UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Architectural Design Specifications

Team Lisp

Surname, First Name	**Student Number**
Mathaba Ntiko	14012503
Smulders Jacques	15003087
van der Mewe Hendrik	15101283
Walsh Brent	15300201
van der Westhuizen Idrian	15078729

Contents

1 External interface requirements **2**

2 Performance requirements **2**
2.1 Sending and recieving of data 2
2.2 Offline mode 3
2.3 Difference in users 3
2.4 Accessibility and generated paths 3
2.5 Notification 3

3 Design constraints **3**

4 Software system attributes **4**
4.1 Scalability 4
4.2 Reliability 4
4.3 Performance 4
4.4 Security 4
4.5 Interoperability 4
4.6 Manageability 5

5 UML Diagrams **5**
5.1 Class diagrams 6
5.2 Deployment diagrams 8
5.3 Use case diagrams 8

1 External interface requirements

External interface requirements discuss how the external features of the system are supposed to communicate with one another and how information is received and sent.

The User will primarily interact with the NavUP system via their mobile devices. The User will use input such as text, for logins or searching of venues, and gestures, such as clicking or navigating through the map. These input methods should be able to interface with the software of the mobile application to send and receive data through the Wi-Fi or GPS systems of the mobile device.

The device will communicate with its GPS system to gain GPS coordinates through satellite information and send this data with other data through the Wi-Fi. The mobile devices Wi-Fi hardware should send this data/information to the nearest router on campus where more information about the Users current location can be gathered, this new data along with the previous ones are then sent to the external server where the database is hosted.

The server must maintain the database and handle all incoming request based on the data received by the various external devices and methods along the way. Based on the request the server searches the database and retrieves the relative information and sends it back to User through the Wi-Fi in a similar route it came.

Once the data is back on the mobile device the NavUP application should then perform the necessary functions and invoke the necessary classes to gain the required results. The NavUP application will then update the output to represent the data retrieved from the request and that was calculated by the various functions and classes.

2 Performance requirements

2.1 Sending and recieving of data

The NavUP system must be able to send and retrieve relative information in real-time. The User application should be able to constantly update the users´ current position and the generated map in order to accurately display it on the users´ mobile device. In order for this information to be displayed properly the server should be able to filter through the relative data quickly based on the incoming requests in order to send it to user as quickly as possible. The users´ mobile device should also be able to do the necessary calculations based on the data received from the server, since the server will send as little data as possible in order to handle more request more frequently, so it is expected that the mobile device will be able to quickly do the desired functions and calculations.

2.2 Offline mode

Since the Hatfield campus does not constantly have Wi-Fi across the campus it would be best if certain information could run in an offline mode i.e. the user should still be able to see the map and the desired route so navigation is still possible between the few areas that do not have a Wi-Fi signal. This could be done by saving proxies of the server on the users´ mobile device that only contain the needed information to maintain an offline functionality.

2.3 Difference in users

The system contains different types/levels of users and they only share some functionality with one another. The system should inforce and maintain these differences i.e. it should not allow a regular user to add and delete points of interest, only an administrative user should be able to it.

2.4 Accessibility and generated paths

There are many differences in people without there being differences in the types of users our system recognises, therefore the NavUP system should be able to cater for these people with disabilities and different preferences. The path generation takes into account a list of preferences the user has, this list would include information such as ,avoid stairs, if the person is in a wheelchair. Another matter being with just the user interface itself, some users are colour-blind and therefore might need a colour blind mode in-order to follow the map accurately.

2.5 Notification

Notifications should not halt or impede the functionality of any other feature of the system. The user should simply see on the interface that a notification is available and be able to view it later. There are exceptions however, for instance an emergency notification might be able to halt the current interface and path generation in order to forcibly display the notification.

3 Design constraints

The client has provided us with free reign in most areas of the development of the NavUP system. Some of the stated design constraints include:

- The application has to prioritise accessibility, so disabled users can make the most of it.
- The system should have support for multiple devices and services.
- The system should make use of some sort of crowd-sourcing or Wi-Fi triangulation.

4 Software system attributes

4.1 Scalability

The NavUP system needs to be able to accommodate the increases in pedestrian traffic on campus that occur at the end/start of each lecture (around xx:20-xx:30, where xx is an hour during the lecture day). A cloud based infrastructure provider (like Amazon Web Services) would be ideal for this as computing resources can be increased/decreased automatically as needed.

Quantification: The university has about 60 000 students, of which 35 000 are on the Hatfield campus. Therefore the system needs to be able to scale from 1000 (at night) to around 30 000 (peak lecture time).

4.2 Reliability

The system needs to be very reliable, as some users will be moving away from other solutions in favour of using this application, especially disabled users.

Quantification: The system should have a fully-operational uptime of 99.5% every month, using the remainder for maintenance downtime (preferably in off-peak times).

4.3 Performance

The application must provide it's functionality to the user in a timely manner, otherwise the user will get frustrated and not use the system. This performance attribute is measured from the time the user interacts with the system until the user receives feedback.

Quantification: The latency of the system (time taken to respond to an event) must be less than 5 seconds at all times.

4.4 Security

The user should not be allowed to perform actions outside of the scope of the system, whether it be malicious or accidental. This also aids in the user experience, because the amount of valid actions a user can take is reduced, minimizing the cognitive load of decision-making.

Quantification: The system must not allow access or modification of unauthorized information.

4.5 Interoperability

The NavUP system must be able to integrate with the other systems that the University of Pretoria already has in place. This is particularly useful when

logging in or signing up, because the system can use a user's existing details from the UP Portal.

Quantification: The system must be able to integrate with the UP Portal for login, sign up and user details.

4.6 Manageability

NavUP must have consistent performance at all times of the day, and to achieve this a system administrator must be able to manage the system.

Quantification: The system must expose instrumentation to monitor and debug performance.

5 UML Diagrams

The 4 modules we decided to model and design further are :

- User management
- Navigation
- Points of interest
- and Notification

5.1 Class diagrams

5.2 Deployment diagrams

5.3 Use case diagrams