Software Requirements Specification

MY-University

Abdelrahman Mamdouh Ali:4141149
Mark Wafik Wiliam:4141127
Abdelrahman Mohamed:4141228
Mohamed Magdy Ali:4141166
Moatsem Salah:4141152

Table of Contents

1. Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the "My University" software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

1.2 Scope

The "My University" is a CMS mobile application which helps student to stay connect with every thing related to their university . started from their courses and social media groups to their university buses locations .The application should be free to download from either a mobile phone application store or similar services.

Teacher assistant can provides grades , assignments and materials using the web-portal and check for assignment plagiarism online between other student submissions and over the internet. the students deliver their assignments . view their grades and download their materials and track their buses .student can register their courses Online .and receive notification for every important announcement.

1.3 Definitions, acronyms, and abbreviations

Table 1 - Definitions

Term	Definition
User	Someone who interacts with the mobile phone application
Admin/Administrator	System administrator who is given specific permission for managing and controlling the system
TA	some one that can post grades , assignments and materials for the student
Web-Portal	A web application which present special facilities for runiversity staff

GPS	Global Positioning System
GPS-Navigator	An installed software on mobile phone which could provide GPS connection and data, show locations on map and find paths from current position to defined destination
Application Store	An installed application on mobile phone which helps user to find new compatible applications with mobile phone platform and download them from Internet
Stakeholder	Any person who has interaction with the system who is not a developer.
DESC	Description
RAT	Rational
DEP	Dependency
TAG	A unique, persistent identifier contained in a PLanguage statement [2]
GIST	A short, simple description of the concept contained in a PLanguage statement [2]
SCALE	The scale of measure used by the requirement contained in a PLanguage statement [2]
METER	The process or device used to establish location on a SCALE contained in a PLanguage statement [2]
MUST	The minimum level required to avoid failure contained in a PLanguage statement [2]
PLAN	The level at which good success can be claimed contained in a PLanguage statement [2]
WISH	A desirable level of achievement that may not be attainable through available means contained in a PLanguage statement [2]
DEFINED	The official definition of a term contained in a PLanguage statement [2]

1.4 References

[1] IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.

[2] Feldt R,"re_lecture5b_100914", unpublished.

[3] Davis M A, "Just Enough Requirements Management: Where Software Development Meets Marketing", New York, Dorset House Publishing, 2005.

[4] Karlsson J, "A Cost-Value Approach for Prioritizing Requirements", Norges Teknisk-Naturvitenskapelige Uni. 1997

1.5 Overview

The remainder of this document includes three chapters and appendixes. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product.

The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

The fourth chapter deals with the prioritization of the requirements. It includes a motivation for the chosen prioritization methods and discusses why other alternatives were not chosen.

The Appendixes in the end of the document include the all results of the requirement prioritization and a release plan based on them.

2. Overall description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

2.1 Product perspective

This system will consist of two parts: one mobile application and one web portal. The mobile application will be used to make the student able to connect easily to their grades ,assignements and even their buses locations while the web portal will be used for managing the information about the restaurants and the system as a whole.

The product is supposed to be an open source, under the GNU general Public License. It is a cross platform mobile application with client-server model. The "My University"

will provide the proper methods to make all the possible communication methods for all students with their colleagues and the university staff.

2.2 Product functions

The product depending on more than one module. the first is a Course Management System with web portal for the university staff and mobile application for the students . the staff will be able to set assignments , quizzes and grades they can send announcements to all students . beside every assignments submitted from a student there will be 2 percentages for plagiarism the first is a compare between all the assignments that is submitted from all the students and the other comparing the submitted assignments will the resources on the Internet the percentage is a percentage of similarity and they can post public and anonymous surveys for the students . the student get an announcement for the the quizzes , assignments and grades . they can download materials and submit assignments only before deadlines . they can chat with each others and with the staff with private message but with permission from the receiver first . and as public message as a post . all the students is divided automatically by courses , groups , sections and individuals so every action on the system can be done according to this groups . the second module is Room Management System only for the staff and student organizations. staff can add , delete or update room information . and every room have a name and a maximum number that it can take .
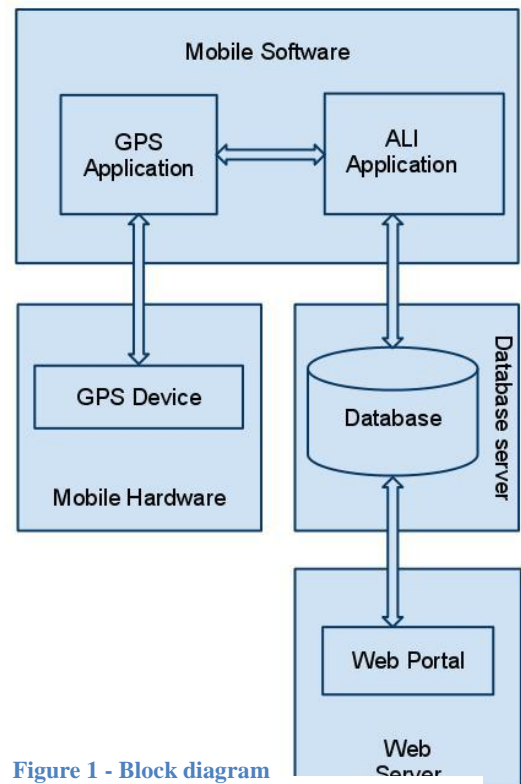

Figure 1 - Block diagram

4

the third module is a bus tracking system every bus on the university will have a tracking chip that is working with GPS that identify every bus location so the students can know the place of the buses that they suppose to ride live .

2.3 User characteristics

our user characteristics is kind of dynamic it's build on permissions so every set of permissions make a new user role but basically they will come around two roles

the most of our users will be two types of users that interact with the system: users of the mobile application : students, and university staff. Each of these two types of users has different use of the system so each of them has their own requirements.

The mobile application users (students) can use the application to check the the quizzes , assignments and grades . they can download materials and submit assignments only before deadlines . they can chat with each others and with the staff with private message but with permission from the receiver first . and as public message as a post .

the staff will be able to set assignments , quizzes and grades they can send announcements to all students . beside every assignments submitted from a student there will be 2 percentages for plagiarism the first is a compare between all the assignments that is submitted from all the students and the other comparing the submitted assignments will the resources on the Internet the percentage is a percentage of similarity and they can post public and anonymous surveys for the students .

2.4 Constraints
The system is constrained by the system of every university because different universities have different routines even the students have different experiences so the sys will have to be customized for universities

The Internet connection is also a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.

Both the web portal and the mobile application will be constrained by the capacity of the database. Since the database is shared between both application it may be forced to queue incoming requests and therefor increase the time it takes to fetch data.

2.5 Assumptions and dependencies
One assumption about the product is that it will always be used on mobile phones that have enough performance. If the phone does not have enough hardware resources available for the application, for example the users might have allocated them with other applications, there may be scenarios where the application does not work as intended or even at all.

Another assumption is that the GPS components in all phones work in the same way. If the phones have different interfaces to the GPS, the application need to be specifically adjusted to each interface and that

would mean the integration with the GPS would have different requirements than what is stated in this specification.

2.6 Apportioning of requirements

In the case that the project is delayed, there are some requirements that could be transferred to the next version of the application. Those requirements are to be developed in the third release

3. Specific requirements

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.
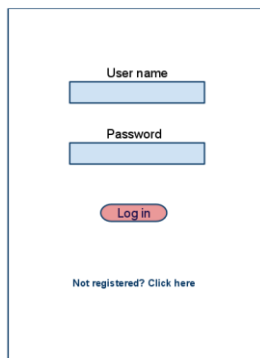
3.1 External interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

3.1.1 User interfaces

A first-time user of the mobile application should see the log-in page when he/she opens the application, see Figure 2. If the user has not registered, he/she should be able to do that on the log-in page.

If the user is not a first-time user, he/she should be able to see the search page directly when the application is opened, see Figure 3. Here the user chooses the type of search he/she wants to conduct.

Every user should have a profile page where they can edit their e-mail address, phone number and password, see Figure 4. Also, the user can set the mobile application to his/her preferred language. The "P" icon shows where the user can click to navigate to his/her profile page.

Figure 2 - Login page

Figure 3 – Search page

Figure 4 – Profile page

In Figure 5, the list view for the results is shown. When a user searches by price, this view should be the default one. The sorting header allows the user to sort the results according to price, restaurant name, distance, restaurant type and specific dish. Each result item includes information about the restaurants, a link to the restaurant's web-page and an information link, which provides a more detailed description of the restaurant. There is also a filtering option, where the user can choose to filter the results by increasing or decreasing the price or distance range, see Figure 7.

In the map view each restaurant is represented by a pin, see Figure 6. Next to every pin there is an information link which provides a more detailed description of the restaurant, as mentioned for the list view. The same filtering option, as for the list view, is included in the map view.

The restaurant owners and administrators interact with the system through a web-portal, see Figure 8. A restaurant owner should be able to register on the web-portal in order to log in and manage the restaurant information. An administrator should also be able to log in to the web-portal where he/she can administer the system by for instance editing restaurant or user information.

Figure 5 – List view

Figure 6 – Map view

Figure 7 – Filter menu

Figure 8 – Web Portal

3.1.2 Hardware interfaces

Since neither the mobile application nor the web portal have any designated hardware, it does not have any direct hardware interfaces. The physical GPS is managed by the GPS application in the mobile phone and the hardware connection to the database server is managed by the underlying operating system on the mobile phone and the web server.

3.1.3 Software interfaces

The mobile application communicates with the GPS application in order to get geographical information

8

about where the user is located and the visual representation of it, and with the database in order to get the information about the restaurants, see Figure 1. The communication between the database and the web portal consists of operation concerning both reading and modifying the data, while the communication between the database and the mobile application consists of only reading operations.

3.1.4 Communications interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for both the mobile application and the web portal.

3.2 Functional requirements

This section includes the requirements that specify all the fundamental actions of the software system.

3.2.1 User Class 1 - The User

3.2.1.1 Functional requirement 1.1

ID: FR1
TITLE: Download mobile application
DESC: A user should be able to download the mobile application through either an application store or similar service on the mobile phone. The application should be free to download.
RAT: In order for a user to download the mobile application.
DEP: None

3.2.1.2 Functional requirement 1.2

ID: FR2
TITLE: Download and notify users of new releases
DESC: When a new/updated version or release of the software is released, the user should check for these manually. The download of the new release should be done through the mobile phone in the same way as downloading the mobile application.
RAT: In order for a user to download a new/updated release.
DEP: FR1

3.2.1.3 Functional requirement 1.3

ID: FR3
TITLE: User registration - Mobile application
DESC: Given that a user has downloaded the mobile application, then the user should be able to register through the mobile application. The user must provide university ID, password and e-mail address. The user can choose to provide a regularly used phone number.
RAT: In order for a user to register on the mobile application.
DEP: FR1

3.2.1.4 Functional requirement 1.4

ID: FR4
TITLE: User log-in - Mobile application
DESC: Given that a user has registered, then the user should be able to log in to the mobile application. The log-in information will be stored on the phone and in the future the user should be logged in automatically.
RAT: In order for a user to register on the mobile application.
DEP: FR1, FR3

3.2.1.5 Functional requirement 1.5

ID: FR5
TITLE: Retrieve password
DESC: Given that a user has registered, then the user should be able to retrieve his/her password by e-mail.
RAT: In order for a user to retrieve his/her password.
DEP: FR1

3.2.1.6 Functional requirement 1.6

ID: FR6
TITLE: Mobile application - feeds
DESC: Given that a user is logged in to the mobile application, then the first page that is shown should be the feeds page. The student will get feeds for the post , announcements , grades , materials and surveys from different courses that he registered in and every post have a sender and a group and a date and files.
RAT: In order for a student to get the latest news of the university.
DEP: FR4

3.2.1.7 Functional requirement 1.7

ID: FR7
TITLE: Mobile application - comment on the posts in the feeds
DESC: every student can post replys on the posts that in the university news feeds and can reply to the reply of the other users .

RAT: replying to the posts.
DEP: FR6

3.2.1.8 Functional requirement 1.8

ID: FR8
TITLE: Mobile application - survey
DESC: student can vote on public surveys and private surveys :
so we have to kind of surveys
1- public : the name of the voters will appear on the submissions
2- private : the submissions on the surveys will be private so there will not be a name on the submissions .

the survey will be a MCQ questions for all the courses at a time

RAT: student will submit survey .
DEP: FR6

3.2.1.9 Functional requirement 1.9

ID: FR9
TITLE: Mobile application - polls
DESC: A user will vote on polls that will be uploaded by any staff member .
RAT: user will vote on polls that staff will make .
DEP: FR7, FR8

3.2.1.10 Functional requirement 1.10

ID: FR10
TITLE: Mobile application - courses
DESC: every student have a list of courses every semester that he will enrolled in . every course will be divided into groups and every group have sections and every section have list of students and every course will have at leasts a Teacher assistant and a doctor and have a list of announcements and materials and grades .

RAT: student will be enroll in courses.
DEP: FR7, FR8

3.2.1.11 Functional requirement 1.11

ID: FR11
TITLE: Mobile application - announcements
DESC: student will receive an announcement and every announcement should be related to specific course or from collage management .
RAT: student will get announcements from different courses.
DEP: FR7, FR8

3.2.1.12 Functional requirement 1.12

ID: FR12
TITLE: Mobile application - grades
DESC: A student will have a list of grades with every course divided to quizzes , assignment and different exams and every course will have this list with a total grade and every student have a CGPA for all the course that the student already attended
RAT: In order for a user to get his grades.
DEP: FR8

3.2.1.13 Functional requirement 1.13

ID: FR13
TITLE: Mobile application - Search by destination
DESC: A user should be able to input a maximum and a minimum distance, according to his/her position. By default the minimum distance is set to 0 km and the maximum to 10 km. The user should be able to input a higher or lower maximum distance and a higher minimum distance than set by default. The result is displayed in a map view by default.
RAT: In order for a user to search by destination.
DEP: FR7

3.2.1.14 Functional requirement 1.14

ID: FR14
TITLE: Accepted input for price and destination search
DESC: Integers should be accepted as input when a user searches by price or destination. If the system receives an invalid input the user should be informed and prompted to insert an accepted input.
RAT: In order for a user to search with valid input.
DEP: FR12, FR13

3.2.1.15 Functional requirement 1.15

ID: FR15
TITLE: Mobile application - Search by restaurant type

DESC: A user should be able to select a restaurant type in a given list as input. The result is displayed in a map view by default.

RAT: In order for a user to search by restaurant type.

DEP: FR7

3.2.1.16 Functional requirement 1.16

ID: FR16

TITLE: Mobile application - Search by specific dish

DESC: A user should be able to select a specific dish in a given list as input. The result is displayed in a map view by default.

RAT: In order for a user to search by specific dish.

DEP: FR7

3.2.1.17 Functional requirement 1.17

ID: FR17

TITLE: Mobile application - Free-text search

DESC: A user should be able to conduct a search by providing either restaurant name, restaurant description, restaurant address, restaurant type or restaurant menu in the free-text search field. The result is displayed in a map view by default.

RAT: In order for a user to search through the free-text search.

DEP: FR7

3.2.1.18 Functional requirement 1.18

ID: FR18

TITLE: Mobile application - No match found

DESC: If no match is found the user should be informed but kept on the search page in order to get the possibility to conduct a new search right away.

RAT: In order for user to conduct a new search if no match is found.

DEP: FR5

3.2.1.19 Functional requirement 1.19

ID: FR19

TITLE: Mobile application - Sorting results

DESC: When viewing the results in a list, a user should be able to sort the results according to price, distance, restaurant type, specific dish or restaurant name.

- When sorting by restaurant name, specific dish or restaurant type the results should be ordered alphabetically.
- When sorting by price the results should be ordered from cheapest to most expensive.
- When sorting by distance the results should be ordered from closets to furthest distance according to the user's position.

When the sort button for a specific search option is clicked, then the order should be reversed and ordered in a descending matter. If the sort button is clicked again the order of the results should be reversed.
RAT: In order for a user to sort results in a list.
DEP: FR8

3.2.1.20 Functional requirement 1.20

ID: FR20
TITLE: Mobile application - Filtering results
DESC: When viewing the results in a list or a map, a user should be able to filter the results in a filtering menu. The filtering options include:

- increasing or decreasing the maximum distance
- increasing or decreasing the maximum price
- choosing a restaurant type
- choosing a specific dish

When filtering the results, only the existing results shall be affected and a new search query should not be sent.
RAT: In order for a user to filter results in a list or a map.
DEP: FR7, FR8

3.2.1.21 Functional requirement 1.21

ID: FR21
TITLE: Mobile application - Profile page
DESC: On the mobile application, a user should have a profile page. On the profile page a user can edit his/her information, which includes the password, e-mail address and phone number. A user should also be able to choose what language the mobile application should be set to. The different language choices are Swedish, English, Spanish and French.
RAT: In order for a user to have a profile page on the mobile application.
DEP: FR1

3.2.2 User Class 2 - Restaurant Owner

3.2.2.1 Functional requirement 2.1

ID: FR22
Feature: Create an account
In order to create an account
A restaurant owner
Should register on the web-portal

    Scenario: Required information for registration
    Given the restaurant owner wants to create an account
    And the restaurant owner does not have an account

When the restaurant owner registers on the web-portal by providing user-name
And password
And address
And e-mail address
And phone number
Then the restaurant owner should be able to apply for verification

Scenario: Full information for registration
Given the restaurant owner wants to create an account
And the restaurant owner does not have an account
When the restaurant owner registers on the web-portal by providing user name
And password
And address
And e-mail address
And phone number
And mobile number
Then the restaurant owner should be able to apply for verification

Scenario: Confirmed registration
Given the restaurant owner has applied for verification
And has not received a confirmation e-mail after registration
When the restaurant owner receives a confirmation e-mail
Then the restaurant owner should be able to log in

3.2.2.2 Functional requirement 2.2

ID: FR23
Feature: Restaurant owner log-in
In order to use the system
A restaurant owner
Should be logged in to the web-portal

Scenario: Successful log-in
Given the restaurant owner wants to log in
When the restaurant owner logs in with his/her account
Then the restaurant owner should be logged in as a restaurant owner

Scenario: Retrieve password
Given the restaurant owner wants to log in
And has lost the password
When the restaurant owner enters his/her email address in the "Retrieve password" form
And submits the form
Then the restaurant owner should receive an email containing the password

3.2.2.3 Functional requirement 2.3

ID: FR24
Feature: Manage information
In order to manage information
A restaurant owner
Should be logged in to the web-portal

    Scenario: Show fields for managing information
    Given the restaurant owner is logged in
    When the restaurant owner wants to manage information
    Then the restaurant owner should be able to manage information in a form

    Scenario: Filling in mandatory fields
    Given the restaurant owner wants to fill in the mandatory fields of the form
    When the restaurant owner provides average price
    And address
    And e-mail address
    And phone number
    And restaurant name
    Then the restaurant owner has filled the mandatory fields of the form

    Scenario: Filling in optional fields
    Given the restaurant owner of a restaurant wants to fill in optional fields in the form
    When the restaurant owner provides restaurant description
    And menu
    And type of restaurant
    And picture of restaurant
    And mobile phone
    Then the restaurant owner has filled in optional fields in the form

    Scenario: Filling in menu field
    Given the restaurant owner wants to fill in the menu field in the form
    When the restaurant owner provides dish name
    And dish description
    And dish price
    Then the restaurant owner has filled in the menu field in the form

    Scenario: Adding information with mandatory fields
    Given the restaurant owner has filled in the mandatory fields of the form
    When the restaurant owner submits the form
    Then the information about the restaurant should be added

    Scenario: Adding information with mandatory and optional fields
    Given the restaurant owner has filled in the mandatory fields of the form

And filled in one or more optional fields of the form
When the restaurant owner submits the form
Then the information about the restaurant should be added

Scenario: Deleting information
Given the restaurant owner is logged in
And information exists
When the restaurant owner deletes information
Then the information should be deleted

Scenario: Editing information
Given the restaurant owner is logged in
And information exists
When the restaurant owner edits information
Then the information should be edited

3.2.2.4 Functional requirement 2.4

ID: FR25
Feature: Restaurant owner - Selecting preferred language on the web-portal
In order to understand the web-portal
A restaurant owner
Should be able to select a preferred language for the web-portal

Scenario: Select English as preferred language
Given the restaurant owner wants to select a preferred language
When the restaurant owner selects English as a new language
Then the web-portal will show all text in English

Scenario: Select Swedish as preferred language
Given the restaurant owner wants to select a preferred language
When the restaurant owner selects Swedish as a new language
Then the web-portal will show all text in Swedish

Scenario: Select French as preferred language
Given the restaurant owner wants to select a preferred language
When the restaurant owner selects French as a new language
Then the web-portal will show all text in French

Scenario: Select Spanish as preferred language
Given the restaurant owner wants to select a preferred language
When the restaurant owner selects Spanish as a new language
Then the web-portal will show all text in Spanish

3.2.3 User Class 3 - Administrator

3.2.3.1 Functional requirement 3.1

ID: FR26
Feature: Administrator log in
In order to administer the system
An administrator
Should be logged in to the web-portal

> **Scenario: Successful log-in**
> Given the administrator wants to log in
> When the administrator logs in with an administrator account
> Then the administrator should be logged in as an administrator

3.2.3.2 Functional requirement 3.2

ID: FR27
Feature: Verify restaurant owner
In order to allow a restaurant owner to use the system
An administrator
Should be able to verify the restaurant owner

> **Scenario: Verify a restaurant owner**
> Given the administrator is logged in
> When the administrator verifies a restaurant owner
> Then the restaurant owner should be able to log in
> And the restaurant owner should be notified by a confirmation email

> **Scenario: Reject a restaurant owner**
> Given the administrator is logged in
> When the administrator rejects a restaurant owner
> Then the restaurant owner should not be able to log in
> And the restaurant owner should be notified by a rejection email

3.2.3.3 Functional requirement 3.3

ID: FR28
Feature: Manage restaurant types
In order to have a list of restaurant types
An administrator
Should be able to manage the restaurant types

> **Scenario: Add a new restaurant type**
> Given the administrator is logged in
> When the administrator creates a new restaurant type
> Then the new restaurant type should be added to the list of restaurant types

Scenario: Editing an existing restaurant type

Given the administrator is logged in

When the administrator edits an existing restaurant type

Then the restaurant type should be updated in the list of restaurant types

Scenario: Delete a restaurant type

Given the administrator is logged in

When the administrator deletes a restaurant type

Then the deleted restaurant type should be removed from the list of restaurant types

3.2.3.4 Functional requirement 3.4

ID: FR29

Feature: Manage restaurant dishes

In order to have a list of dishes

An administrator

Should be able to manage the dishes

Scenario: Add a new dish

Given the administrator is logged in

When the administrator creates a new dish

Then the new dish should be added to the list of dishes

Scenario: Editing an existing dish

Given the administrator is logged in

When the administrator edits an existing dish

Then the dish should be updated in the list of dishes

Scenario: Delete a dish

Given the administrator is logged in

When the administrator deletes a dish

Then the deleted dish should be removed from the list of dishes

3.2.3.5 Functional requirement 3.5

ID: FR30

Feature: Manage restaurant information

In order to manage restaurant information

An administrator

Should be logged in to the web-portal

Scenario: Add restaurant information

Given the administrator is logged in

When the administrator adds restaurant information

Then the information should be added to the restaurant

Scenario: Delete restaurant information

Given the administrator is logged in

And information about a restaurant exists

When the administrator deletes the information

Then the information about the restaurant should be deleted

Scenario: Edit restaurant information

Given the administrator is logged in

And information about a restaurant exists

When the administrator edits the information

Then the information about the restaurant should be edited

3.2.3.6 Functional requirement 3.6

ID: FR31

Feature: Manage users

In order to keep track of the users

An administrator

Should be able to manage the users

Scenario: Edit an existing user's information

Given the administrator is logged in

When the administrator edits an existing user

Then the user information should be updated

Scenario: Delete/Inactivate an existing user

Given the administrator is logged in

When the administrator deletes an existing user

Then the user should be deleted

3.2.3.7 Functional requirement 3.7

ID: FR32

Feature: Manage restaurant owners

In order to keep track of the restaurant owners

An administrator

Should be able to manage the restaurant owners

Scenario: Add a new restaurant owner

Given the administrator is logged in

When the administrator creates a new restaurant owner

Then the new restaurant owner should be added

Scenario: Edit an existing restaurant owner

Given the administrator is logged in

When the administrator edits an existing restaurant owner

Then the restaurant owner information should be updated

Scenario: Delete an existing restaurant owner

Given the administrator is logged in

When the administrator deletes an existing restaurant owner
Then the restaurant owner should be deleted
And the restaurant information should be deleted

3.2.3.8 Functional requirement 3.8

ID: FR33
Feature: Administrator - Selecting preferred language on the web-portal
In order to understand the web-portal
An administrator
Should be able to select a preferred language for the web-portal

    Scenario: Select English as preferred language
    Given the administrator wants to select a preferred language
    When the administrator selects English as a new language
    Then the web-portal will show all text in English

    Scenario: Select Swedish as preferred language
    Given the administrator wants to select a preferred language
    When the administrator selects Swedish as a new language
    Then the web-portal will show all text in Swedish

    Scenario: Select French as preferred language
    Given the administrator wants to select a preferred language
    When the administrator selects French as a new language
    Then the web-portal will show all text in French

    Scenario: Select Spanish as preferred language
    Given the administrator wants to select a preferred language
    When the administrator selects Spanish as a new language
    Then the web-portal will show all text in Spanish

3.3 Performance requirements

The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the system performance.

3.3.1 Prominent search feature

ID: QR1
TITLE: Prominent search feature
DESC: The search feature should be prominent and easy to find for the user.
RAT: In order to for a user to find the search feature easily.
DEP: none

3.3.2 Usage of the search feature

ID: QR2
TITLE: Usage of the search feature
DESC: The different search options should be evident, simple and easy to understand.
RAT: In order to for a user to perform a search easily.
DEP: none

3.3.3 Usage of the result in the list view

ID: QR3
TITLE: Usage of the result in the list view
DESC: The results displayed in the list view should be user friendly and easy to understand. Selecting an element in the result list should only take one click.
RAT: In order to for a user to use the list view easily.
DEP: none

3.3.4 Usage of the result in the map view

ID: QR4
TITLE: Usage of the result in the map view
DESC: The results displayed in the map view should be user friendly and easy to understand. Selecting a pin on the map should only take one click.
RAT: In order to for a user to use the map view easily.
DEP: none

3.3.5 Usage of the information link

ID: QR5
TITLE: Usage of the information link
DESC: The information link should be prominent and it should be evident that it is a usable link. Selecting the information link should only take one click.
RAT: In order to for a user to use the information link easily.
DEP: none

3.3.6 Response time

ID: QR6
TAG: ResponseTime
GIST: The fastness of the search
SCALE: The response time of a search
METER: Measurements obtained from 1000 searches during testing.
MUST: No more than 2 seconds 100% of the time.
WISH: No more than 1 second 100% of the time.

3.3.7 System dependability

ID: QR8
TAG: SystemDependability
GIST: The fault tolerance of the system.
SCALE: If the system loses the connection to the Internet or to the GPS device or the system gets some strange input, the user should be informed.
METER: Measurements obtained from 1000 hours of usage during testing.
MUST: 100% of the time.

3.4 Design constraints
This section includes the design constraints on the software caused by the hardware.

3.4.1 Hard drive space

ID: QR10
TAG: HardDriveSpace
GIST: Hard drive space.
SCALE: The application's need of hard drive space.
METER: MB.
MUST: No more than 20 MB.
PLAN: No more than 15 MB.
WISH: No more than 10 MB.
MB: DEFINED: Megabyte

3.4.2 Application memory usage

ID: QR11
TAG: ApplicationMemoryUsage
GIST: The amount of Operate System memory occupied by the application.
SCALE: MB.
METER: Observations done from the performance log during testing
MUST: No more than 20 MB.
PLAN: No more than 16 MB
WISH: No more than 10 MB
Operate System: DEFINED: The mobile Operate System which the application is running on.
MB: DEFINED: Megabyte.

3.5 Software system attributes
The requirements in this section specify the required reliability, availability, security and maintainability of the software system.

3.5.1 Reliability

ID: QR9
TAG: SystemReliability

GIST: The reliability of the system.

SCALE: The reliability that the system gives the right result on a search.

METER: Measurements obtained from 1000 searches during testing.

MUST: More than 98% of the searches.

PLAN: More than 99% of the searches.

WISH: 100% of the searches.

3.5.2 Availability

ID: QR7

TAG: SystemAvailability

GIST: The availability of the system when it is used.

SCALE: The average system availability (not considering network failing).

METER: Measurements obtained from 1000 hours of usage during testing.

MUST: More than 98% of the time.

PLAN: More than 99% of the time.

WISH: 100% of the time.

ID: QR22

TITLE: Internet Connection

DESC: The application should be connected to the Internet.

RAT: In order for the application to communicate with the database.

DEP: none

ID: QR23

TITLE: GPS Connection

DESC: The application should be connected to the GPS device.

RAT: In order for the application to get the users location, the map and to calculate the distance.

DEP: none

3.5.3 Security

ID: QR12

TAG: CommunicationSecurity

GIST: Security of the communication between the system and server.

SCALE: The messages should be encrypted for log-in communications, so others cannot get user-name and password from those messages.

METER: Attempts to get user-name and password through obtained messages on 1000 log-in session during testing.

MUST: 100% of the Communication Messages in the communication of a log-in session should be encrypted.

Communication Messages: Defined: Every exchanged of information between client and server.

ID: QR13

TAG: RestaurantOwnerLoginAccountSecurity

GIST: Security of accounts.

SCALE: If a restaurant owner tries to log in to the web portal with a non-existing account then the restaurant owner should not be logged in. The restaurant owner should be notified about log-in failure.
METER: 1000 attempts to log-in with a non-existing user account during testing.
MUST: 100% of the time.

ID: QR14
TAG: AdminLoginAccountSecurity
GIST: Security of accounts.
SCALE: If an admin tries to log in to the web portal with a non-existing account then the admin should not be logged in. The admin should be notified about log-in failure.
METER: 1000 attempts to log-in with a non-existing user account during testing.
MUST: 100% of the time.

ID: QR15
TAG: RestaurantOwnerAccountSecurity
GIST: Security of restaurant owners accounts.
SCALE: A restaurant owner and IP address should not be able to log-in for a certain time period after three times of failed log-in attempts.
METER: 1000 attempts to log-in during the lock period after user account has been locked because of failed log-in attempts of three times.
MUST: The locking period should be half an hour, and during that period the log-in function is disabled.

ID: QR16
TAG: AdminAccountSecurity
GIST: Security of admin accounts.
SCALE: An admin and IP address should not be able to log-in to the web portal for a certain time period after three times of failed log-in attempts.
METER: 1000 attempts to log-in during the lock period after user account has been locked because of failed log-in attempts of three times.
MUST: The locking period should be half an hour, and during that period the log-in function is disabled.

ID: QR17
TAG: UserCreateAccountSecurity
GIST: The security of creating account for users of the system.
SCALE: If a user wants to create an account and the desired user name is occupied, the user should be asked to choose a different user name.
METER: Measurements obtained on 1000 hours of usage during testing.
MUST: 100% of the time.

ID: QR18
TAG: RestaurantOwnerCreateAccountSecurity
GIST: The security of creating account for restaurant owners of the system.
SCALE: If a restaurant owner wants to create an account and the desired user name is occupied, the restaurant owner should be asked to choose a different user name.
METER: Measurements obtained on 1000 hours of usage during testing.
MUST: 100% of the time.

3.5.4 Maintainability

ID: QR19
TITLE: Application extendibility
DESC: The application should be easy to extend. The code should be written in a way that it favors implementation of new functions.
RAT: In order for future functions to be implemented easily to the application.
DEP: none

ID: QR21
TITLE: Application testability
DESC: Test environments should be built for the application to allow testing of the applications different functions.
RAT: In order to test the application.
DEP: none

3.5.5 Portability

ID: QR20
TITLE: Application portability
DESC: The application should be portable with iOS and Android.
RAT: The adaptable platform for the application to run on.
DEP: none

4. Prioritization and Release Plan

In order to get a view of how to divide the requirements into different releases and what requirements should be included in which release, a prioritization of the requirements is needed. This section discusses the choice of prioritization methods and gives a suggestion of how the release plan for these requirements could look like.

4.1 Choice of prioritization method

When prioritizing the requirements the ten most important ones were picked out first. This was done with a simple "1 to 10" ranking method, with one being "not important" and ten "very important". Based on the elicitation meetings, and the perceived ideas of what was important to the different stakeholders, a number was set for each requirement. The numbers were then summed up for each requirement and the ten with the highest score were chosen to be prioritized with the cost value approach. The results, which are red-marked, can be seen in Appendix I and as shown, it turned out to be five functional requirements and five quality requirements. These requirements were then prioritized according to the cost value approach and the results can be viewed under Appendix II.

The remaining requirements were prioritized according to the "Five-Way Priority Scheme" as shown in Appendix III. This method was chosen since it gives the different stakeholders the same importance and has an enough wide range for determining which requirement is more important than the other [3]. However, in this prioritization process, the development team was not included as a stakeholder since the different features were not considered to be as important to them as for the other stakeholders.

Other methods for prioritization, such as the hundred-dollar test and the yes-no vote, were also considered. The hundred-dollar test is quite similar to the five-way priority scheme, since it also gives a wide range for ranking the requirements. However, it is more easily misused since someone could save all their money and put them on a requirement that they think is very important [3]. Others might not agree that this requirement is important but it might still get the most votes since one person cared about it [3].

The yes-no vote method might be fairly simple to carry out, however the range is too narrow. For instance, if two requirements are not very important it would be hard to determine which of those requirements that is more important than the other [3].

In conclusion, weighing the disadvantages and advantages of these methods against each other lead us to choose the five-way priority scheme.

4.2 Release Plan

The requirements were divided into three releases based on the prioritization and their dependencies. The three different releases were assembled so that each would work as a fully functional application.

In the first release the requirements that build up the foundation of the application were included, together with the most highly prioritized requirements and their dependencies.

27

The second release also includes important requirements. However, these requirements are not vital for a functional application. They are more suited to act as additional features that can contribute to making the software product more attractive.

The third release includes the requirements that can be afforded to discard if the project gets delayed or overruns the budget.

For further details about the release plan, see Appendix IV.