2019

Hofstra Campus Map Application

REQUIREMENTS SPECIFICATION

J. ABRAHAM; F. CHEN; X. CHEN; X. HU; W. SANTOS

Contents

1. Introduction

This application will allow the end user to navigate the Hofstra campus. Its primary focus will be on new students and visitors who will be navigating the campus for the first time, allowing them to orient themselves and navigate the campus. By inputting a destination, the user will receive guidance on how to navigate towards that location.

2. Requirements Gathering

Requirements Discovery: The purpose of the application is to assist users in the navigation and familiarization of the Hofstra campus. For navigation, the application will, in tandem with a pre-established map of the campus, building locations, and predetermined road values, calculate the ideal path between two locations to create an ideal path for the user to follow. Aside from navigation, descriptions of the buildings will also be provided to facilitate familiarity with the campus environment. Its scope is currently limited to Android devices due to logistic reasons, as not everyone on the team is familiar with Swift and XCode. This eliminates the risk of cross platform bugs as our focus is limited to one type of system.

Requirements classification and organization:

- **Navigation System:** These function groups work to assist in the navigation of the user from one location to another in coordination with the 3rd Party Map System.
- **Familiarization:** These groups work to contain data relevant to the individual locations on the Hofstra campus.
- **Development Language:** Java
- **Application Management**: This pertains to the specifications for the application's size, hardware reliability, etc.

Hardware Constraints: The mobile application is constrained by the preset environment of the 3rd Party Map System within the mobile phone. This 3rd Party will be preselected by the development team and optimized accordingly. Android versions and device types will be another constraint to consider as hardware specifications will vary between devices.

Required system performance: The application will be optimized to balance minimum power consumption with efficient resource usage. The performance of the application will be tested based on the connectivity of the mobile application, resource utilizer, and user interface. The application performance on a mobile phone is measured in the following two categories:

- **Application Performance (AP)**
- **Hardware Power Consumption Performance (HPCP)**

Each category will determine the level of performance for the application. For example:

Application Performance (AP): Optimized

HPCP: Optimized

This represents a Level 3 performance. Each category is scored with either 1 or 0, 1 being optimized and 0 not being optimized.

Requirement Negotiation and Prioritization: The goal of the project is to navigate Hofstra University; thus, the project will prioritize the implementation of the **Navigation System.** Another important function is the Data management. The navigation system will need preset JSON Data for proper pathfinding. If time constraints become a problem, the goal is to finish all the necessary functional requirements, and drop functions based on the lowest value on the priority list. The priority list, once fully defined, will be ordered in such a way that all essential functions are on top, and unnecessary functions are on the bottom.

3. Requirements Specification

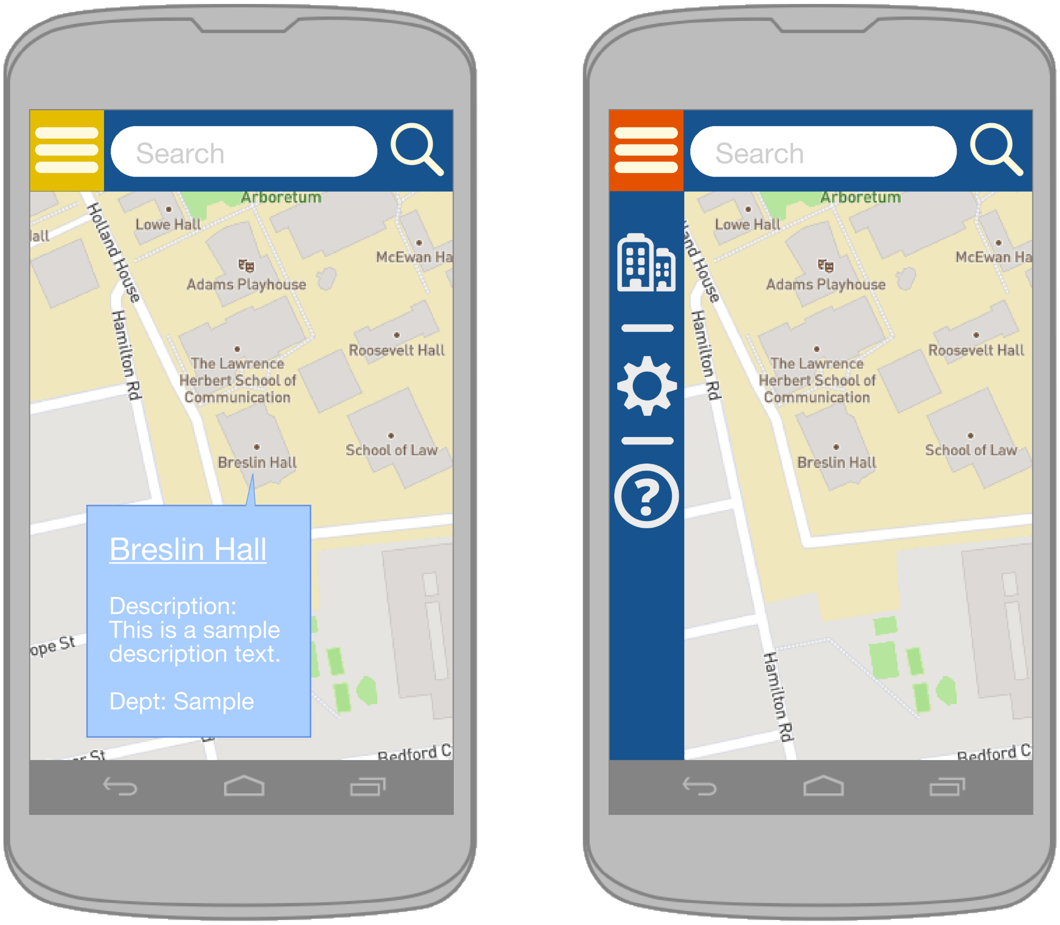
The following sections are the **general requirements** and *specifications* for the application:

3.1 The user shall enter a valid destination within Hofstra University.

3.1.2. The user needs to enter a valid name, halls, structure, bridges, buildings, and areas.

3.2 The destination shall be highlighted, and a colored solid line will highlight the pathway to the destination.

3.2.1. The application shall implement a UI similar to the mockup below:



3.3 The application shall show the shortest route from the current location to the destination.

3.3.1. The application shall display measurement as in the English (Imperial) System of Measurement (e.g. feet, miles).

3.3.2. The application shall display the closest structures to the current location.

3.3.3 The application shall use the closest structures to determine the shortest route to the destination

3.4 The application shall display a full representation of the path from starting point to destination.

3.7 The application shall be optimized for the target device.

3.7.1. The application shall be power efficient on the current generation of the Android system.

3.7.2. The application shall be resource efficient on the current generation of the Android system.

3.7.3. The application shall be optimized for storage based on standard Android application size.

3.7.4 The application shall follow the Official User Interface Guideline for Android Devices.

3.8 The application shall be easily maintainable by anyone with sufficient training.

3.8.1. The application shall have an update, either scheduled or manual, with changes ranging from location names to new structures to new areas.

3.8.2. The update shall not stop the older version of the application from working properly; although the older version of the application might contain outdated information.

In general, the application will only be used inside Hofstra University as a method to help newcomers find and locate specific buildings. The application will find the shortest path to the destination and also display information about distance. The application is perfectly safe and harmless to the user's device.

4. System Architecture

The application may be divided into the following three main components:

4.1 **Mapping**: The map will be represented as an undirected graph of nodes, containing both

the relevant information about the node as well as its location data. Nodes may represent

structures, such as buildings or monuments, path start and end points, or path intersections. The map will consist of building and monument nodes connected via path nodes.

Relevant information may come in the form of names, hours of operation, a picture or description of the building, or IDs for system use. This list is non-exhaustive.

4.2 **Navigation**: All navigation begins with a start node and end node, as specified by the user. It will then make an informed search along the nodes to find the shortest route, then return to the user a visual representation of the route. The route will be recalculated should the user deviate too far from the path; the deviation distance will be predetermined by the developer team.

4.3 **Display and Interaction**: The user may expect a visual representation of the map, including the route provided as well as the building destination, should navigation currently be underway. Paths shall appear connected but may not display distinct nodes. Displayed nodes may show their names to help orient the user. Some paths may also show their names for the same reason. Overall, the user interface will be designed with visual clarity and an intuitive user experience in mind.

5. Requirements Validation

To ensure that data accuracy and integrity within the system aligns with product safety, the validation process will be primarily focused on finding problems with the requirements and ensuring that the specified requirements are in accordance with the end user's needs.

5.1 Internal Validation

During internal validation, ensure that every project member can understand and express the stakeholders' requirements thoroughly and comprehensively.

5.1.1. Simple checks

Check if the application meets the requirements after every programming section. Examples of these include, but are not limited to: Mapping, Navigation, etc.

5.1.2. Functional test design

5.1.2a. The identification of functions that the software is expected to perform

Ensure the application meets the requirements.

5.1.2b. The test case data which is based on the function's specifications

New students and visitors may need different cases.

5.1.2c. The comparison of actual and expected outputs

Compare the actual and expected outputs carefully; if there are discrepancies, backtrack to identify the cause.

5.1.2d. To check whether the application works as per the customer's needs

Final check for the testing phase.

5.1.3. Reviews and inspections

Review all validation processes and inspect the work of all project members, ensuring that the work meets the requirements specifications.

5.2 External Validation

During external Validation, asking stakeholders or users about whether the software meets their needs or not. Different level of users(stakeholders) needs different functions. Thus, make sure the software meets all level of users' needs.

5.3 Gathering the Internal and External Validation

Gathering the result of both internal and external validation to conduct final inspections of the application.

6. Requirements Change & Change Control

6.1 Change

During the development and validation period, new data and information may arise. When proposals are made to alter the course of the project, however slightly, meetings will be held to discuss the feasibility and superiority of the changes, carefully deciding whether or not to implement the changes. If most project members agree on the change, the changes will be implemented and updated on the relevant documents.

Hofstra University, 2019. Version 1.4, *"Software Requirement Specification"* 8 | P a g e

6.2 Change Control

Quality management systems (QMS) and information technology (IT) systems are two major ways to moderate change control and ensure that changes to the application are introduced in a controlled and coordinated manner. These will reduce the likelihood of errors appearing within the project. The goals of a change control procedure are primarily to minimize the disruption to services and development setbacks.

7. Requirements Update Pipeline

Requirements that did not meet the project schedule have been moved here. These are features that will be implemented if there is an opportunity in the future. Note that they have not been prioritized and maintain the same numbers as they had originally.

3.1.1. The user will need to be a certain distance within Hofstra University

3.4 The application shall be able to zoom in/focus on the user once the tracking has started and continue following the user until the destination is reached.

3.4.1. The user shall be represented as a certain shape.

3.4.2. The shape will move as the user moves.

3.4.3. The camera of the application shall always follow the shape.

3.5 There shall be a clear difference between representations/shapes of the building, pathways, and other structures.

3.5.1. Different types of structures shall be represented with distinct icons.

3.5.2. The user shall able to click on the icon in order to open a secondary display to see the detailed information about the structure.

3.6 The application shall be understandable and usable by people of age 7 or above and shall be organized in such a way that minimum confusion occurs.

3.6.1. The user shall be able to quickly understand how to use the application after a very brief tutorial upon initial application launch.

3.6.2. The user shall be able to review the above tutorial at any time while using the application.

3.8.3. The application shall be designed in a way such that a future plan to port the application to another system is viable.