Software Requirements Specification

Team Dodger

24 February 2017

Contents

1 Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms and Abbreviations

1.4 References

1.5 Overview

2 Overall Description

2.1 Product Perspective

2.2 Product Function

2.3 User Characteristics

2.4 Constraints

2.5 Assumptions and Dependencies

3 Specific Requirements

This section expands on the functional requirements of the system. It gives a detailed description of the system and all of its use cases.

3.1 External Interface Requirements

3.2 Functional Requirements

This section includes all functional requirements in detail. It includes all use case diagrams, Actor-System interaction diagrams as well as a traceability matrix.

3.2.1 Use cases

1. Navigation Subsystem

    1.1. Get current location

        i. **Description:** The NavUP system must be able to determine a users location at any point in time while the user is on the Hatfield campus. The location must be determined regardless of whether the user is indoors or outdoors.

        ii. **Precondition:** The user must have an active account and must be within range of WiFi routers.

        iii. **Postcondition:** The users location is determined and displayed.

1.2. Search location

i. **Description:** The NavUP system must provide functionality that enables a user to search for any location (lecture hall, day-house, restaurant) on the Hatfield Campus.

ii. **Precondition:** The user must have an active account

iii. **Postcondition:** Matching locations are returned to the user. If no buildings match the search criteria, an appropriate error message is displayed.

1.3. View location details

i. **Description:** The NavUP system must allow users to view details related to specific locations. This could include restaurant menus, lecture hall timetable schedules as well as images of the buildings.

ii. **Precondition:** The user must have an active account and a valid location must be selected on the map.

iii. **Postcondition:** Relevant location details shown to user.

1.4. View places of interest

i. **Description:** The NavUP system must be able to display places of interests to a user based on their current location. This will include places like restaurants and day-houses that must be displayed in a list form.

ii. **Precondition:** The user must have an active account and their current location must be known.

iii. **Postcondition:** Relevant places of interest are listed and displayed to the user based on their location.

1.5. Navigate to location

i. **Description:** The NavUP system must be able to provide directions and navigate to a location given the users current location as well as a desired destination. The system should calculate the most optimal route by looking at the shortest path as well as pedestrian traffic.

ii. **Precondition:** The user must have an active account. The users current location must be known and the must have specified a destination through the search interface.

iii. **Postcondition:** The user is provided with directions from their current location to their desired destination.

1.6. Show pedestrian traffic

i. **Description:** The NavUP system must be able to display pedestrian traffic on campus in the form of a heatmap. When navigating to a specified location, the system must show traffic on that specific route. A user should also be able to view an overall heatmap of the campus to see traffic.

ii. **Precondition:** Users must all have the NavUP app installed and must be registered in order for them to show up on the heatmap.

iii. **Postcondition:** A heatmap of the campus is displayed.

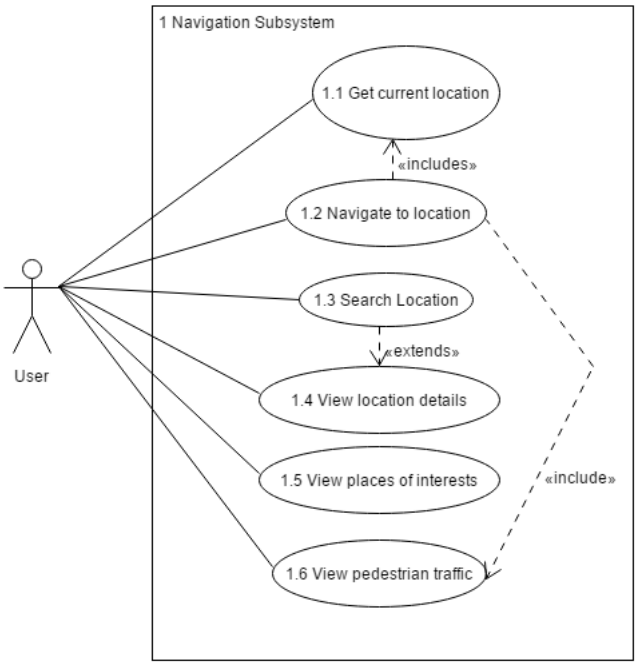

Figure 1: Navigation Subsystem

2. Location Management Subsystem

3. User Account Management Subsystem

4. Entertainment Subsystem

   4.1. View events

   i. **Description:** The NavUP system must enable users to view all events that are happening around campus in chronological order. The system should suggest events to a user based on their preferences and most visited locations.

ii. **Precondition:** The user must have an active account and must be logged in.

iii. **Postcondition:** Various campus-wide events are returned to the user.

4.2. Save event

i. **Description:** The NavUP system must enable users to save events that they are interested so that they can be viewed later.

ii. **Precondition:** The user must have an active account, must be logged in and there must be events available to save.

iii. **Postcondition:** An event is saved.

4.3. Delete event

i. **Description:** The NavUP system must enable a user to delete any saved events

ii. **Precondition:** The user must have an active account, must be logged in and must have saved events

iii. **Postcondition:** A saved event is deleted .

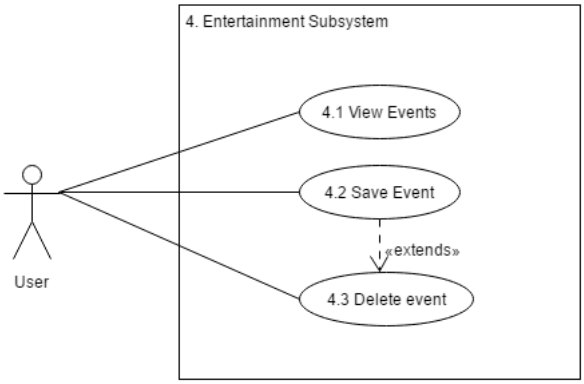

Figure 2: Navigation Subsystem

5. Achievements Subsystem

5.1. Set Goals

i. **Description:** The NavUP system must enable users to Set Goals for Achievements created on the system.

ii. **Precondition:** An Achievement must be created on the system before Goals can be set for it.

iii. **Postcondition:** Every Goal set must be associated with an achievement.

5.2. View Leaderboard

i. **Description:** The NavUP system will have a leaderboard interface that will showcase the top x users of the system based on the goals and achievements they have completed successfully.

ii. **Precondition:** A maximum number of users to view on the leaderboard must be specified.

iii. **Postcondition:** None

5.3. Record Steps

i. **Description:** The NavUP system must be able to record steps that a user has taken.

ii. **Precondition:** The application must be running on their chosen device i.e. not closed.

iii. **Postcondition:** The application must be able to, at any given moment inform the user of how many steps they have.

5.4. Challenge friend

i. **Description:** The NavUP system must enable users to interact with other users on the system and challenge them to complete achievements.

ii. **Precondition:** The users in question must be enrolled on the system.

iii. **Postcondition:** The user being challenged must be notified of the challenge and who it is from.

6. Administration Subsystem

6.1. Manage Users

i. **Description:** The NavUP system must enable an administrator to add users to the system.

ii. **Precondition:** User details must be valid and correct (e.g a valid email address)

iii. **Postcondition:** User must be notified that they have been successfully registered onto the system.

6.2. Manage Locations

i. **Description:** The NavUP system must enable the administrator to mark important locations on the system. E.g. The CSC is an important location that should be highlighted when using the system.

ii. **Precondition:** The user managing the locations must have administrator access.

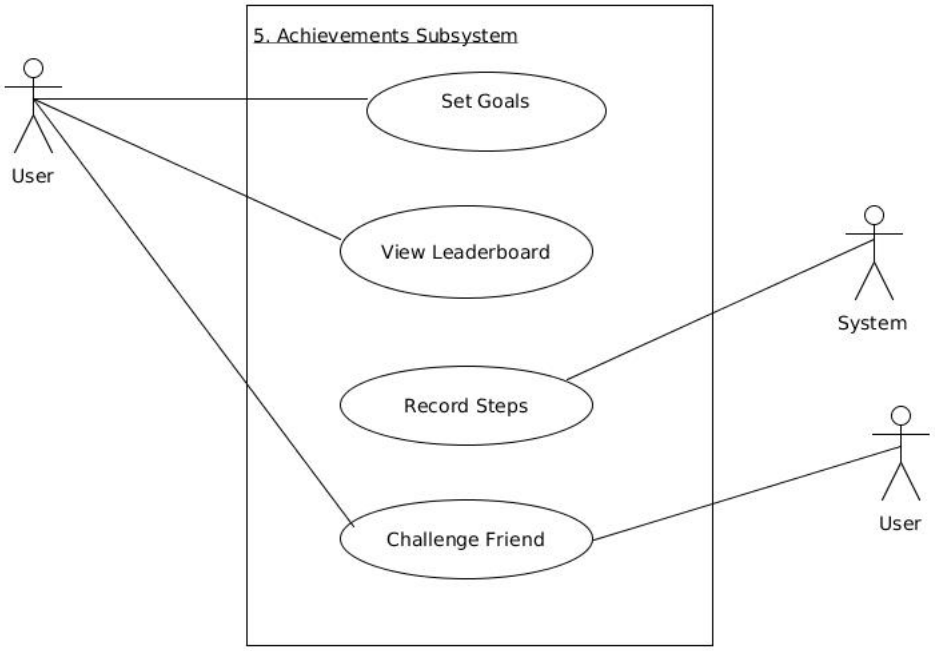

Figure 3: Achievements Subsystem

iii. **Postcondition:** None

6.3. Manage Events

i. **Description:** The NavUP system must enable an administrator to create and publish, update and remove events on the system.

ii. **Precondition:** The user managing the event must have administrator access.

iii. **Postcondition:** All users subscribed to the event must be notified of changes.

6.4. Notify Users

i. **Description:** The NavUP system must enable an administrator to notify users of various actions being performed on the application.
This includes actions such as Publishing events etc.

ii. **Precondition:** The users being notified should be subscribers of the event

iii. **Postcondition:** None

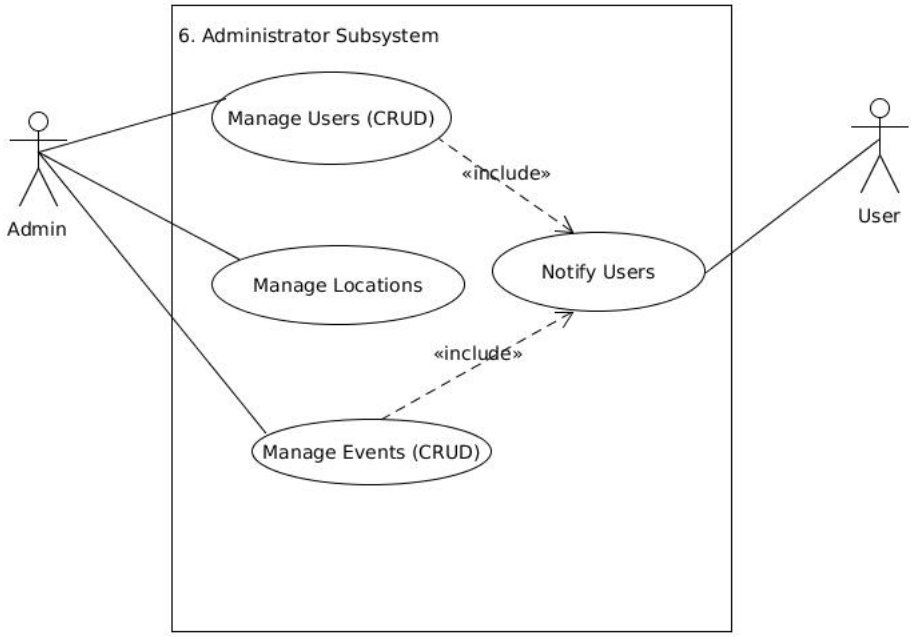

Figure 4: Administrator Subsystem

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software System Attributes

3.6 Other Requirements

References

[1] D. C. Kung, *Object Oriented Sofware Engineering.* Dummy Publisher, 2015.