

Phase 2 - System Requirements Documentation

By Team Basic:
Marin Peroski
Muhammed Carrim
Mpho Baloyi
Keoagile Dinake
Keanan Jones
Daniel Malangu
Nkosingathi Mothoa

March 12, 2017

Contents

1	External Interface Requirements	3
1.1	User Interfaces	3
1.2	Hardware Interfaces	3
1.3	Software Interfaces	4
1.4	Communication Interfaces	4
2	Performance Requirements	5
3	Design Constraints	6
4	Software System Attributes	7
4.1	Design Qualities	7
4.2	Run-time Qualities	7
4.3	System Qualities	8
4.4	User Qualities	9
5	Technologies Used	9
5.1	Web	9
5.2	Connectivity	9
5.3	Operating Systems	10
5.4	Cellular Hardware	10
6	UML Diagrams	11
6.1	Navigation Module	11
6.2	Points of Interest Module	18
6.3	Users Module	25
6.4	Notifications Module	29

1 External Interface Requirements

External interface requirements are categorised under four main points. Each underlines an important interface aspect that will be fundamental to the successfully operation of the NavUP application and system.

1.1 User Interfaces

User Interfaces The presentation of an application is as important as its underlying algorithms and components that make it tick. The appearance can chase away a user or draw them into using the application more and more. A simple interface that has large buttons for each option would be the most optimal solution. Having a menu appear first providing multiple options to the user such as enter directions, save a route, choose from common routes etc. The fonts will need to be clear and the text concise. Once the user has chosen their option they can be taken to the appropriate page, as in the case of entering a direction a text box can appear with an enter and cancel button. After directions have been entered the map should take up most of the display area. The only buttons that should be available is to cancel the navigation or modify it. This will allow the navigation to take up the largest portion of the display as it is the most important visual interface element. The user interface should scale to different resolution sizes and resolutions above ultra-high definition should not be considered as well as, those below 720p.

1.2 Hardware Interfaces

Beauty is not the only aspect of an application that needs to be accounted for. The guts of which this application runs on is extremely important. It needs to be optimised for ARM and Intel processors however, the main optimisation should be focused on ARM as that is the common mobile central processing unit platform. With regards to storage and main memory the application will also need to be reasonable in size (no more than 100 MB or so) and ensure that the assets that are loaded into main memory (images of the map for example) are not large as well. This is due to the limited processing and main memory resources available. Having an application take up most of those resources will result in poor performance for other programs and a lower battery life for the duration of its use.

1.3 Software Interfaces

As there are three different platforms the software will be implemented on namely, IOS, Android and through web based browser access, it is important to ensure that these different platforms can communicate between each other. This communication will mainly occur through the heat maps as each device connected to the Wireless network will need to be queried for location so that a real time display for heat maps can be provided. With regards to data exchange the software will need to obtain position through hardware interfaces (Wi-Fi in this case) based on which Wireless router the device is connected to. This information along with the heat maps will be used along with user input (users destination) to generate a step by step directional output on the map interface which the user can interact with through the user interface.

1.4 Communication Interfaces

Communication for this application is key to its functional success. The main communication medium that will be used is the devices Wi-Fi connection which will be connected to the wireless routers located in various areas across campus. This will allow the users location to be determined, directions to be generated from the current location to the destination that the user has provided and update the heat maps based on how many other users are connected to the network and are currently using the application. Wi-Fi standards will also need to be considered as there are multiple bands, channels and protocols available. The 2.4 GHz frequency band is the common communication frequency used in wireless technologies. The application should cater for both of the data transfer rates of the 2.4 GHz and 5 GHz bands as performance would be wasted if the application does not keep up with the 5 GHz band.

2 Performance Requirements

Performance requirements for any form of software is an important aspect to consider. As NavUP will be running on a mobile platform where resources are more limited than that of a laptop or desktop the software will need to be optimised to run as efficiently as possible.

One of the most important things to consider with mobile software is how much battery power does it consume whilst it is running in the foreground or background. While running in the foreground the largest consuming factors of processing power and battery life (excluding user settings for the phone such as screen brightness etc) would be the direction generation and heat map information that is pulled from the server. When the user minimises the application the most efficient and effective way for it to run in the background should be to halt updating the users location as well as, pulling any heat map data. This will save energy being needlessly used by the application and allow the phones battery to last longer.

Taking a look at the heat maps the server will need to do a lot of the processing with regards to where users are located in the vicinity. That will then be relayed to the smartphone of users trying to navigate through campus so that the shortest or simplest routes can be calculated. The heat maps will then be updated to the phone and displayed as an overlay on the map. The updates for the heat maps will also need to occur as a real time update. This update frequency will definitely require a relative amount of processing power from the smartphone so that it can constantly show the heat maps as well as update the directions based on them.

The geographical maps will most likely be images that do not exceed a full HD resolution so that the application does not take up too much space on the phone. They will also be stored on the phone to limit the amount of data that constantly needs to be pulled from the server and allow for more important features to take precedence. Another benefit will be that the images will load a lot more quickly. They can be of a .jpg type image compression which will result to around 50 MB of storage for just the images).

3 Design Constraints

Design constraints are major factor to consider when designing anything in the real world. Taking the endless possibility of ones imagination for an idea and scaling it to real world application can be quite challenging but, without doing so the creation may never be feasible in its implementation. These same real world restrictions apply to the NavUP software as well.

Hardware constraints are a major issue for software developers in any field of the industry. With NavUP one would need to consider how certain taxing features will be implemented on older and less powerful hardware. Taxing operations such as real time heat mapping effecting how directions are generated and modified during runtime may need to be scaled to occur at intervals rather than constantly refreshing. This can be determined by the software based on the detected hardware that it is running on (1 GHz single core processor and 512 MB of RAM or less as an example). The geographical images could also be displayed at a lower resolution for phones that do not have a full high definition display. Battery life is another major factor to consider when designing an application for mobile devices. While running in the foreground all operations should continue as expected however, once the user places the application in the background or puts their phone into a sleep state it needs to halt all navigation and real time heat map updates. This will prolong the battery life of the phone as less energy will be expended on wasted processes that the user will not even utilise.

Software constraints are another factor to consider. As this application is being developed for both the IOS and Android operating systems as well as, being able to be used thorough a browser there are multiple access channels that need to be considered. The application will need to be programmed differently for each operating system along with its web based form as well. This will include elements such as the graphical user interface. Communication between each platform will need to occur through the application for features such as heat mapping to be accurate.

With regards to constraints that do not include software or hardware but, involve a more social and ethical dilemma is that of privacy. The will need to provide consent to their location being used by the application. The information provided by the user will need to be kept in a secure format so that no information he or she has provided is compromised. The prompts for information sent to the user will also need to be limited to the requirements of the application so that it can run optimally.

4 Software System Attributes

The attributes of a system collectively describe major components and aspect of it. There are four main categories each of which contain a wide variety of sub-categories which describe different aspects of the system ranging all the way from user interaction to the hardware the system will run on.

4.1 Design Qualities

- Conceptual integrity: as the different modules of the application will be designed by different individuals, maintaining conceptual integrity will require standardisation of documentation, user interface design and programming methodologies.
- Maintainability: changing and adding to the application should be done in a manner in which an addition or removal of an element from one module should only require that element to be incorporated or erased from the other modules without effect the rest of the functionality.
- Reusability: as the application caters for a specific need, the reusability for the NavUP system may be limited to applications that only require some form of positioning system using a wireless network interface.

4.2 Run-time Qualities

- Availability: the availability of the systems position features will be dependent on the users location on campus and whether or not there are any wireless connection points within the vicinity. If there are no connection points a downtime of directional services can be expected. This offline functionality will allow the user to still see the map but, not have an update on heat maps or location.
- Interoperability: the system will involve information exchange between devices in order to determine the amount of devices that are present within the vicinity. This information will then be relayed to the device in order to determine the best directional navigation based on the users location, their destination and the density of individuals (that are using the application) within the vicinity.

- **Manageability:** system administrators will be able to manage the application through various means. They can receive feedback from the numerous amount of students that use the system as well as test the application themselves if fine tuning is necessary. Accuracy of directions can also be tested for in a simulated environment where heat maps values are entered into the program along with destination and location to determine whether the program does indeed produce the shortest or simplest route.
- **Performance:** As a wireless connection will be used to determine various pieces of information the latency between the device and router should vary depending on signal on the signal strength and congestion of that connection point. The heat maps occur in real time thus, minimising the latency between updates.
- **Reliability:** the reliability of the NavUP system will solely depend on whether or not the user can connect to wireless access points. If there are no routers within the users vicinity the user will not be able to obtain full functionality from the application. A lot of the campus has Wi-Fi hot spot access thus, the system should have a high reliability rate.
- **Scalability:** the congestion of users accessing the wireless network on a single route will determine the systems scalability. In general the wireless network of the university can support many concurrent users thus, allowing the system to be operational from small to large scale situations.
- **Security:** as the user is only prompted minimal information such as, destination and an optional timetable entry the risk of any private information being compromised is minimised. The information will be stored in manner that if the application is indeed compromised there will not be a clear distinction as to whose timetable schedule belongs whom.

4.3 System Qualities

- **Supportability:** If there is no connection to the wireless network the user needs to be notified of the limited functionality that they will receive. The user also needs to be notified if they enter incorrect destination names.

- Testability: the system can be tested with ease as all users creating the application have access to the campus environment as well as, the wireless network connection. Creators can traverse the campus to ensure their system is operating as expected. Other students could also be asked to test the applications interface and other various features

4.4 User Qualities

- Usability: the NavUP system needs to have an easy to use interface which also caters for those who are visually impaired. The interface needs to prioritise easy to interpret directions and user friendly geographical maps which clearly indicate important locations.

5 Technologies Used

For an idea to take form one needs to acquire and utilise the appropriate tools. There are various forms of technologies that will be incorporated in the creation of the NavUP system. They will range software components that support the web based system to the different programming languages that will be used to develop the application in its native operating systems.

5.1 Web

1. AJAX can be used for seamless information retrieval from the server if the device makes a requests for any information from the server.
2. Node can be used server side in replacement of Apache from the AMP stack for a more dynamic server side request environment.
3. Java script will be as it is can easily be integrated into any Node features.
4. Mongo DB can be used for dynamic database storage methods.

5.2 Connectivity

1. The system will rely heavily on a wireless connection which will be used to connect to the server locally.

2. If no wireless connection is available, the mobile connection of the phone can be used to take its place and connect to the server instead.
3. The server needs to cater for both 2.4 GHz and 5 GHz wireless connection bands.
4. Catering for all the various wireless network protocols to ensure that the maximum number device can connect.
5. The server needs to send information relative to the transfer rate speeds available based on the client connection type thus, the server will not become a bottleneck.

5.3 Operating Systems

1. Android development for the application can be done so in a Java environment utilising all of the interface elements java can provide for the operating system.
2. IOS development can be done through swift to stick as close as possible to the native operating system environment.
3. Libraries can be included from the various programming languages used to relieve some of the tedious coding (such as the built in data structures in Java).

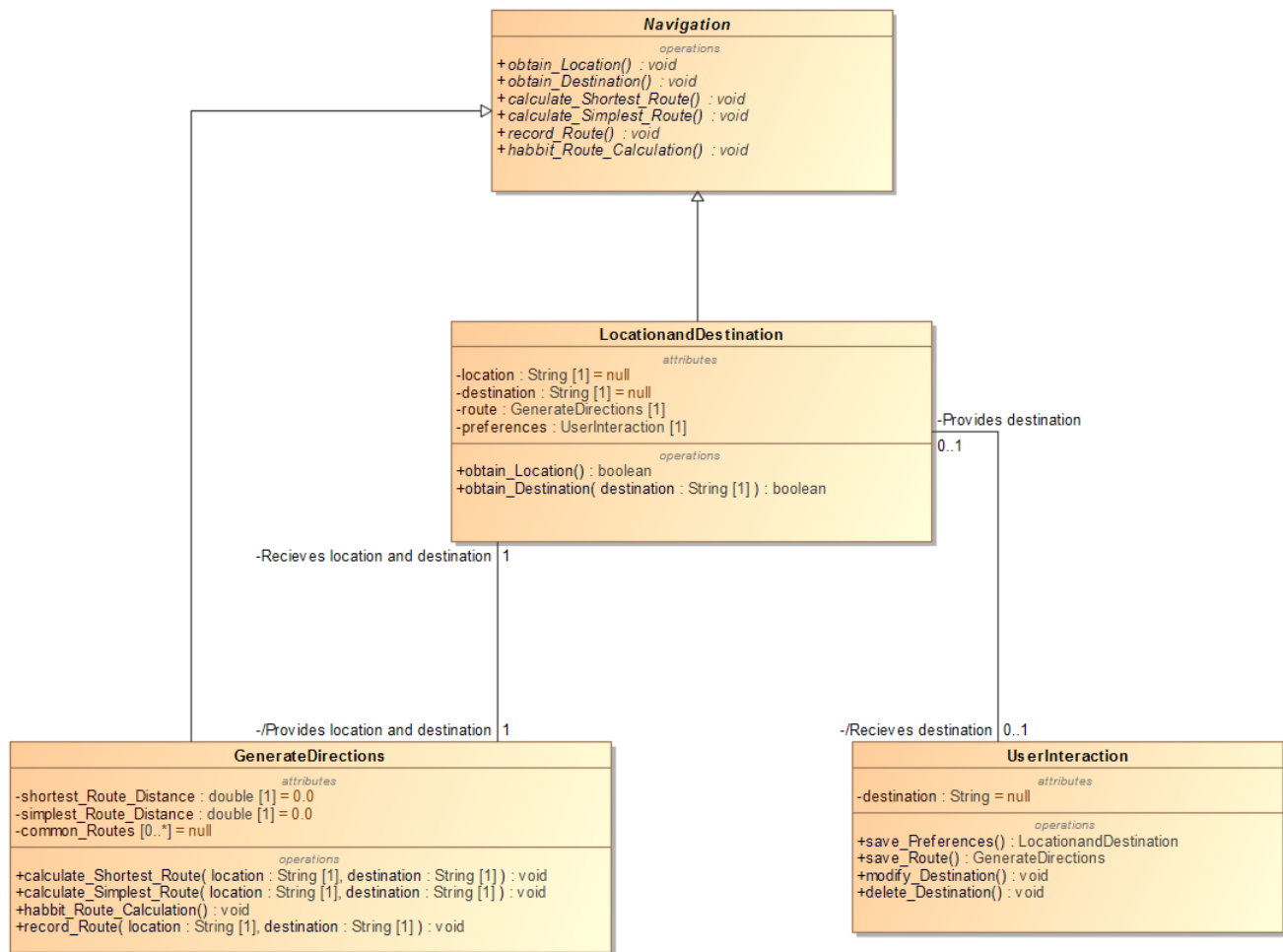
5.4 Cellular Hardware

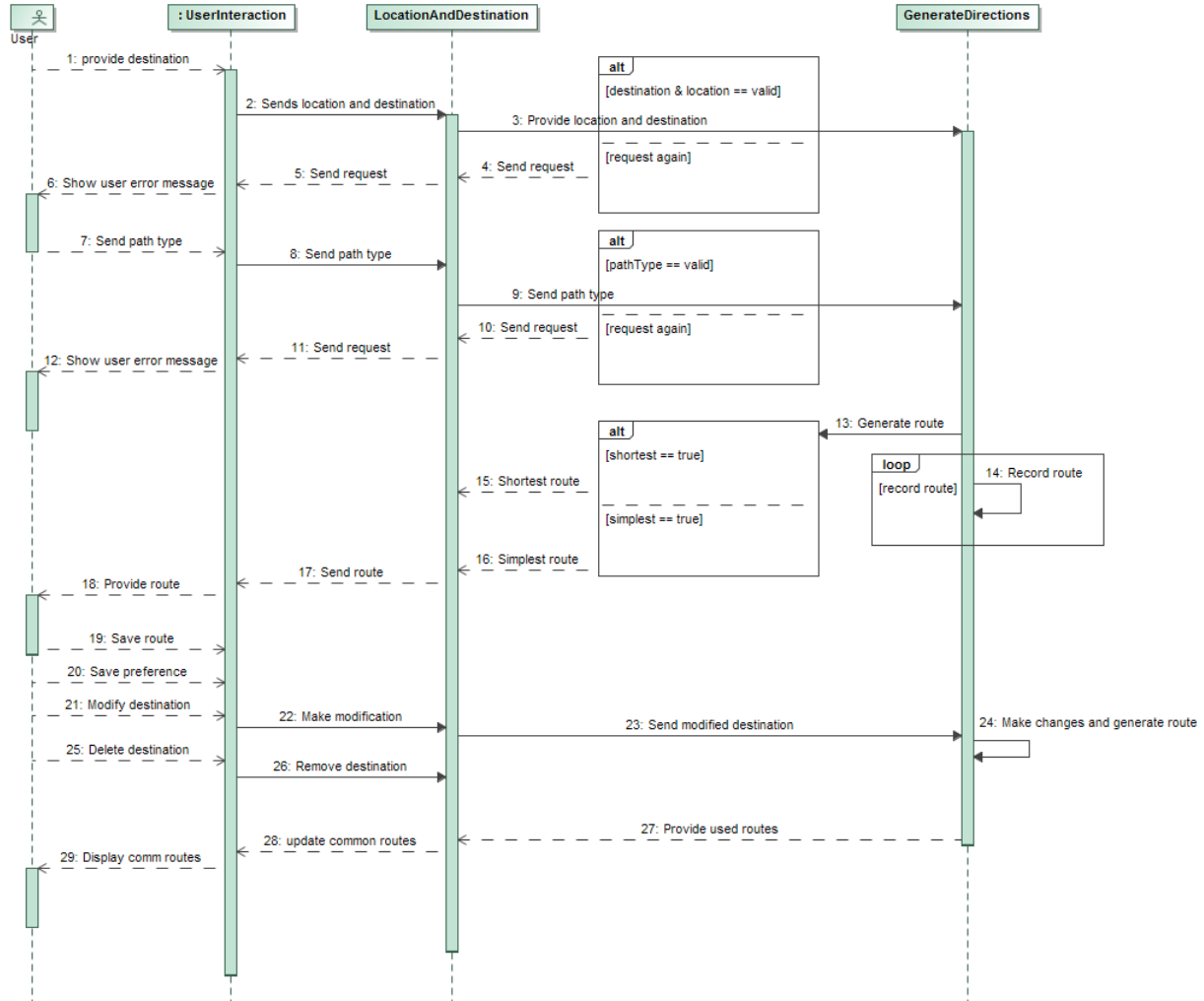
1. Hardware optimisation will be focused on ARM processors however, Intel processors will also need to be considered as some tablets do make use of that central processing unit platform.

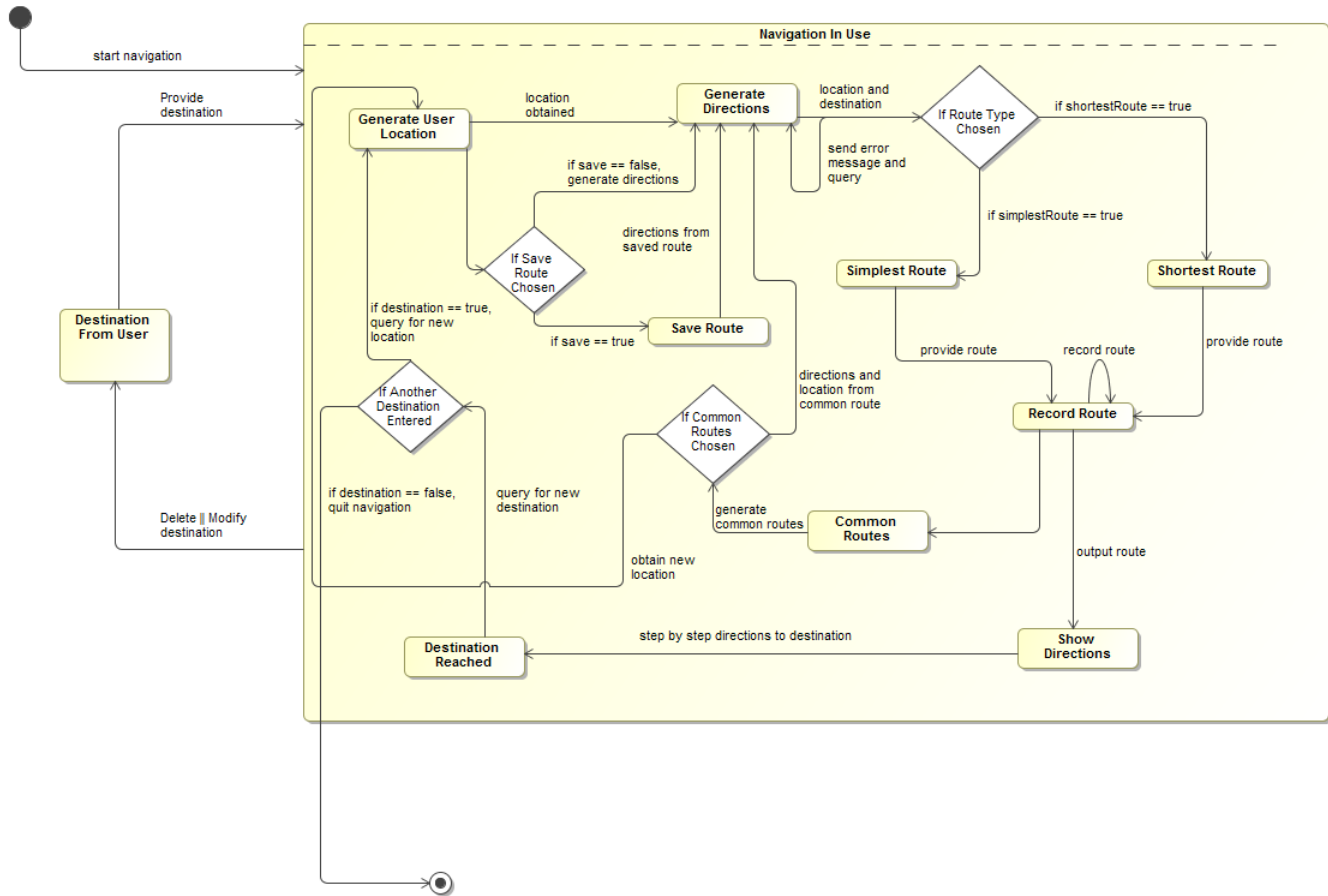
6 UML Diagrams

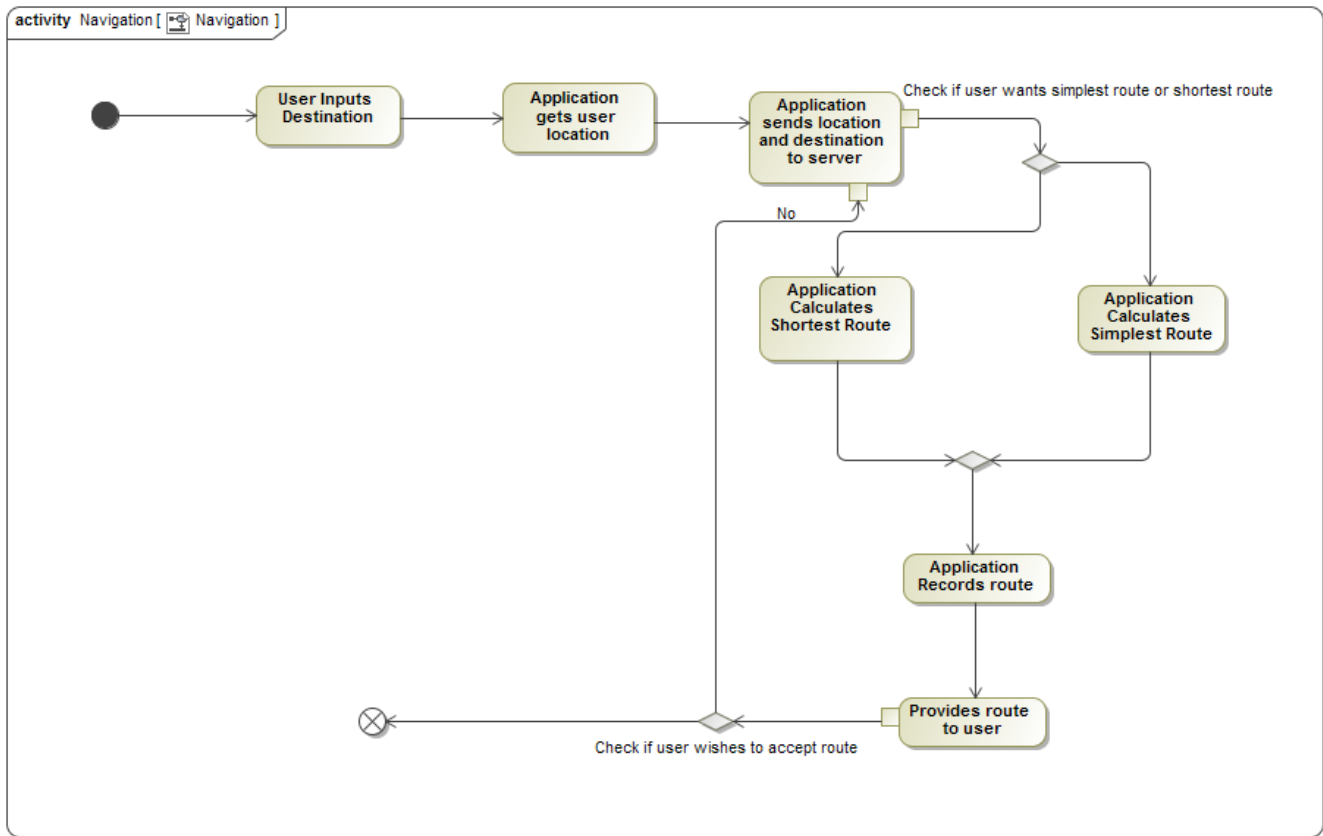
The various Unified Modelling Diagrams for the four chosen modules can be found below. The modules that the diagrams will be describing in detail are as follows: Navigation, Points of Interest, Users and Notifications. The various diagrams represent how these modules will interact with themselves and the system as well.

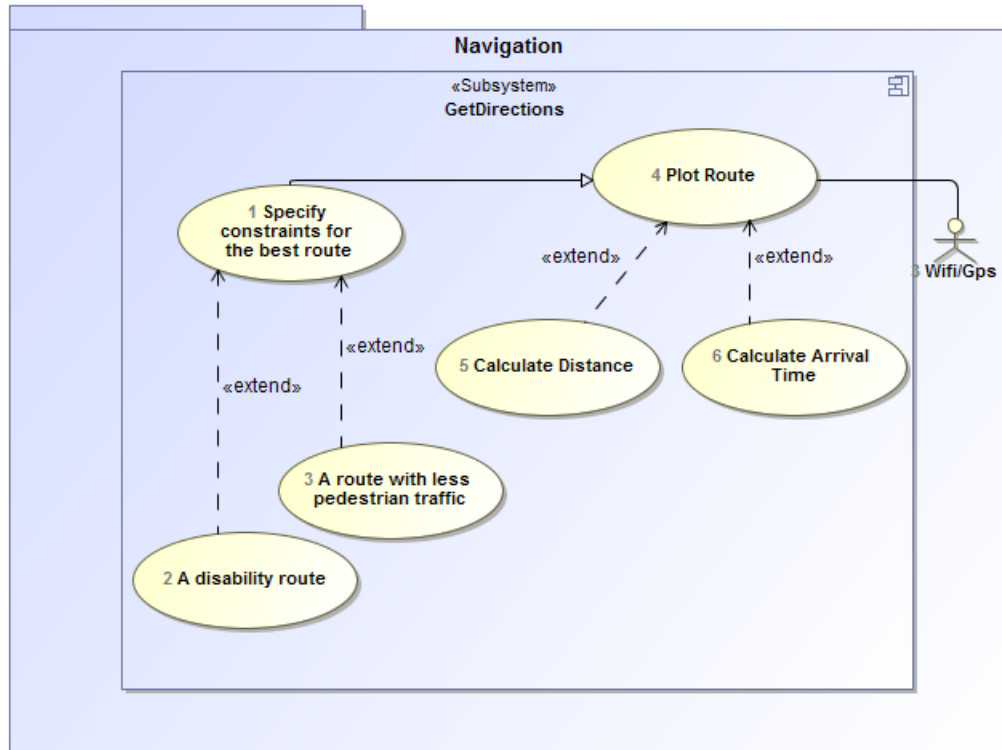
6.1 Navigation Module

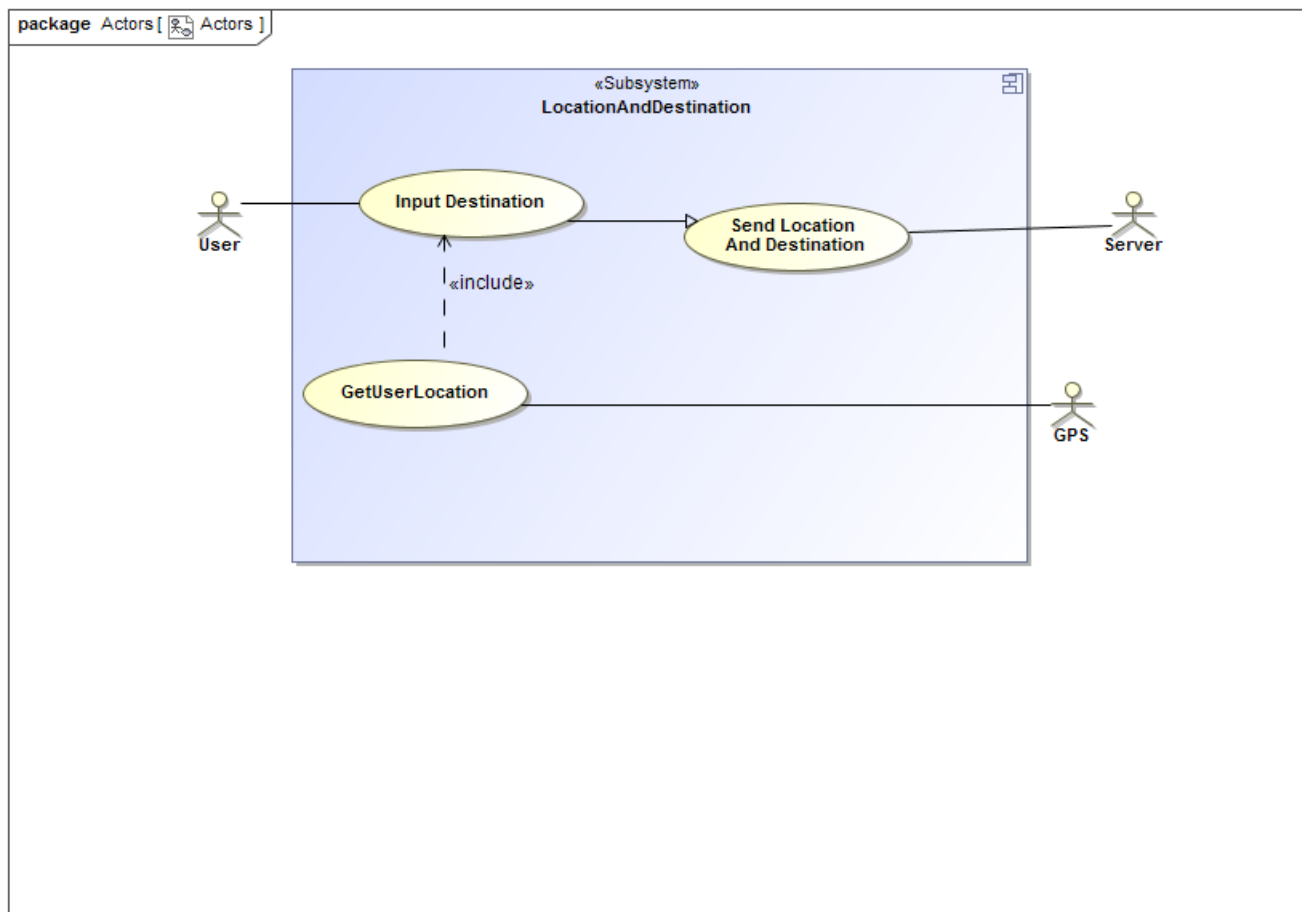


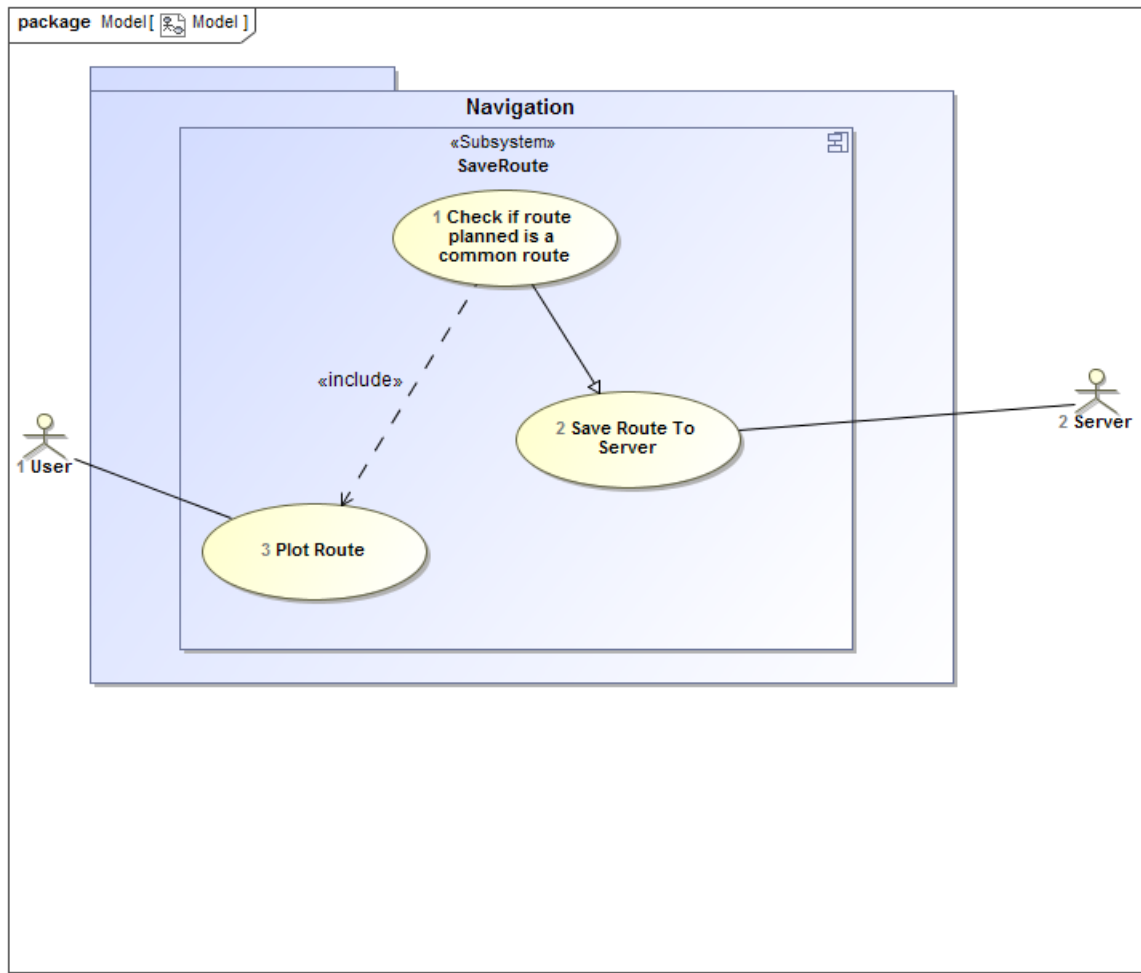






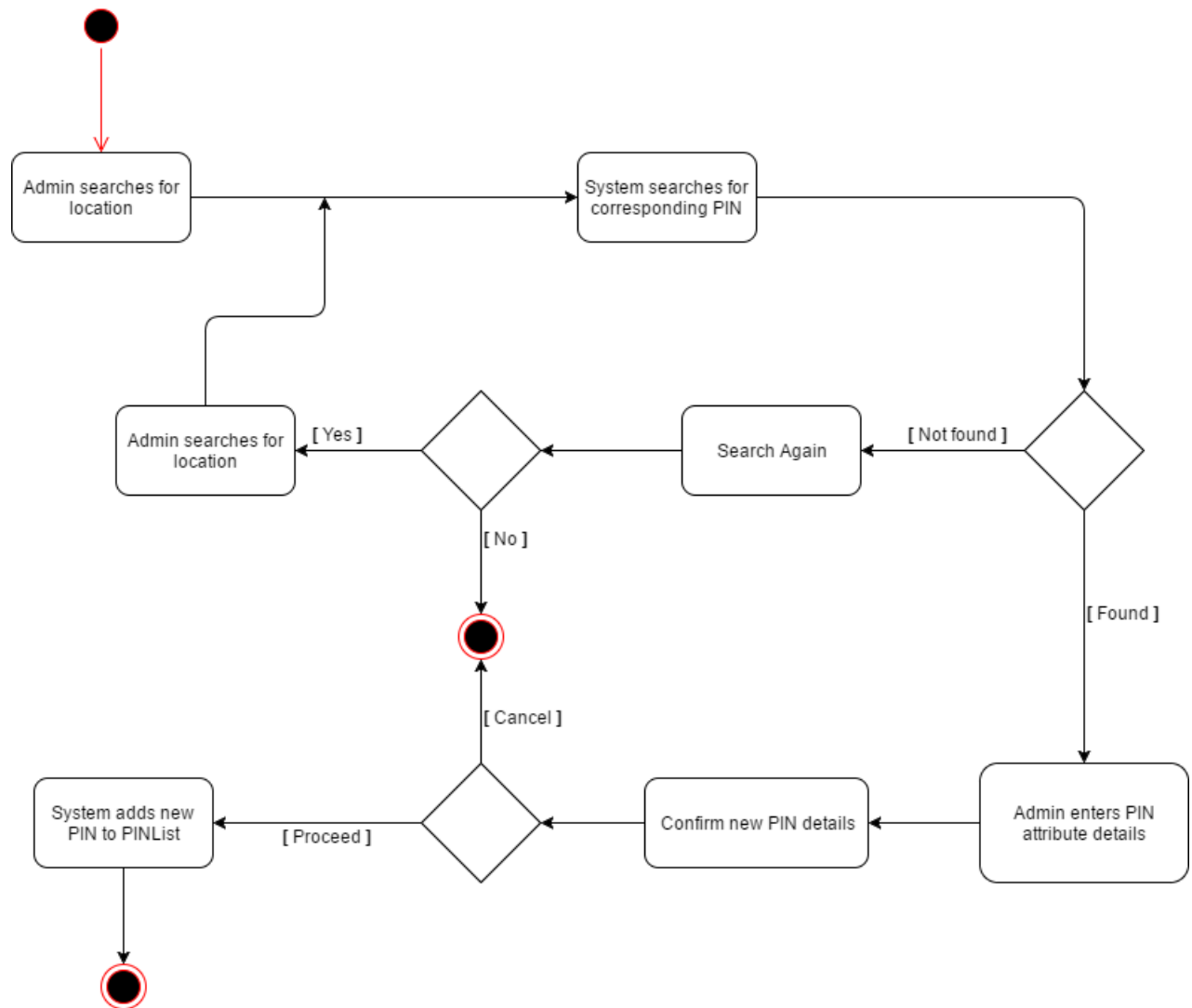




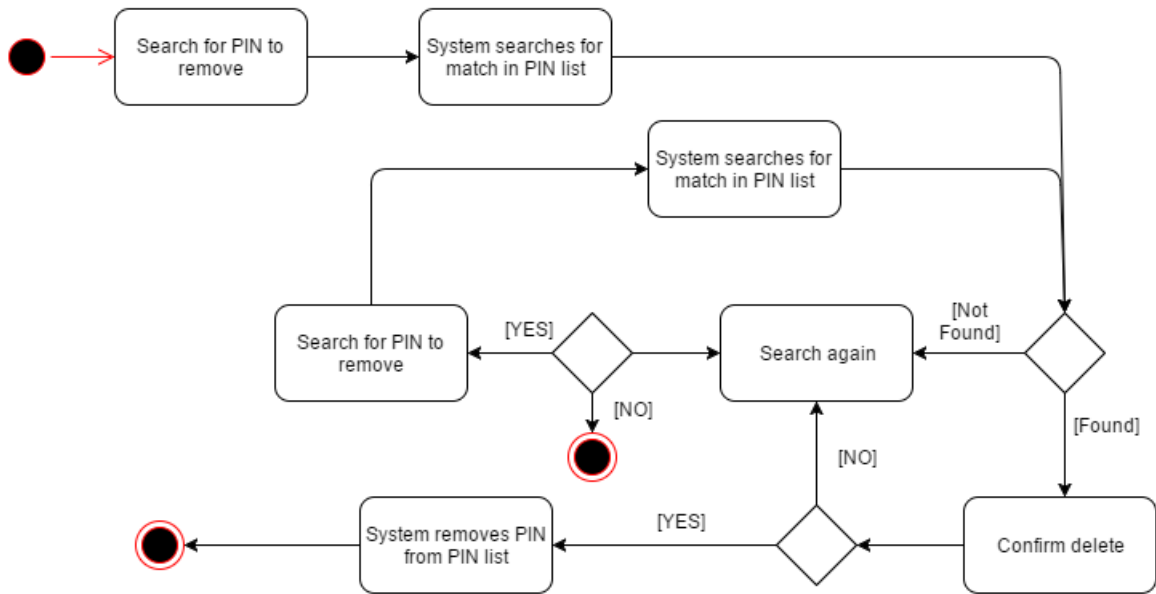


6.2 Points of Interest Module

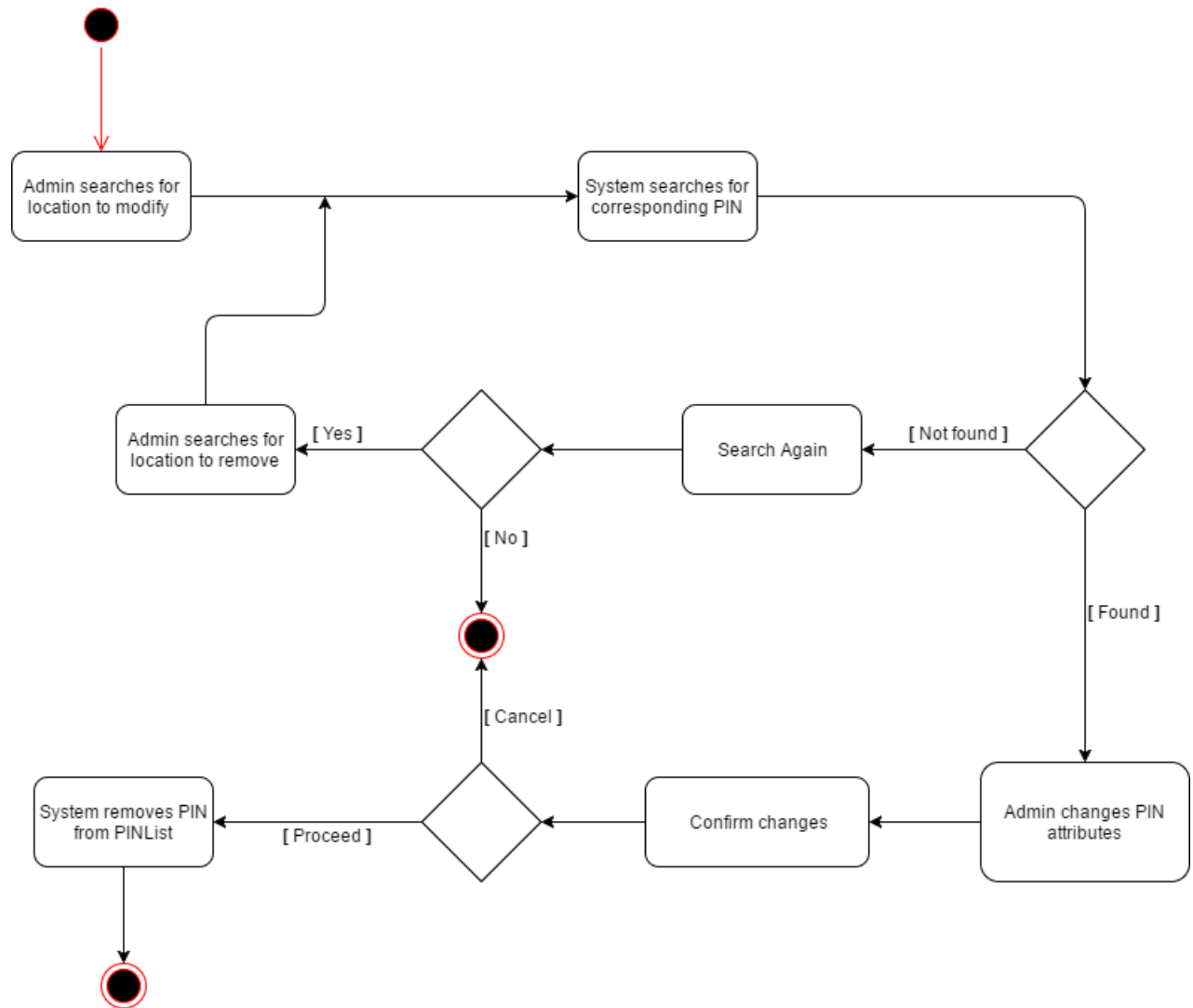
Admin Adds Location Activity Diagram

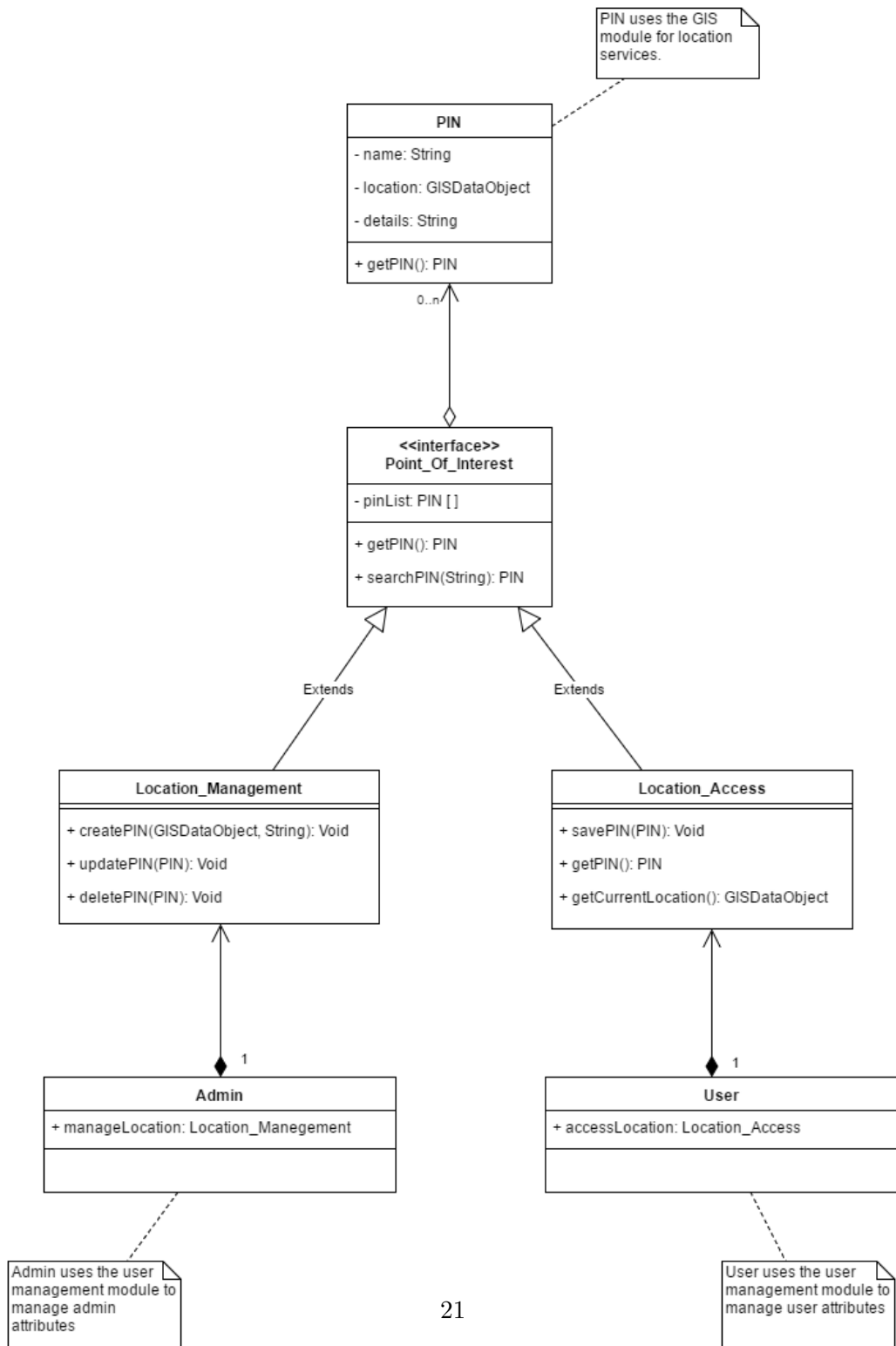


Admin Delete PIN

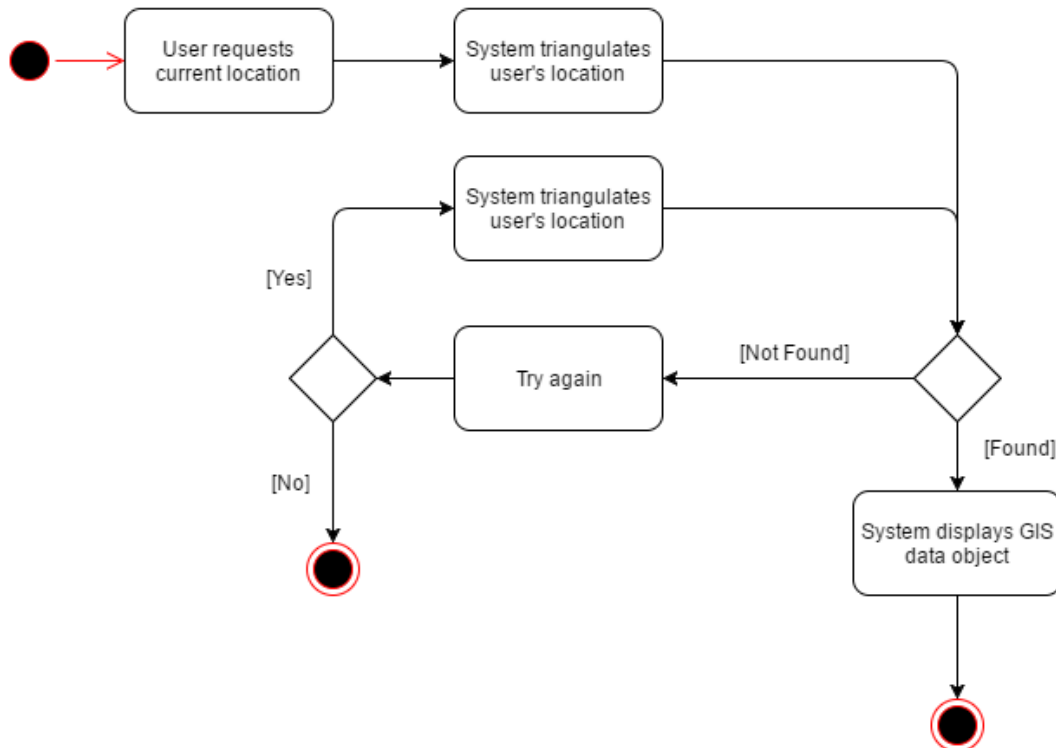


Admin Modifies Location Activity Diagram

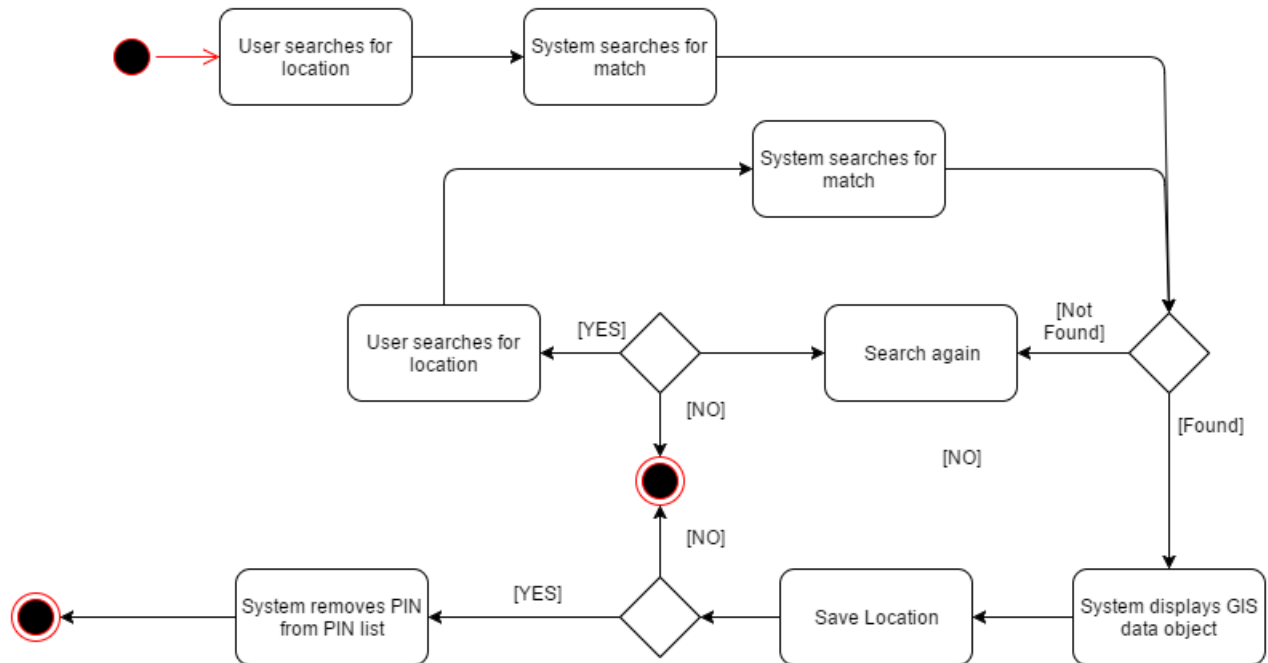




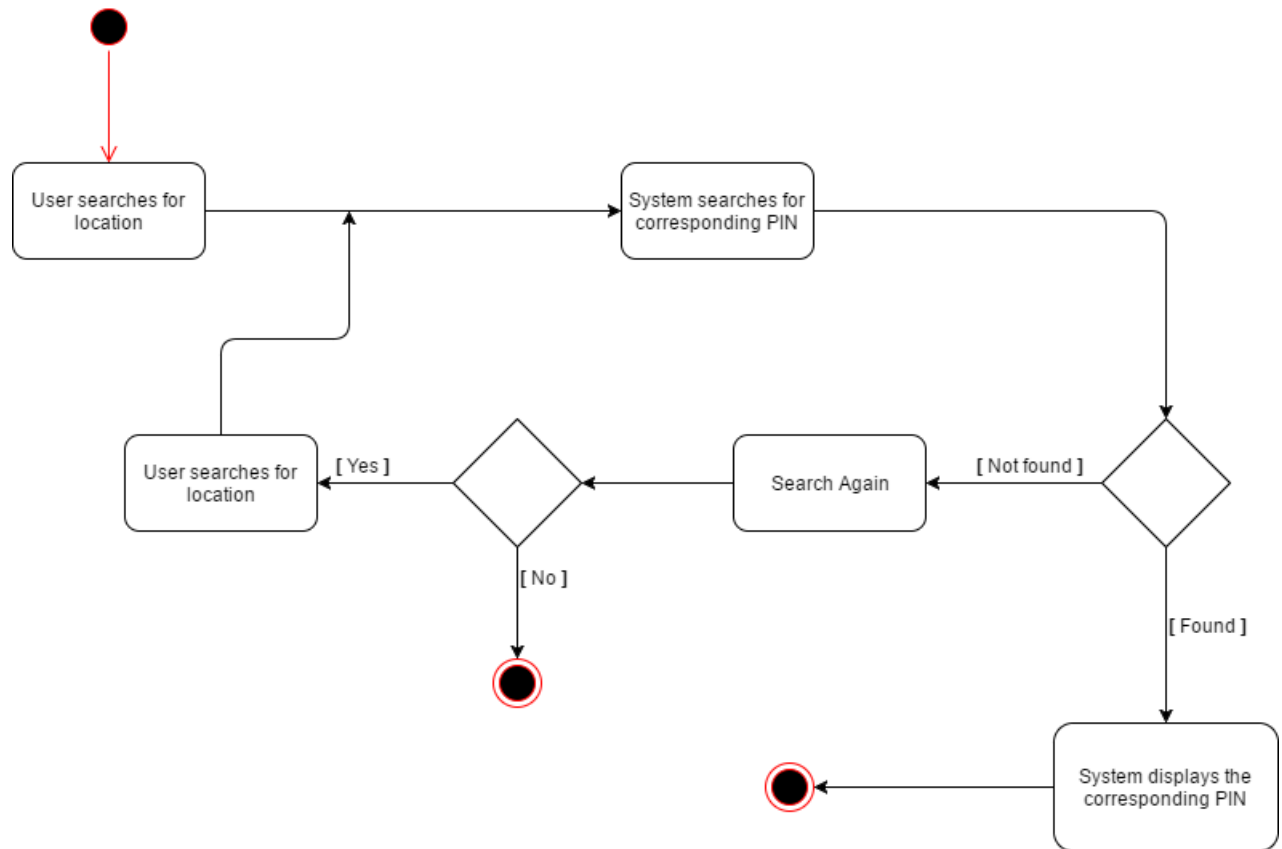
Search For Current Location



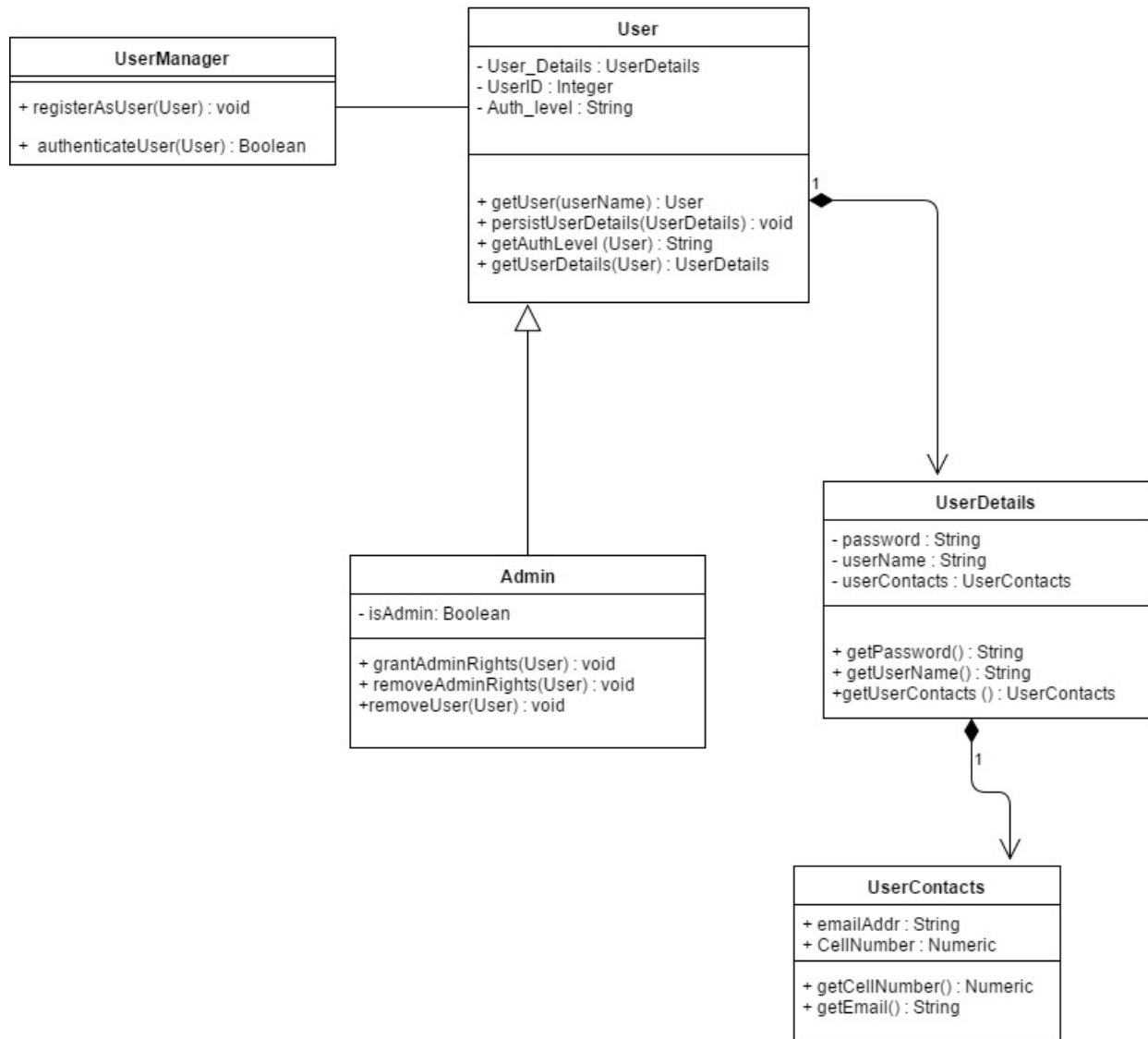
User Save Location

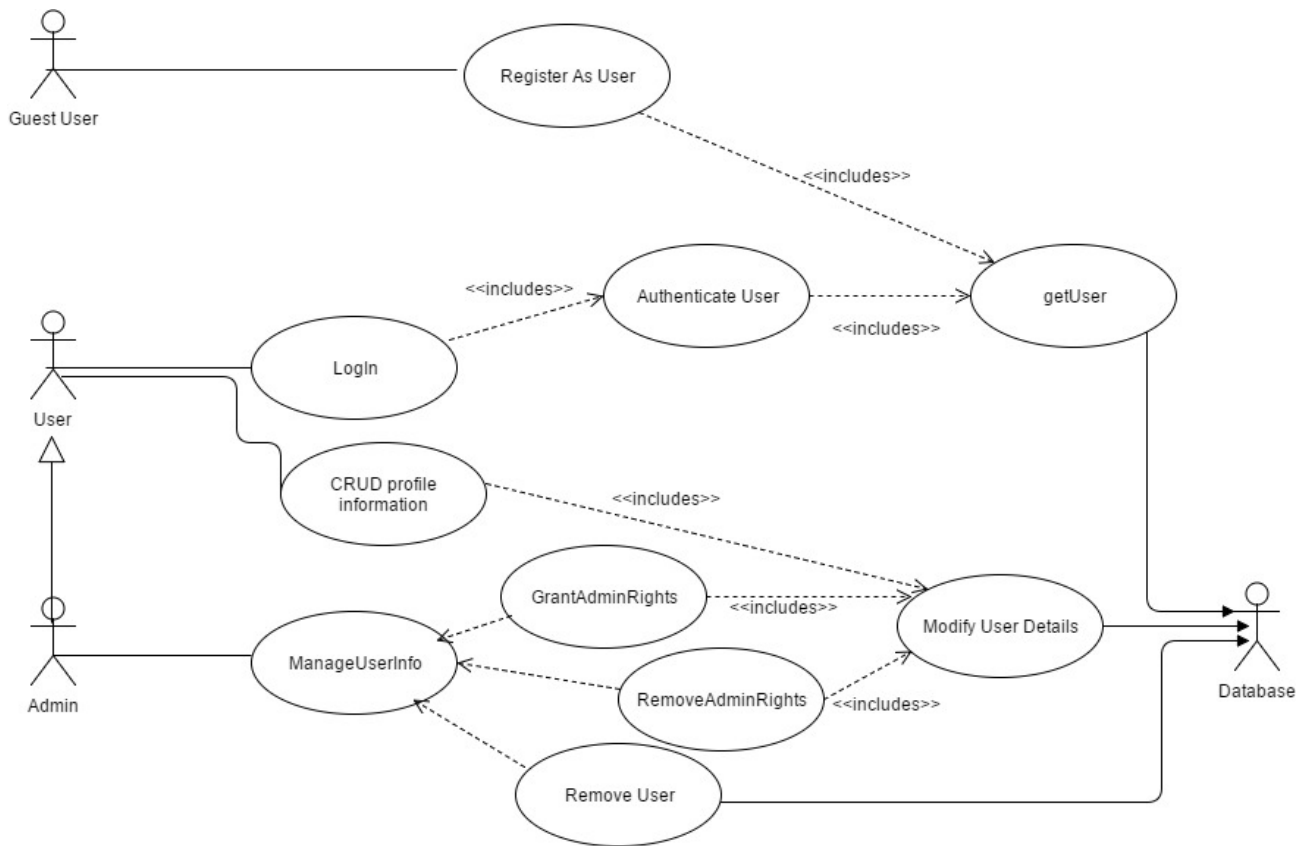


User Searches for Location Activity Diagram

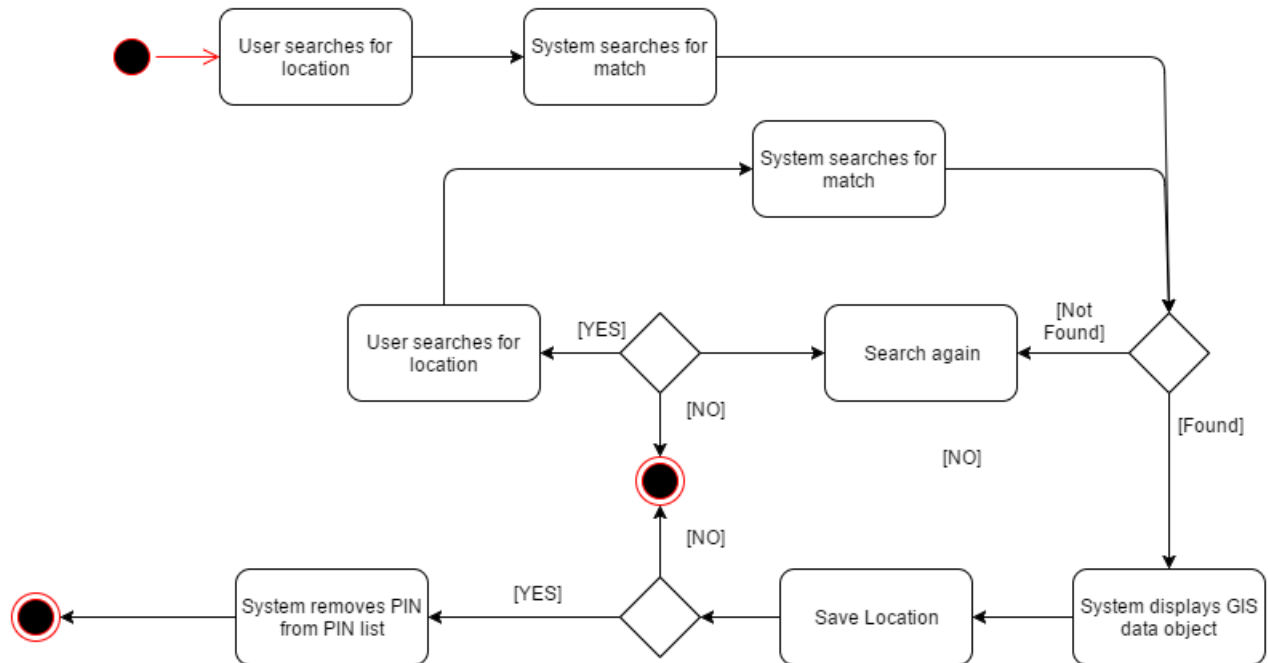


6.3 Users Module

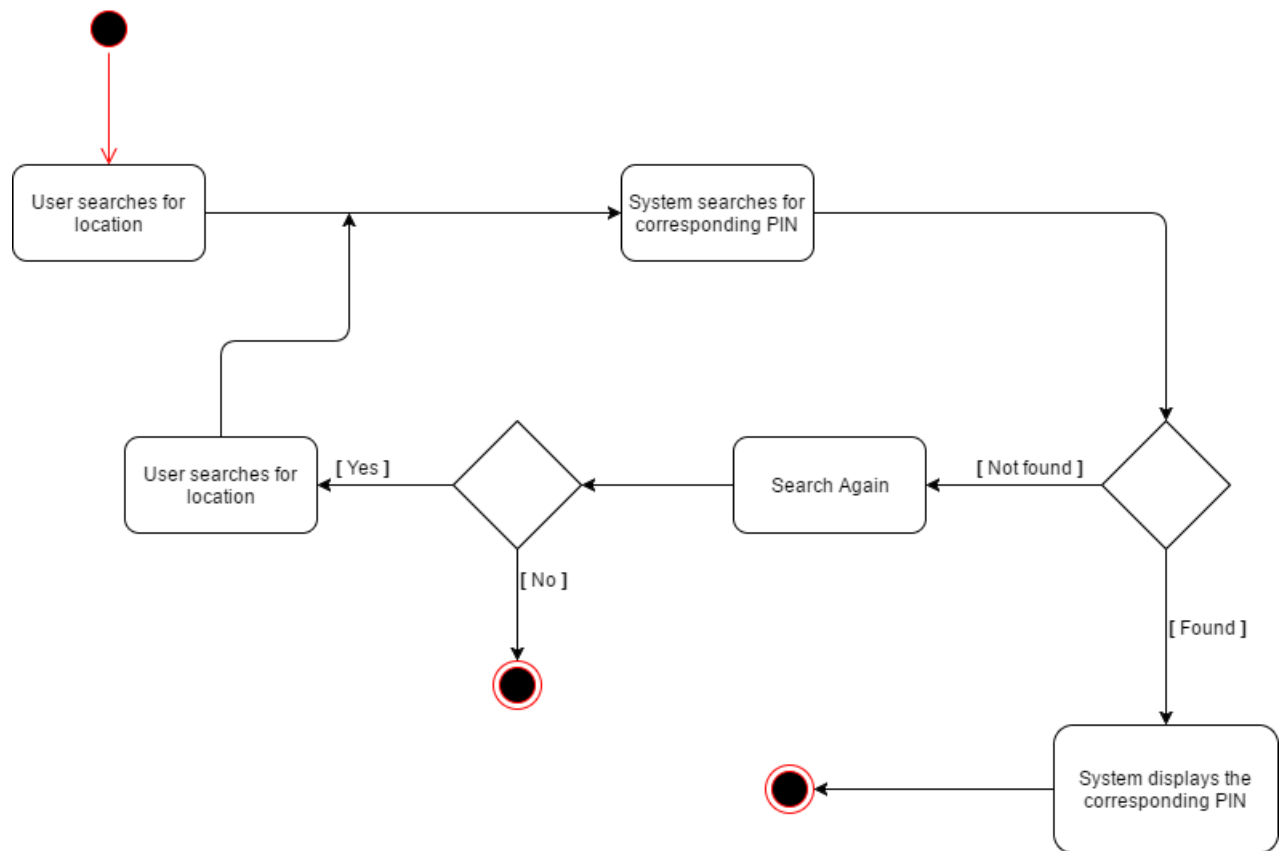




User Save Location



User Searches for Location Activity Diagram



6.4 Notifications Module

