Requirements Specification

Navjot Singh Virk, Student Number: X13112406, Software Development Stream 4th Year

Project Name: MeetingRoom Pro

Github: https://github.com/Virksaabnavjot/MeetingRoom-Pro

Website: http://roomassistant.navsingh.org.uk

Documentation: Github Repository here

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this document is to set out the requirements for the development of Meeting Room Pro project. That assist users to find meeting rooms among multiple buildings in multiple locations, with ease of use and review rooms and find relevant information about the rooms.

The intended customers of the application are medium to big size corporate organisations with multiple buildings and hundreds of rooms. And educational institutions like universities and colleges. The target users of the application includes but not limited to – company employees (managers, team leads, interns), facilities department, It department and in educational institutes college staffs and students will be major target audience.

The intended audience for this document is myself, potential clients of the project and the academic staff at the National College of Ireland.

1.2 Project Scope

The scope of the project is to develop quality native IOS application allowing the users to use it on their iPads and iPhones. The application shall show the current location of user in a map view which will allow users to see how far they are from the meeting room, also user shall receive notifications, and the app shall also allow booking and availability feature and some other features. Which will attract users and potential client to get involved with the project also form my research I found no product similar to this application exist in the market. And, the project scope is more stronger due to the fact the application works out of the box (*application client can be easily used for any organisation without any prior changes or any dependencies. Only, different .csv files with building and meeting room information tailor the application for any organisation*).

I was involved in several discussions with Manuel Saez, my manager at SAP SE (internship) and my academic supervisor Christina Muntean. To elicit the following requirements

Here is a list of main features of the application:

- The app shall allow the user to book a meeting room while specifying the date and time of the meeting.
- The app should be able to read users device calendar and check if a meeting exists, the user receives a notification.
- The app shall be able to run smoothly on all iPhones and iPads.
- The app shall have responsive GUI / interface which allows the app to function properly and look good and function properly on different screen sizes.
- The system shall have a database in order to store building and meeting rooms information.
- The database shall have the capability to store geo-spatial data.
- The app must make a secure connection with backend the web service using authentication.
- The system shall have a web service to return data in either JSON/XML format, that will be displayed in the app.
- The application shall have a simple web based dashboard to allow the administrator to add more buildings and rooms with ease.
- The application shall be scalable.
- The application shall have Geo location feature so that when a user is within a certain radius of the meeting room or if outside building gets notified through a friendly notification. “Hey you are this close (*approx. location*) to your destination”.
- The application shall have a gallery to showcase photos of the room and an upload feature that allows the users to upload photos of a room which will also be available to other users to see and benefit from.
- The application shall have an expandable map with building drawn as a polygon and room as point and with apple maps most of the buildings have 3d view available which will allow the user to see the building in 3d view and find where the meeting room is located in the building.
- The application shall provide functionality to make a call from the app to key contacts of the meeting room like IT or Facilities department.

The application will be implemented using open-source programming language Swift 3 and Xcode Editor with continuous unit testing. And on the backend, SAP Hana Spatial and a JavaScript based web service.

1.3 Definitions, Acronyms, and Abbreviations

SAP SE – German Multinational Software Company.

NCI – National College of Ireland

SAP Hana – In memory, column oriented RDBMS.

RDBMS - Relational Database Management System.

Hana Spatial – feature of SAP Hana that allows us to store Geo-Spatial Data.

Geo Spatial Data – Data that has geographic positioning information.

GUI – Graphical User Interface

Shall - The term "*shall"* is used in this document to describe features which the system must have.

Should - The term "*should*" is used in this document to describe a feature which the system should have but may not.

Customer – The term "customer" is used in this document in context of potential users and clients.

2 User Requirements Definition

The requirements that I have outlined in the project scope section of this document are that of the customer after D-Shop design thinking and several consultations with myself.

The most of user requirements for this project were collected/suggested by potential users during my internship in SAP SE and during the project idea presentation at SAP D-Shop innovation day on 17th and 18th August 2016 through design thinking (*heading 3*). And some were suggested my my manager at SAP SE and some by project supervisor at NCI.

MeetingRoom Pro will be an all-in-one room assistant application that will help users find meeting rooms, view them on map of building and also find relevant information like floor, room type, capacity, which building the room is located in, city, country and number of floors in that building.

The user will not have to register in order to access the app content. However, in order to review, book meeting rooms, and contribute towards gallery, the user will have to register.

Users will complete a standard registration in order to gain full access to the application features. Not only will registered users have access to content and features but will also have the permission to delete, moderate or upload content.

The users can have different account types –

- Guest *(unregistered)* – View Content Permission
- Basic User *(registered user)* – Review rooms and upload photos to gallery permission
- Moderator/Admin – Review, Upload and Delete content permissions

3 Design Thinking for User Requirements Definition

Design thinking is a methodology used by designers like myself to solve complex problems and find desirable solutions for clients. Design thinking can help all sorts of organizations uncover new ways of thinking and doing things.

These are the design thinking process that were used during user requirements definition –

Empathized with Users – Observed, engaged and tried to understand user requirements.

Because as designer / developer problems we try to solve are rarely ours so we need to understand user first to develop an application for them.

Defined the problem – Brought clarity and focus to design space and framed the problem.

Idea Generation – Allowed the users to come up with idea, features and functionality they would like to see in the application. Through brain storming and putting ideas white board.

4 Requirements Specification

All requirements should be verifiable. For example, experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

4.1 Functional requirements

This section lists the functional requirements in **ranked order**. Functional requirements describe the possible effects of a software system, in other words, *what* the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe *how* the system accomplishes its functional requirements. Each functional requirement should be specified in a format similar to the following:

Short, imperative sentence stating highest ranked functional requirement.

4.1.1 Use Case Diagram

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

The Use Case Diagram provides an overview of all functional requirements.

4.1.2 Requirement 1 <name of requirement in a few words>

The heading of this section should read, e.g., “Requirement 1: User registration” or “Requirements 1: Participant takes test”

4.1.2.1 Description & Priority

A description of the requirement and its priority. Describes how essential this requirement is to the overall system.

4.1.2.2 Use Case

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

Scope

The scope of this use case is to …….

Description

This use case describes the ……..

Use Case Diagram

Diagram should highlight actors and uses cases……..

Flow Description

Precondition

The system is in initialisation mode……..

Activation

This use case starts when an <Actor>……

Main flow

1. The system identifies the …….
2. The <Actor> ……..(See A1)
3. The system ……..(See E1)
4. The <Actor> …….

Alternate flow

A1 : <title of A1>

1. The system ……..
2. The <Actor> …….
3. The use case continues at position 3 of the main flow

Exceptional flow

E1 : <title of E1>

4. The system ……..

5. The <Actor> …….
6. The use case continues at position 4 of the main flow

Termination

The system presents the next …….

Post condition

The system goes into a wait state

4.1.3 Requirement 2 <name of requirement in a few words>

4.1.3.1 Description & Priority

A description of the requirement and its priority. Describes how essential this requirement is to the overall system.

4.1.3.2 Use Case

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

Scope

The scope of this use case is to …….

Description

This use case describes the ……..

Use Case Diagram

Diagram should highlight actors and uses cases……..

Flow Description

Precondition

The system is in initialisation mode……..

Activation

This use case starts when an <Actor>……

Main flow

5. The system identifies the …….
6. The <Actor> ……..(See A1)
7. The system ……..(See E1)
8. The <Actor> …….

Alternate flow

A1 : <title of A1>

7. The system ……..
8. The <Actor> ……..
9. The use case continues at position 3 of the main flow

Exceptional flow

E1 : <title of E1>

10. The system ……..
11. The <Actor> ……..
12. The use case continues at position 4 of the main flow

Termination

The system presents the next …….

Post condition

The system goes into a wait state

List further functional requirements here, using the same structure as for Requirements 1 & 2. Most systems would have at least five main functional requirements.

4.2 Non-Functional Requirements

Specifies any other particular non-functional attributes required by the system. Examples are provided below. **Remove the requirement headings that are not appropriate to your project.**

4.2.1 Performance/Response time requirement

4.2.2 Availability requirement

4.2.3 Recover requirement

4.2.4 Robustness requirement

4.2.5 Security requirement

4.2.6 Reliability requirement

4.2.7 Maintainability requirement

4.2.8 Portability requirement

4.2.9 Extendibility requirement

4.2.10 Reusability requirement

4.2.11 Resource utilization requirement

5 Interface requirements

This section describes how the software interfaces with other software products or users for input or output. Examples of such interfaces include APIs, web services, shared memory, data streams, and so forth. Most systems would have a GUI. Add more subsections for other interfaces as reuired.

5.1 GUI

Include mock-ups of the key pages or stages of the system. Explain how they are linked. Explain how you addressed above requirements in the design. It is important that the mock-ups are in line with the functional requirements above, e.g., if one of your requirements is “user registration” then one of the screens listed in this section should show a registration page.

5.2 Application Programming Interfaces (API)

Explain which interfaces your system offers or which are used by your system. Examples include Google maps and Weka.

6 System Architecture

Use a class diagram to outline the structure of the system. Explain briefly why you have chosen this architecture. You might want to use Visio or Rational Rose to create these.

7 System Evolution

This section describes how the system could evolve over time.