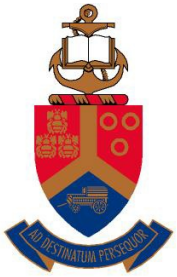


UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

NavUP

Architectural Requirements Specifications and Design

01 March 2017

Team MatLab

Frederick Ehlers
u11061112

Nokuthula Manana
u12064115

Vignesh Iyer
u15031625

Jacobus Marais
u15188397

Stephanie Groutsch
u14293324

Heinrich Burgers
u15059538

Neo Thokoa
u14163285

GitHub https://github.com/FredJEhlers/Matlab

Contents

1 External Interface Requirements

1.1 Navigation

The navigation system communicates to three main entities. The network, the user and the user's device. There needs to be an interface between each one of these entities to allow communication.

User interface: The user interacts with the navigation system via the GUI. This GUI needs to be flexible enough to function on devices of various sizes and operating systems. The interface must allow the user to have some method of control over the navigation system like choosing destinations, waypoints, routes and change in these routes.

The navigation system should also communicates to the user by showing information on the screen and through audio. This information includes messages on the screen, the map, locations on the map and the possible routes on the map.

Other aspects of the user interface like font, icons and the more visual aspects will be decided upon when the GUI is created.

Device Interface: In order to allow communication between the device and the navigation system an interface needs to be created. The navigation system needs access and information from the device in order to function. This includes access to its hardware like the screen, device memory, connection to Wi-Fi or the internet, the location system and to information like the device identity.

The navigation system will be installed on the device along with the application. Being on the device itself will give it the needed access, if permission was granted.

Network Interface: The navigation system functions best with communication between it and the network. This connection is established through the device's connection. The connection can be to a WI-FI network or the mobile data network.

The interface needs to facilitate exchange of information like location, identity, traffic, route information and aspects that might influence how navigation will take place. These aspects can be events, personal preference or personal information like disabilities.

2 Performance Requirements

3 Design Constraints

4 Software System Attributes

5 UML Diagrams for the Chosen Subsystems

Design Patterns have been integrated and a discussion of their use are made below the respective diagrams.

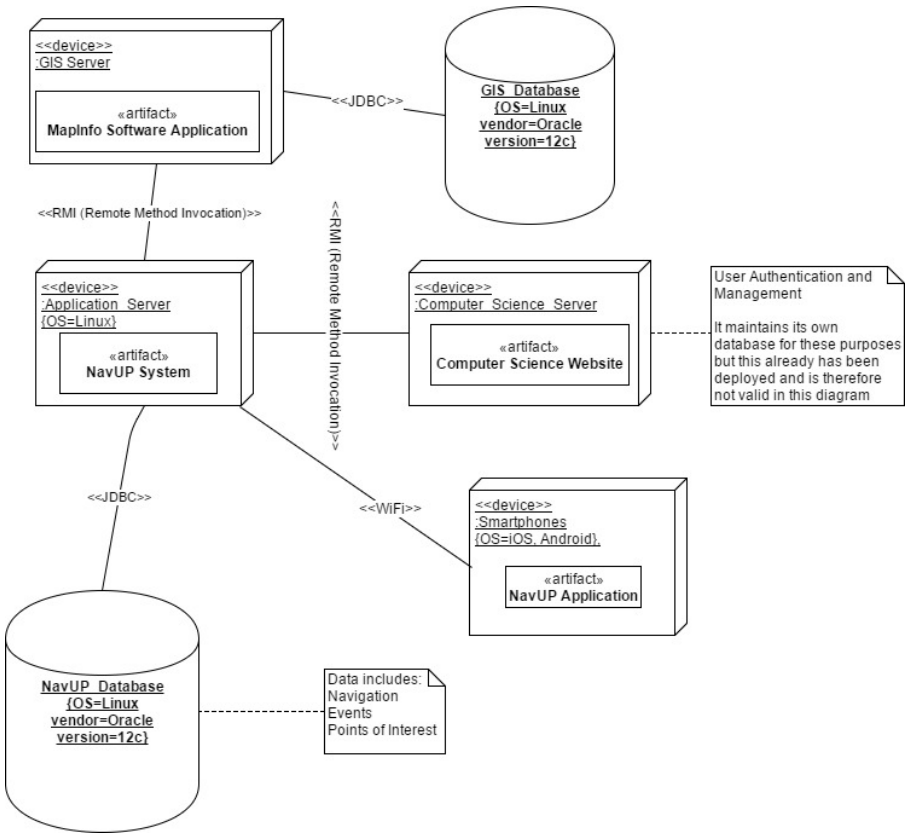


Figure 1: Deployment Diagram for NavUP

5.1 User

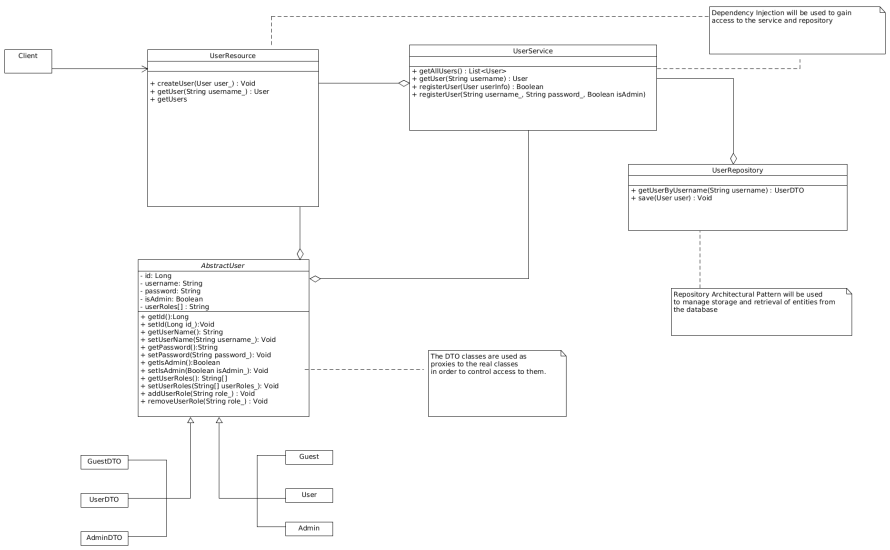


Figure 2: Class Diagram for the User Module

5.2 Navigation

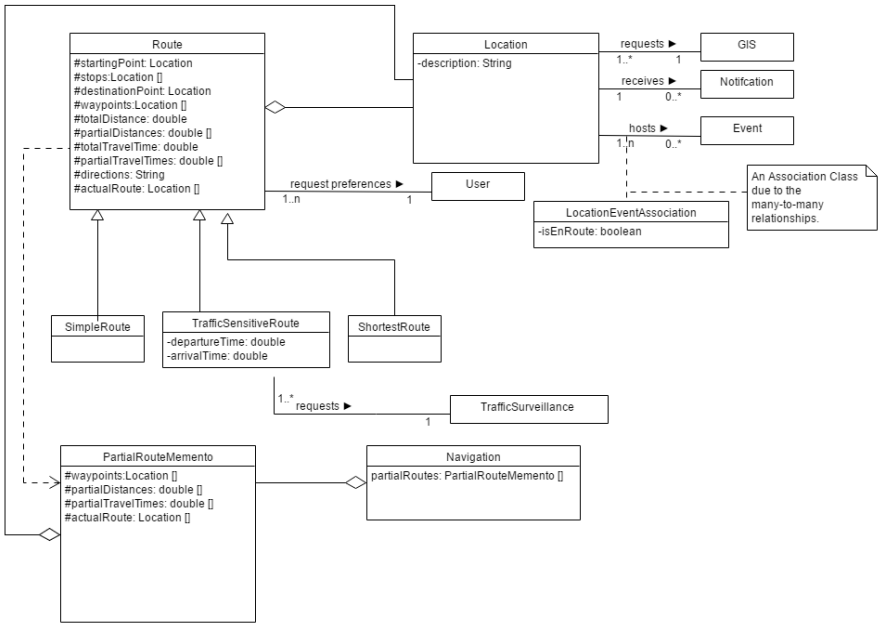


Figure 3: Class Diagram for the Navigations Module

5.2.1 Explanation of Design Patterns in Navigation Module

- Memento Pattern - The Memento Pattern has been used in the navigation module in order to capture the internal state of a "route" object while the route is being calculated.
- The internal state of the route that we would like to capture is the respective arrays for waypoints, partial distances, partial times and actual distances in the route.
- While routing, connection can be lost or the user may diverge from the specified paths during which the system will have to save the internal state so that it can return to this state at a later stage.
- The role-back function in this case would be the re-route function.
- The originator in the module is the Route class.
- The caretaker is the Navigation class.
- The memento is the PartialRouteMemento class.
- Strategy Pattern - The Strategy Pattern has been used in the navigation module in order to define a family of routing algorithms and make them interchangeable.
- The users of the system will specify by clicking on different icons, the type of routing they want.
- The abstraction class would be the Route class which is the interface to the user and is very generic with fundamental functionality.
- The implementation classes would be the SimpleRoute, ShortestRoute and TrafficSenstiveRoute classes that will override the functionality of the superclass with algorithms for their respective needs.

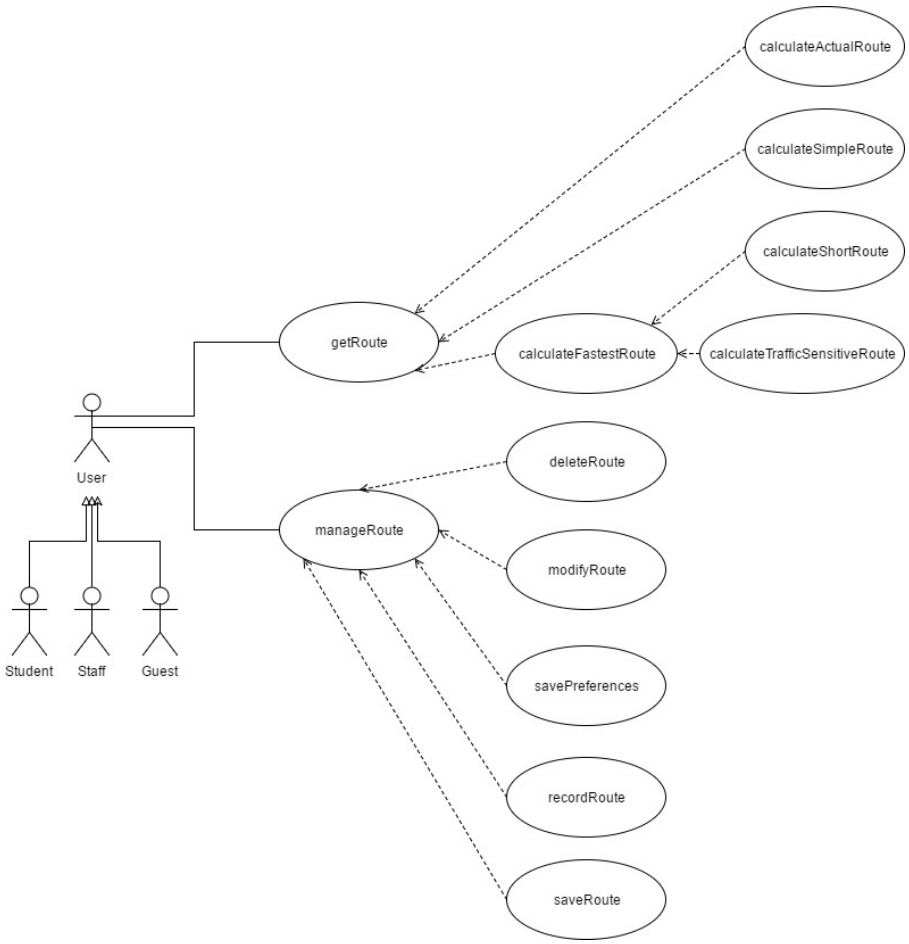


Figure 4: Use Case Diagram for the Navigations Module

5.3 Notifications

5.3.1 External Interface Requirements

5.3.1.1 Software Interface

The mobile application will connect to the database in order to obtain the users' information. One such example includes obtaining information regarding the individuals selected notification medium. The mobile application will also connect to the server in order to update information regarding the users email address and mobile number.

The server will be connected to the notification plugins, which include email and SMS plugins, as well as the application itself in times when notifications are displayed as a pop-up on the users' mobile device.

5.3.1.2 Hardware Interface

The user will interact with the mobile device screen in order to set their notification medium preference that is sent and stored on the server.

When there is a notification, the server will then send data packets to the mobile application by making use of sockets. This will cause the mobile device to display the message as well as vibrate or make a sound.

The user will once again interact with the screen on their mobile device in order to view the notification.

5.3.1.3 User Interface

The user will initially interact with the notifications module through the application's GUI where they will request to be notified about a certain event as well as provide their chosen notification medium.

If the chosen notifications medium is email, the user will receive a generic email with the given notification. The email will provide a description of the notification in its title. If required, the email will provide a link to the relevant website for more information concerning the notification. If the user selected to receive notifications via SMS, a message describing the notification will be sent to each user. Once again, a link will be provided in the SMS for more information concerning the notification as required. If the user decided to receive notifications through the application, a pop-up will appear on the applications' GUI when there is a notification.

5.3.2 Performance Requirements

The system should be able to send out a large number of notifications without any delay (within 5 seconds).

Multiple administrators should be able to add new notifications without failure.

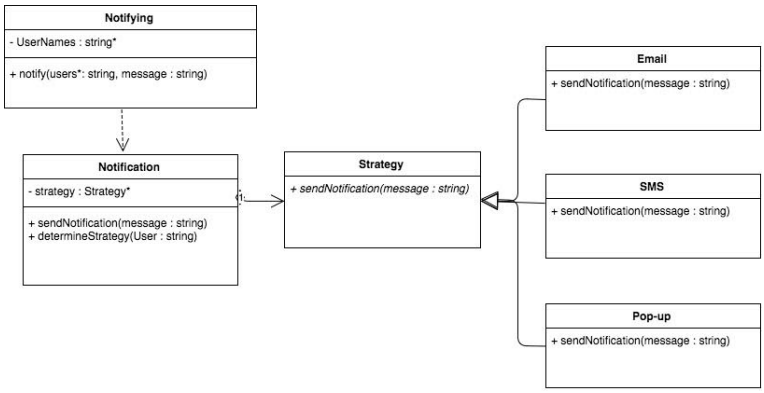
5.3.3 Design Constraints

5.3.4 Software System Attributes

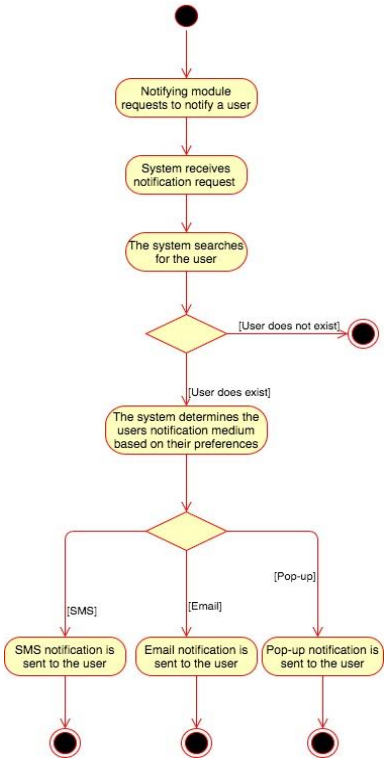
- Flexibility: The system should allow for the addition of new notification means.
- Maintainability: The users' contact information should be updated on a regular basis.
- Security: The users' contact information should be kept secure in order to ensure that third parties do not have access to it.
- Auditability: The system should keep track of the sent and requested notifications.
- Integrability: The system should allow for future expansions.

5.3.5 UML Diagrams

5.3.5.1 Class Diagram



5.3.5.2 Activity Diagram



5.3.6 Technology Choices

5.4 Point of Interest

6 Technology Choices

7 References

Kung, D.C. (2013) Object-oriented software engineering: An agile unified methodology. New York: McGraw Hill Higher Education.