavUP application to be used on mobile devices. Google has a well-defined set of rules that stand to the classic principles of good design and graphics specifically for mobile interfaces. The goal is to create a visual language that synthesizes good design with innovation and technology.

Standards for fonts, icons, button and labels, images and colour schemes Typography is a make or break for any mobile application because of the limited screen sizes and resolution. Although fonts may be customizable by the user, depending on their preference, the default font used in the NavUP application is Open Sans.

Open Sans is a sans-serif typeface that is designed with open forms and a natural appearance so its easy to read. It is optimized for print, web as well as mobile interfaces. It has great legibility characteristics in its letterforms and adapts well with various styles.



Colour Schemes Colour schemes in material designed is mostly motivated by deep bold shades compared with soft environments, deep shadows and bright highlights. NavUP will incorporate a combination of fresh, warm and cool colours. Each colour will react differently when an action is triggered within the application. Tints and shades will adjust accordingly to let the user know that an action has been taken. Below are the standards for appropriate values for each to ensure consistency and harmony. The standard is 20



Icons, buttons and images Material icons and images use geometric shapes to visually prepresent core ideas and capabilities. NavUp incorporates both application/product icons as well as system icons.

Application icons visually express the application or brands product and services. For example, the NavUP logo or icon. It would be simple, bold and friendly to communicate the core ide and intent of the product/application. To help user navigate their way around campus, in this case. The application icon is the brand identity.

A system icon, or UI icon, symbolizes a command, file, device, or directory. System icons are used to represent common actions. Below are sample system icons that are find in the NavUp application. Icons are required to be simple, consistent, intuitive and related to an action.

For Square Icons: Height: 152 dp Width: 152 dp For circular icons: Diameter: 176 dp

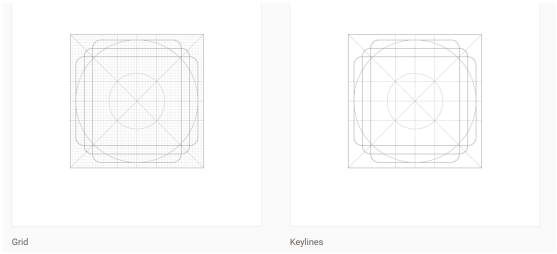
- **Grid dimension requirierens :**
  All grid icons have been developed to facilitate consistency and institute a clear set of rules for the positing of graphics elements. Icon dimension requirements are as follows:

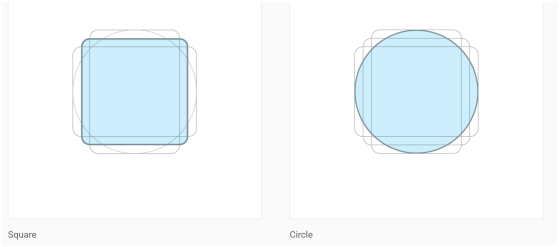
  For Square Icons:

- **Height:** 152 dp
- **Width:** 152 dp

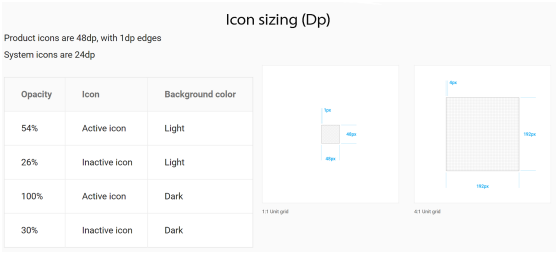
  For circular icons:

- **Diameter:** 176 dp

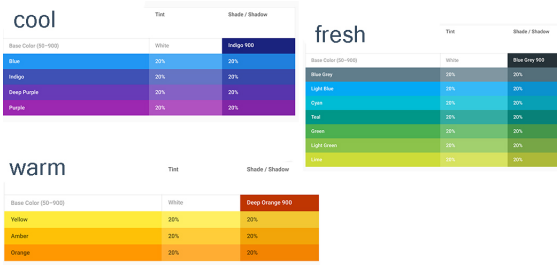




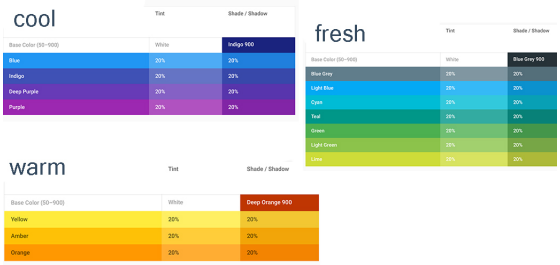
Icons are measured in Density Pixel units and are required to adhere to the sizes depicted in the table below.



Colour Schemes Colour schemes in material designed is mostly motivated by deep bold shades compared with soft environments, deep shadows and bright highlights. NavUP will incorporate a combination of fresh, warm and cool colours. Each colour will react differently when an action is triggered within the application. Tints and shades will adjust accordingly to let the user know that an action has been taken. Below are the standards for appropriate values for each to ensure consistency and harmony. The standard is 20



Colour Schemes Colour schemes in material designed is mostly motivated by deep bold shades compared with soft environments, deep shadows and bright highlights. NavUP will incorporate a combination of fresh, warm and cool colours. Each colour will react differently when an action is triggered within the application. Tints and shades will adjust accordingly to let the user know that an action has been taken. Below are the standards for appropriate values for each to ensure consistency and harmony. The standard is 20



2.2 Hardware Interface

2.3 Software Interface

2.4 Communication

3 Performance Requirements

The system should be able to run in the following way:

- The system will be run through campus servers as well as web servers.
- The performance of the application will be dependent on the users hardware device.
- The loading time of each page will be dependent on the strength of the users internet connection as well as the number of users currently using campus Wi-Fi.
- The time it takes to log in will be dependent on the internet connection.
- All new users should be added to the system in less than a second however, it may take the user longer to see this, depending on the internet connection.
- Users should be sent notifications as soon as they are available on the system.
- Admin should be able to access the GIS module as well as user module with ease in less than a second as these details will be located on campus servers.
- Points of interest should appear as soon as a user searches for a location or searches for a specific point of interest however, this will once again be dependent on the internet connection.
- Users should be able to search, delete, save as well as modify their route. This should be able to take place quickly. Saving and modifying will be dependent on the user and searching and deleting will be dependent on the internet connection.
- Rendering display service will differ from device to device while requesting service relies on the internet connection.
- Push services for events should happen as soon as the event has been added to the application
- Fitness service and heat maps should be updated in real time.

4 Design Constraints

This section describes restrictions on design alternatives regarding standards and limitations of hardware capabilities

1. **Storage space**

    - **Description:** The amount of storage space required by the application must be within the maximum storage limits of a budget phone to accomodate a range of phones typically used by students.
    - **Maximum:** 90MB
    - **Reasonable:** 40MB
    - **Optimal:** 10MB.

2. **Memory usage**

    - **Description:** The amount of RAM used by the application should be a reasonable amount considering that many smartphones used by students only have a capacity of 1GB RAM.
    - **Maximum:** 100MB
    - **Reasonable:** 90MB
    - **Optimal:** 40MB.

3. **Data Retrieval**

    - **Description:** The return rate of data from relevant databases such as the GIS database should be close to perfect in order to prevent delays and inaccurate navigation.
    - **Minimal:** 98
    - **Reasonable:** 99
    - **Optimal:** 100

4. **Networks and Location**

    - **Description:** The system will need to make use of Wi-Fi, mobile networks and GPS in order to enhance accuracy of location and results as far as possible. In addition, the system will need to use relevant networks in the their respective areas (i.e., using GPS within buildings is pointless).
    - **Minimal:** 98
    - **Reasonable:** 99
    - **Optimal:** 100

5 Software System Attributes

5.1 Reliability

When transferring data from point A to point B, it is crucial that the integrity of every piece of information be maintained and accounted for. One piece of misinformation

could very well mean the difference between the locations of two different class rooms or even completely different buildings. One of the applications features is to ensure the fastest route, which relies heavily on making the right decisions based on the data it receives, therefore if the information is not accurate or even correct, the user could end up late to their desired destination.

5.2 Robustness

All functionality of NavUP must work correctly even under exceptional conditions; The most important being correctness when there is no Wi-Fi connection, extremely slow Internet connectivity or low availability of resources on either the host system for each access channel or the server.

The data module should provide means of communication even when the server is under strain in order to be robust. This may be done in terms of a pipeline system where messages are queued and receipt of them are acknowledged, along with the error detection and correction discussed under Reliability.

5.3 Efficiency

Sending data between remote locations relies heavily on the network traffic and its capabilities/limitations. Thus the idea would be to put the least amount of stress and congestion on that network and only make strategic requests from client to server. The idea would be to make relatively frequent requests to small yet crucial information, and much less frequent requests to information that is less likely to change within a small period of time.

5.4 Interoperability

5.5 Maintainability

5.6 Testability

5.7 Portability

5.7.1 Hardware Independence

Seeing as how the system will be deployed with a web base as a back-end API, the hardware choices are not limited to a specific set of components, so long as they meet the minimum or preferably recommended requirements to be able to handle the estimated network traffic to keep the system performing at optimal levels.

5.7.2 Software Dependence

The software on the other hand will be limited to being web specific, i.e. being able to use or integrate with applications like AJAX, Node etc.

5.8 Reusability

5.9 Modularity

5.10 Cohesion

The NavUP system can be designed with the high cohesion principle in mind. The system as a whole should be formulated in to a hierarchical tree-structure, comprising of subsystems, or modules. The modules should be able to either function independently, or be able to call on one or more modules in the hierarchy to perform a given task. Thus adding or removing modules should not break the system so much as increase or reduce its functionality as a whole.

The Data module comprises mainly out of a client and a server subsystem. The server for example would be able to make use of a Database Management System, for handling records relevant to the NavUP system, such as location information, events etc. Furthermore, both the Client and Server would need to have some form of communication method, which could also be in the form of a subsystem or module.

One of the more beneficial advantages of modularity is the fact that a module/subsystem can be focused on to either make improvements or even completely swap it out for another more efficient subsystem to improve overall performance.

5.11 Coupling

The subsystems should have minimal impact on and dependency on the other subsystems of the system. The system as a whole applies the low coupling principle. Subsystems should be abstracted and implementation changes should only affect the subsystem that was modified. This abstraction layer will provide an interface between all the subsystems that prevents interference and augments communication between them.

The data module will be represented as a module with requests and responses that are used both by access channels and the server. This allows for other subsystems to easily use it without having to worry about implementation specific details and the implementation can be changed to use a different technology or update the application programming interface (API).

6 Technologies

6.1 Data

Message passing is done in a transactional manner in this subsystem. The access channels, or clients, need a way to send requests to the server and receive responses. Likewise, the server needs a way to service requests and issue responses. The University of Pretoria Wi-Fi infrastructure can be advantageous in this scenario as we can assume the user is likely connected to the Wi-Fi and fall back on cellular data otherwise. This is done by having the server accessible both locally and publicly via the Internet. If a connection

cannot be established to the local server a connection via cellular data or a different Wi-Fi connection is made to the server.

The connection is established via either the HyperText Transfer Protocol (HTTP), or it's secured counterpart HTTPS that makes use of the Secure Socket Layer (SSL) depending on the nature of the message contents, and is maintained only for the duration of the message send and receive process. The requests will, more specifically, be made using Asynchronous JSON and XML (AJAX) and will always be paired with a response. The protocol that HTTP is built on, the Transmission Control Protocol (TCP), ensures the delivery of messages in the correct order and that the messages remain intact.

6.2 Users

6.3 Events

6.4 Points of Interest

7 UML Diagrams

7.1 Data

7.2 Users

7.3 Events

7.4 Points of Interest

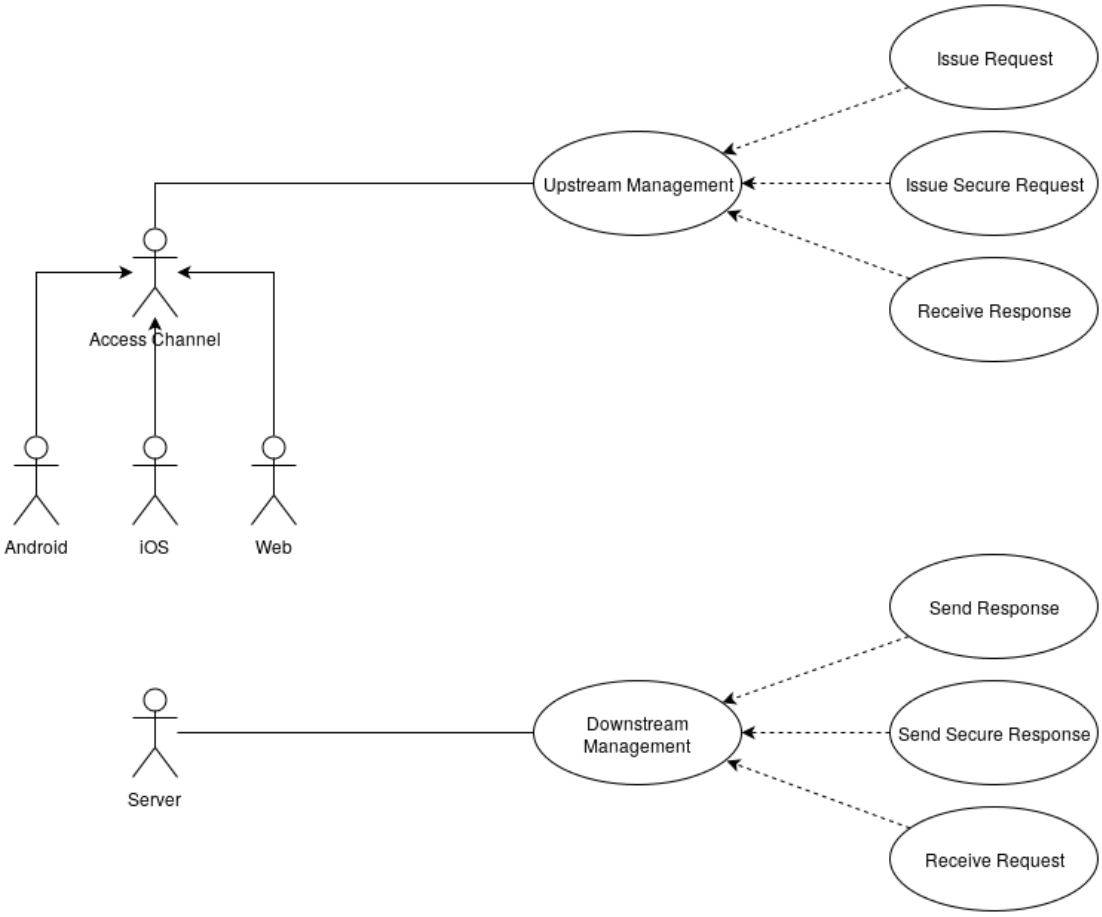


Figure 1: Data Use Case Diagram

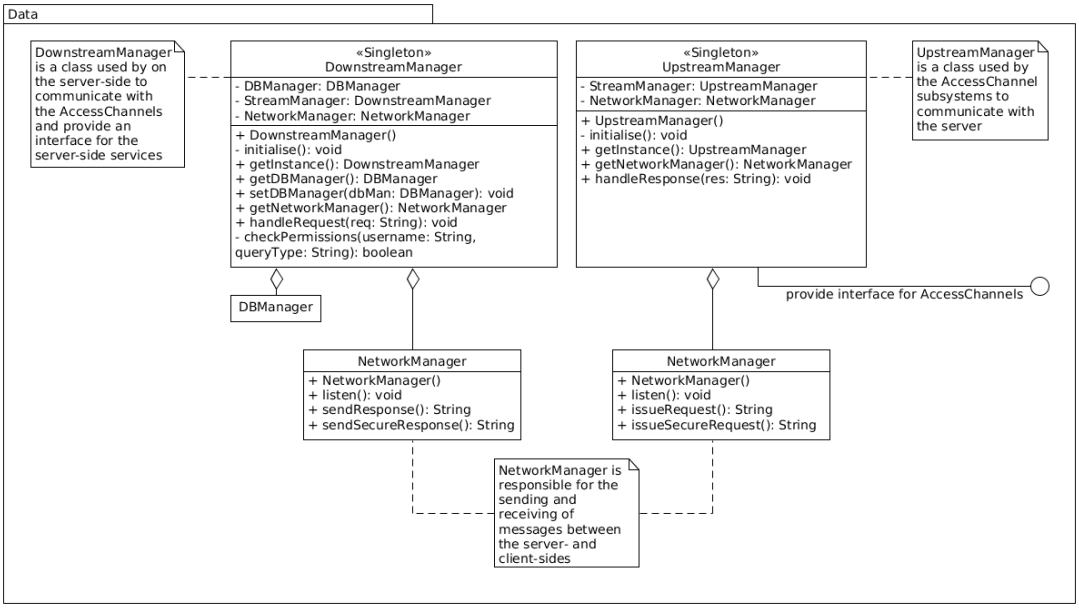
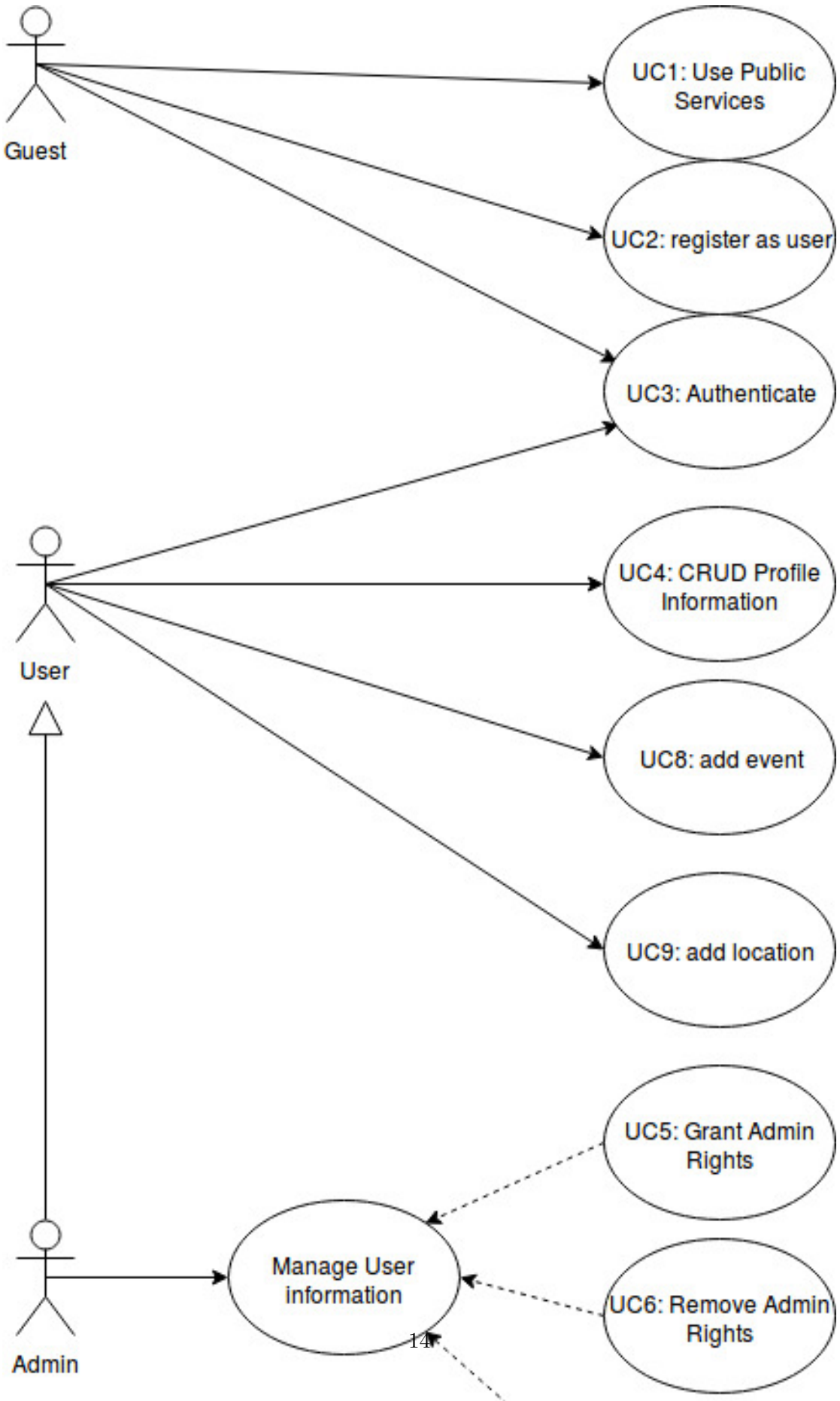


Figure 2: Data Class Diagram


Guest
UC1: Use Public Services
UC2: register as user
UC3: Authenticate
User
UC4: CRUD Profile Information
UC8: add event
UC9: add location
Admin
Manage User information
UC5: Grant Admin Rights
UC6: Remove Admin Rights

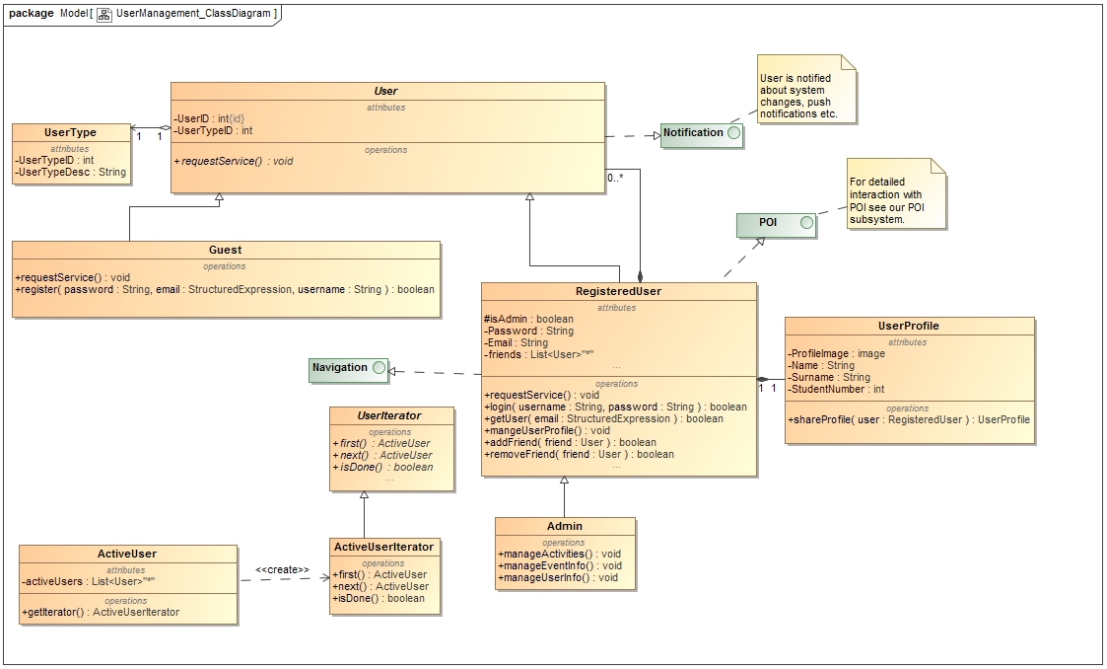


Figure 4: User Management Class Diagram

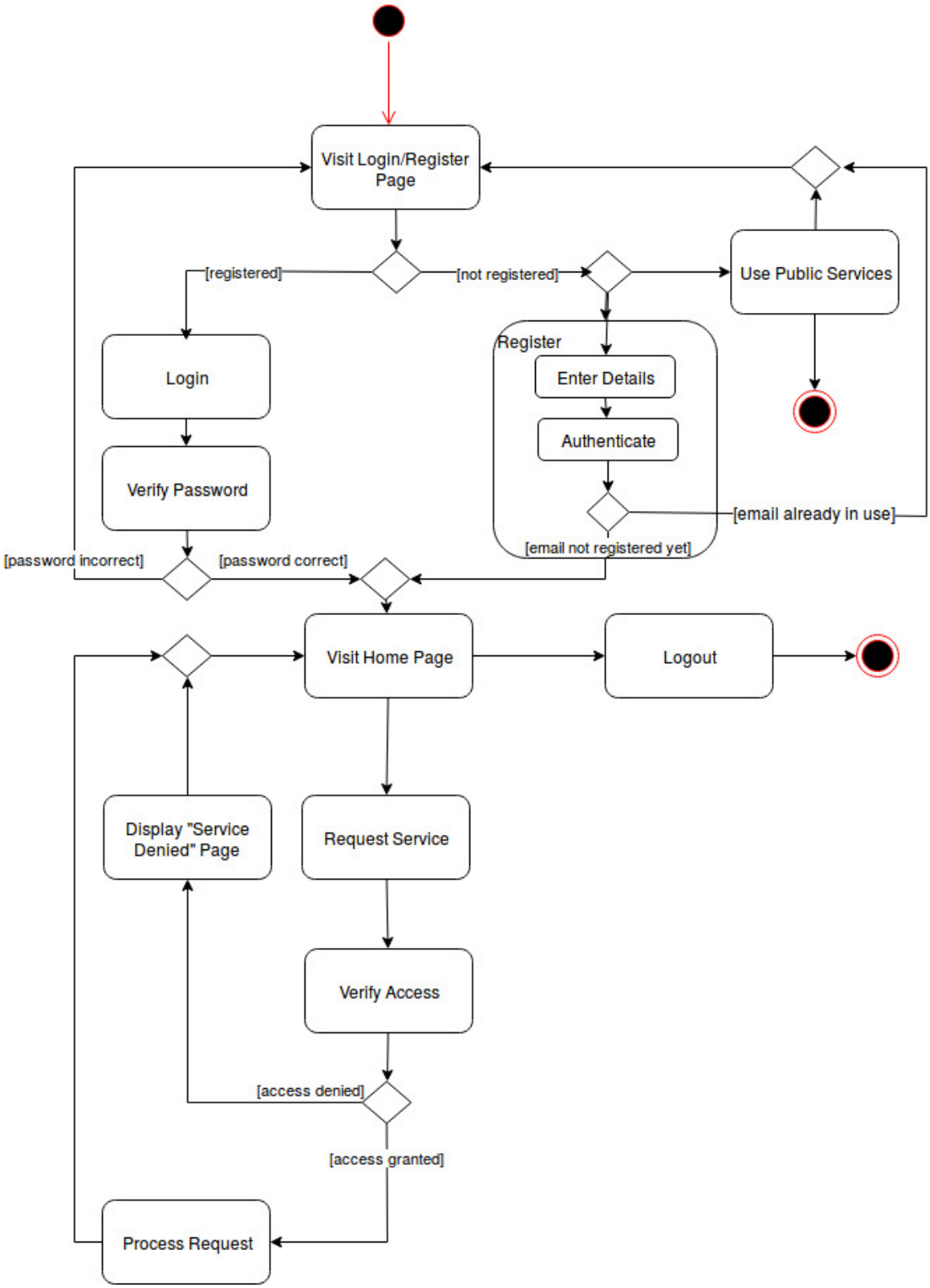


Figure 5: User Management Activity Diagram

Figure 6: Events Use Case Diagram

Figure 7: Events Class Diagram

Figure 8: Points of Interest Use Case Diagram

Figure 9: Points of Interest Class Diagram