Software Requirements Specification

Amazing Lunch Indicator

Sarah Geagea 881024-4940
Sheng Zhang 850820-4735
Niclas Sahlin 880314-5658
Faegheh Hasibi 870625-5166
Farhan Hameed 851007-9695
Elmira Rafiyan 840724-5383
Magnus Ekberg 851022-1933

Do not follow this template, only follow the IEEE template for your project!!!

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 "Amazing Lunch Indicator" (ALI) 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 "Amazing Lunch Indicator" is a GPS-based mobile application which helps people to find the closest restaurants based on the user's current position and other specification like price, restaurant type, dish and more. The application should be free to download from either a mobile phone application store or similar services.

Restaurant owners can provide their restaurant information using the web-portal. This information will act as the bases for the search results displayed to the user. An administrator also uses the web-portal in order to administer the system and keep the information accurate. The administrator can, for instance, verify restaurant owners and manage user information.

Furthermore, the software needs both Internet and GPS connection to fetch and display results. All system information is maintained in a database, which is located on a web-server. The software also interacts with the GPS-Navigator software which is required to be an already installed application on the user's mobile phone. By using the GPS-Navigator, users can view desired restaurants on a map and be navigated to them. The application also has the capability of representing both summary and detailed information about the restaurants.

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
Restaurant Owner	Someone who has a restaurant and wants his restaurant to be a part the application
Web-Portal	A web application which present special facilities for restaurant owner

	and admin
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 find restaurants and view information about them while the web portal will be used for managing the information about the restaurants and the system as a whole.

The mobile application will need to communicate to a GPS application within the mobile phone, which in turn communicates with a physical GPS device to find the location of the user, see Figure 1. The GPS will provide the mobile application with locations of both the user and the restaurants and the distance between them, but it will also provide maps and the functionality to display the application's data on the map. The functionality provided by the GPS will be embedded into the application in order for the user to be able to use the functions in the application in a seamlessly manner.

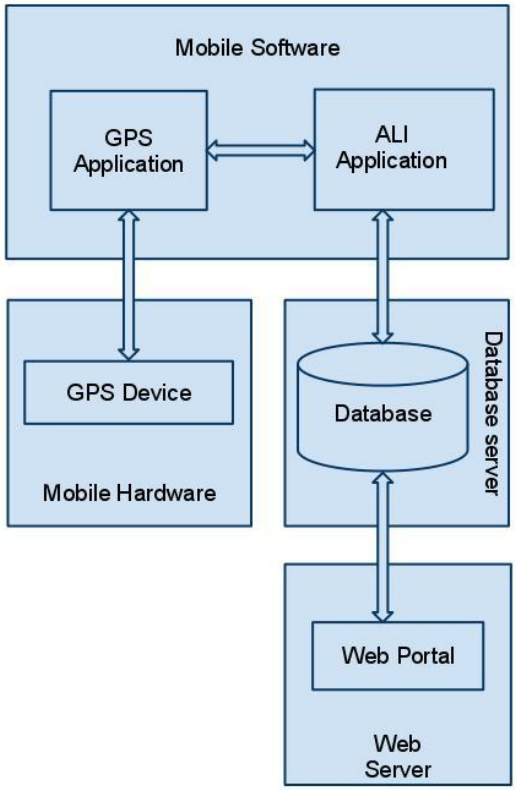

Figure 1 - Block diagram

Since this is a data-centric product it will need somewhere to store the data. For that, a database will be used. Both the mobile application and web portal will communicate with the database, however in slightly different ways. The mobile application will only use the database to get data while the web portal will also add and modify data. All of the database communication will go over the Internet.

The mobile application has some restrictions about the resource allocation. To avoid problems with overloading the operating system the application is only allowed to use 20 megabytes of memory while running the application. The maximum amount of hard drive space is also 20 megabytes.

2.2 Product functions

With the mobile application, the users will be able to search for restaurants. The result will be based on the criteria the user inputs. There are several search criteria and it will be possible for the administrator of the system to manage the options for those criteria that have that.

The result of the search will be viewed either in a list view or in a map view, depending on what criteria included in the search. The list view will have one list item for each restaurant matching the search criteria and show a small part of the restaurant information so the user can identify the restaurant. The

map view will show each restaurant location as a pin on the map as well as the user's own location. In both views the users will be able to either select a restaurant as target destination or get information how to get there, or view the information of a specific restaurant.

The web portal will provide functionality to manage the system and the restaurant information. It will also provide information about the system, for example show when there is a new update.

2.3 User characteristics

There are three types of users that interact with the system: users of the mobile application, restaurant owners and administrators. Each of these three types of users has different use of the system so each of them has their own requirements.

The mobile application users can only use the application to find a restaurant. This means that the user have to be able to search for restaurants, choose a restaurant from that search and then navigate to it. In order for the users to get a relevant search result there are multiple criteria the users can specify and all results matches all of those.

The restaurant owners will not use the mobile application but the web portal instead. There they will manage the information about their restaurant, for example a description of the restaurant, contact information and their menu.

The administrators also only interact with the web portal. They are managing the overall system so there is no incorrect information within it. The administrator can manage the information for each restaurant as well as the options for both the mobile application users and the restaurant owners.

2.4 Constraints

The mobile application is constrained by the system interface to the GPS navigation system within the mobile phone. Since there are multiple system and multiple GPS manufacturers, the interface will most likely not be the same for every one of them. Also, there may be a difference between what navigation features each of them provide.

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, see Appendix IV.

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.

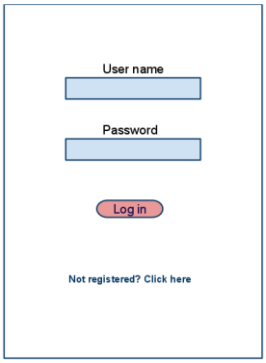

Figure 2 - Login page

Figure 3 – Search page

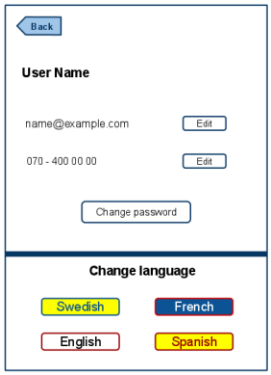

Figure 4 – Profile page

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.

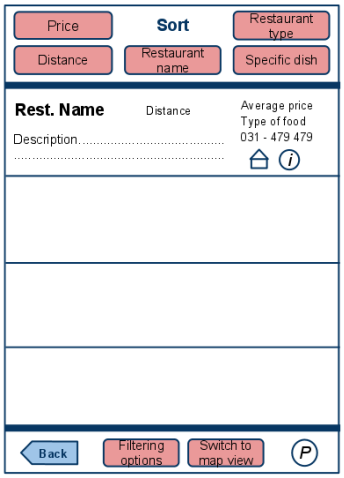

Figure 5 – List view

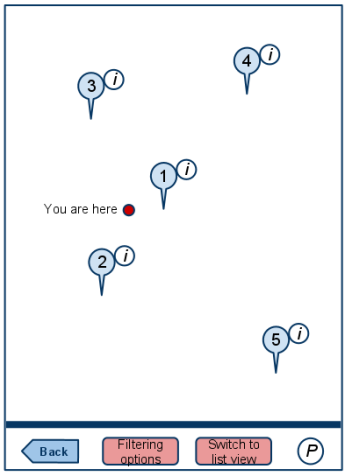

Figure 6 – Map view

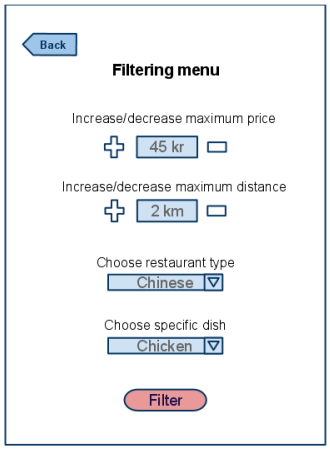

Figure 7 – Filter menu

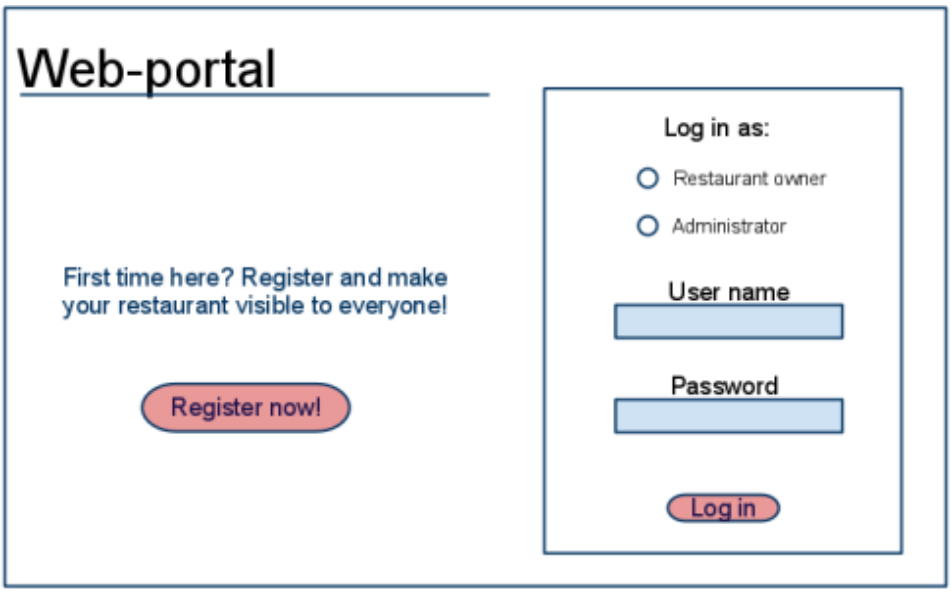

Figure 8 – Web Portal

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

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 user-name, 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

UML diagram (e.g. Use case diagram) should be added!

ID: FR5
TITLE: Retrieve password
DESC: Given that a user has registered, then the user should be able to d 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 - Search
DESC: Given that a user is logged in to the mobile application, then the first page that is shown should be the search page. The user should be able to search for a restaurant, according to several search options. The search options are Price, Destination, Restaurant type and Specific dish. There should also be a free-text search option. A user should be able to select multiple search options in one search.
RAT: In order for a user to search for a restaurant.
DEP: FR4

3.2.1.7 Functional requirement 1.7

ID: FR7
TITLE: Mobile application - Search result in a map view
DESC:

- Search results can be viewed on a map. On the map, the relevant and closest restaurants according to the user's position are shown.
- A specific pin will represent a specific restaurant location. On each pin there should be an information link.
- There should be maximally 100 results displayed. The map view should have a default zoom.
- The map view should include a button that, when selected, should display different filtering options in a filtering menu.

RAT: The way results are displayed in a map.
DEP: FR6

3.2.1.8 Functional requirement 1.8

ID: FR8
TITLE: Mobile application - Search result in a list view
DESC:

- Search results can be viewed in a list. Each element in the list represents a specific restaurant. Each element should include the restaurant name, telephone number, type of food, distance according to the user's position, average price, a short two-line description, a link to the restaurant's web-page and an information link.
- There should be maximally 100 results displayed. If the result contains more restaurants than what can be displayed on the screen at one time, the user should be able to scroll through them.
- When searching by price the restaurants should be sorted according to the following order:
    1. average price
    2. distance
    3. restaurant type
    4. specific dish
- When searching by a search option, other than price, the restaurants should be sorted according to the following order:
    1. distance
    2. average price
    3. restaurant type
    4. specific dish
- The list view should include a header with different selectable sorting options.
- The list view should include a button that, when selected, should display different filtering options in a filtering menu.

RAT: The way results should be displayed in a list.
DEP: FR6

3.2.1.9 Functional requirement 1.9

ID: FR9
TITLE: Mobile application - Navigation to restaurant
DESC: A user should be able to select a pin on a map or an element on a list. When a selection is made, the location of the restaurant should be sent to the mobile phone's GPS-navigation program. The user should then be navigated to the destination. When the destination is reached, a user should be able to go back to the search page on the mobile application.
RAT: To navigate a user to a chosen restaurant.
DEP: FR7, FR8

3.2.1.10 Functional requirement 1.10

ID: FR10
TITLE: Mobile application - Switch result view
DESC: A user should be able to switch between a map view and a list view for all search options.

RAT: In order for a user to switch between result views.
DEP: FR7, FR8

3.2.1.11 Functional requirement 1.11

ID: FR11
TITLE: Mobile application - Selecting the information link
DESC: A user should be able to select the information link, which is included on all result items. The link will direct the user to an information page, which includes a picture of the restaurant, the restaurant name, address, phone number, e-mail address, type of food, average price, restaurant description and a menu with name, description and price of the different dishes.
RAT: In order to show information about the restaurants.
DEP: FR7, FR8

3.2.1.12 Functional requirement 1.12

ID: FR12
TITLE: Mobile application - Search by price
DESC: A user should be able to input a maximum and a minimum price range. The result is displayed in a list view by default.
RAT: In order for a user to search by price.
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.

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.

Appendix I: Selection for Cost-Value Approach

Table 2 – Select of ten most important requirements

Requirement ID	Magnus	Elmira	Faegheh	Niclas	Farhan	Sean	Sarah	Total
FR1	9	7	5	6	6	8	6	47
FR2	3	6	4	3	4	5	3	28
FR3	4	6	8	6	7	7	6	44
FR4	6	5	3	6	7	6	6	39
FR5	6	9	3	5	7	6	5	41
FR6	8	10	10	9	9	10	10	66
FR7	10	8	10	10	8	10	8	64
FR8	10	10	10	8	8	10	8	64
FR9	6	8	5	8	9	7	8	51
FR10	3	6	7	5	6	8	4	39
FR11	3	4	3	6	5	5	5	31
FR12	4	3	7	6	6	7	7	40
FR13	4	6	9	7	7	7	7	47
FR14	4	4	3	6	6	6	5	34
FR15	4	7	7	5	6	6	6	41
FR16	4	7	5	5	6	6	6	39
FR17	4	3	4	2	5	7	3	28
FR18	6	6	3	7	4	4	6	36
FR19	5	4	5	5	4	7	5	35
FR20	5	7	6	5	6	6	5	40
FR21	5	4	4	6	5	4	6	34
FR22	6	8	6	6	8	7	6	47

FR23	8	7	6	5	5	6	6	43
FR24	5	10	9	10	9	5	10	58
FR25	3	5	5	4	4	3	4	28
FR26	8	5	8	6	5	7	6	45
FR27	9	9	8	7	8	6	8	55
FR28	7	7	5	5	7	5	5	41
FR29	7	7	5	5	6	5	5	40
FR30	5	2	6	3	5	4	4	29
FR31	9	10	5	8	6	4	8	50
FR32	8	7	5	8	6	5	8	47
FR33	3	4	5	4	4	3	4	27
QR1	8	7	7	7	7	6	7	49
QR2	6	6	6	7	5	7	7	44
QR3	6	8	5	6	5	7	6	43
QR4	6	8	5	6	7	7	6	45
QR5	6	6	4	5	4	5	5	35
QR6	8	9	5	7	9	10	8	56
QR7	9	8	7	9	7	8	9	57
QR8	6	7	6	8	7	8	7	49
QR9	6	9	9	9	7	7	10	57
QR10	4	4	3	3	4	6	3	27
QR11	4	6	2	3	4	6	3	28
QR12	9	9	7	8	6	8	8	55
QR13	7	5	6	7	4	5	6	40
QR14	8	5	8	9	5	5	7	47

QR15	7	7	7	6	6	7	6	46
QR16	8	9	8	8	6	7	7	53
QR17	6	6	5	5	5	6	5	38
QR18	6	6	5	8	5	6	6	42
QR19	6	8	7	7	7	7	7	49
QR20	7	8	6	6	7	5	6	45
QR21	8	6	4	7	7	7	6	45
QR22	8	9	9	8	8	4	8	54
QR23	7	9	8	7	7	4	7	49

Appendix II: Prioritization Result of 10 selected Requirements Using Cost-Value Approach

Table 3 – 10 most important requirements

Requirement ID	Title	Requirement Type
FR6	Mobile application - Search	Function
FR7	Mobile application - Search result in a map view	Function
FR8	Mobile application - Search result in a list view	Function
FR24	Restaurant owner manages information	Function
FR27	Administrator verifies restaurant owner	Function
QR6	System response time	Quality
QR7	System Availability	Quality
QR9	System Reliability	Quality
QR12	Communication Security	Quality
QR22	Internet Connection	Quality

Table 4 – Value

Value	FR6	FR7	FR8	FR24	FR27	QR6	QR7	QR9	QR12	QR22
FR6	1	5	7	7	1/3	1/5	1/3	1/3	5	7
FR7	1/5	1	3	5	6	1/5	1/3	1/3	1/3	5
FR8	1/7	1/3	1	4	5	1/6	1/4	1/4	1/5	3
FR24	1/7	1/5	1/4	1	1/3	1/5	1/5	1/3	5	4
FR27	3	1/6	1/5	3	1	1/9	1/5	1/5	1/7	2
QR6	5	5	6	5	9	1	3	3	2	8

QR7	3	3	4	5	5	1/3	1	3	1/3	7
QR9	3	3	4	3	5	1/3	1/3	1	1/3	5
QR12	1/5	3	5	1/5	7	1/2	3	3	1	9
QR22	1/7	1/5	1/3	1/4	1/2	1/8	1/7	1/5	1/9	1

Table 5 – Cost

COST	**FR6**	**FR7**	**FR8**	**FR24**	**FR27**	**QR6**	**QR7**	**QR9**	**QR12**	**QR22**
FR6	1	1/5	1/2	3	5	1/7	1/3	1/5	1/3	7
FR7	5	1	1/5	7	3	1/5	1/3	1/5	3	5
FR8	2	5	1	3	5	1/9	1/5	1/6	1/5	7
FR24	1/3	1/7	1/3	1	3	1/3	2	1/5	3	7
FR27	1/5	1/3	1/5	1/3	1	1/7	1/6	1/7	1/6	2
QR6	7	5	9	3	7	1	3	2	3	9
QR7	3	3	5	1/2	6	1/3	1	2	3	7
QR9	5	5	6	5	7	1/2	1/2	1	3	5
QR12	3	1/3	5	1/5	1/3	6	1/3	1/3	1	5
QR22	1/7	1/5	7	7	1/2	1/9	1/7	1/5	1/5	1

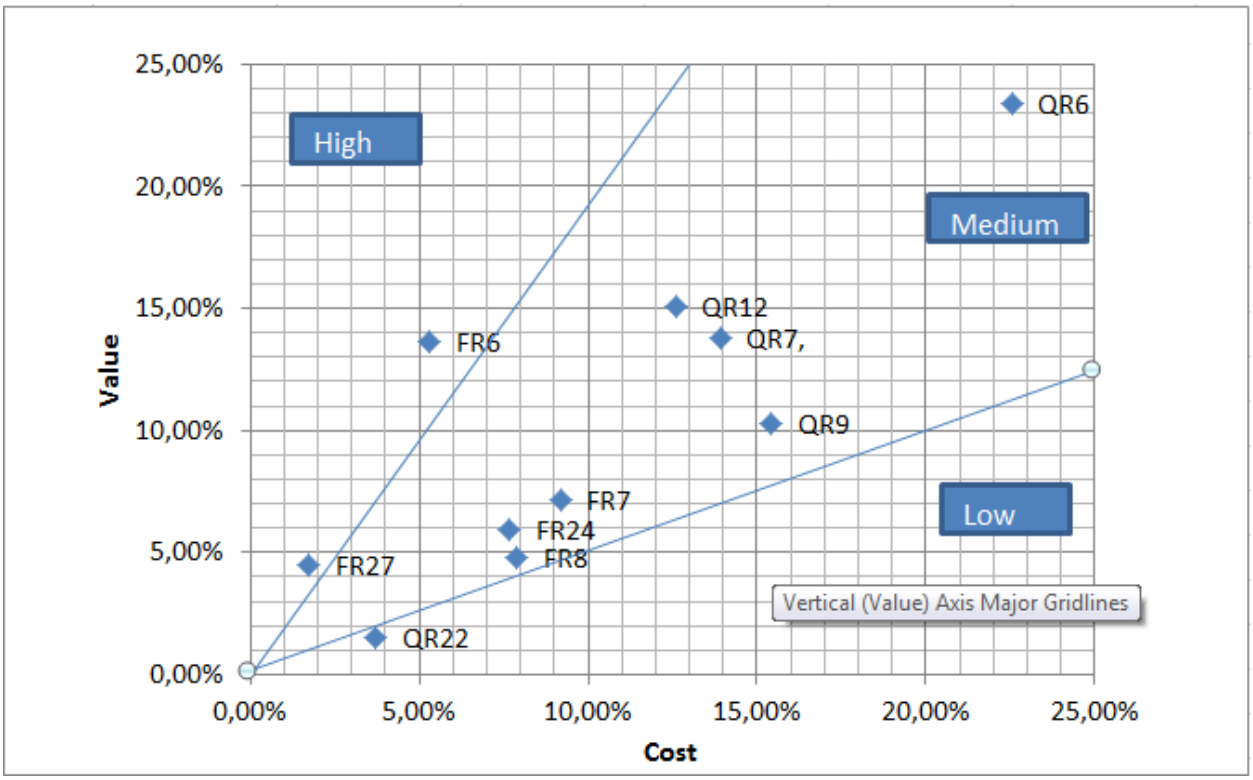

Figure 9 – Result of AHP method

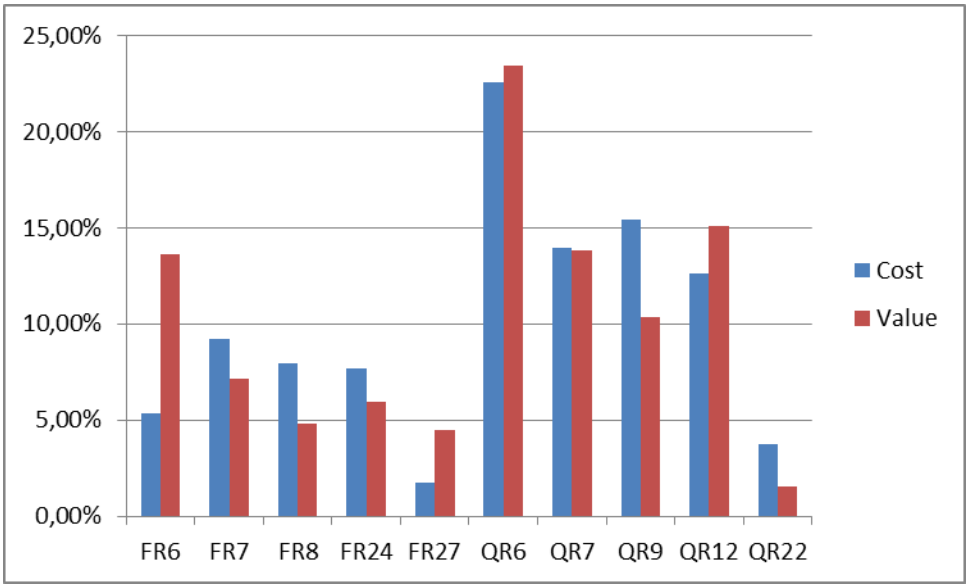

Figure 10 – The value distribution and estimated cost

Table 6 - The value distribution and estimated cost

	FR6	FR7	FR8	FR24	FR27	QR6	QR7	QR9	QR12	QR22
Cost	5,31%	9,18%	7,91%	7,65%	1,73%	22,55%	13,92%	15,43%	12,60%	3,71%
Value	13,60%	7,12%	4,80%	5,93%	4,45%	23,41%	13,78%	10,31%	15,06%	1,53%

Appendix III: Five-Way Priority Scheme

Table 7 – Data of five-way priority scheme prioritization

Requirement ID		Priority by Stakeholder			
		User	Restaurant owner	Administrator	Customer
FR1	Magnus	1	-2	-2	2
	Elmira	1	-1	-2	2
	Niclas	1	-2	-2	1
	Faegheh	1	-2	-2	1
	Sarah	1	-1	-1	1
	Average	**1**	**-1,6**	**-1,8**	**1,4**
	Sum				**-1**
	Rank of Req.		**13**		
		User	Restaurant owner	Administrator	Customer
FR2	Magnus	1	-2	-2	1
	Elmira	1	-1	-2	1
	Niclas	0	-2	-2	1
	Faegheh	-1	-2	-2	0
	Sarah	0	-2	-2	0
	Average	**0,2**	**-1,8**	**-2**	**0,6**
	Sum				**-3**
	Rank of Req.		**34**		
		User	Restaurant owner	Administrator	Customer
FR3	Magnus	1	-2	-2	1
	Elmira	0	-2	-2	1
	Niclas	1	-2	-2	0
	Faegheh	0	-2	-2	1
	Sarah	1	-2	-1	1
	Average	**0,6**	**-2**	**-1,8**	**0,8**
	Sum				**-2,4**
	Rank of Req.		**29**		
		User	Restaurant owner	Administrator	Customer
FR4	Magnus	0	-2	-2	0
	Elmira	0	-2	-2	0
	Niclas	1	-2	-2	0
	Faegheh	0	-2	-2	-1
	Sarah	0	-2	-2	0
	Average	**0,2**	**-2**	**-2**	**-0,2**
	Sum				**-4**
	Rank of Req.		**41**		
		User	Restaurant owner	Administrator	Customer
FR5	Magnus	1	-2	-2	0
	Elmira	1	-2	-2	0
	Niclas	0	-2	-2	-1

	Faegheh	-1	-2	-2	-1
	Sarah	0	-2	-2	-1
	Average	**0,2**	**-2**	**-2**	**-0,6**
	Sum				**-4,4**
	Rank of Req.		**42**		
		User	Restaurant owner	Administrator	Customer
FR9	Magnus	1	1	-2	1
	Elmira	1	0	-2	2
	Niclas	1	0	-2	-1
	Faegheh	1	-2	-2	1
	Sarah	1	0	-2	0
	Average	**1**	**-0,2**	**-2**	**0,6**
	Sum				**-0,6**
	Rank of Req.		**8**		
		User	Restaurant owner	Administrator	Customer
FR10	Magnus	1	-2	-2	1
	Elmira	1	0	-2	0
	Niclas	1	-2	-2	-2
	Faegheh	1	-2	-2	-1
	Sarah	0	-2	-2	0
	Average	**0,8**	**-1,6**	**-2**	**-0,4**
	Sum				**-3,2**
	Rank of Req.		**36**		
		User	Restaurant owner	Administrator	Customer
FR11	Magnus	1	1	-2	1
	Elmira	1	1	-2	1
	Niclas	0	1	-2	-2
	Faegheh	0	1	-2	1
	Sarah	0	1	-2	1
	Average	**0,4**	**1**	**-2**	**0,4**
	Sum				**-0,2**
	Rank of Req.		**5**		
		User	Restaurant owner	Administrator	Customer
FR12	Magnus	1	-1	-2	0
	Elmira	1	0	-2	0
	Niclas	2	-2	-2	-2
	Faegheh	2	-2	-2	0
	Sarah	2	-1	-2	1
	Average	**1,6**	**-1,2**	**-2**	**-0,2**
	Sum				**-1,8**
	Rank of Req.		**18**		
		User	Restaurant owner	Administrator	Customer
FR13	Magnus	1	-1	-2	0
	Elmira	1	0	-2	0

	Niclas	2	-2	-2	-2
	Faegheh	2	-2	-2	1
	Sarah	2	-1	-2	1
	Average	**1,6**	**-1,2**	**-2**	**0**
	Sum				**-1,6**
	Rank of Req.		**17**		
		User	Restaurant owner	Administrator	Customer
FR14	Magnus	1	-1	-2	1
	Elmira	0	-2	-2	-1
	Niclas	1	-2	-2	1
	Faegheh	0	-2	-2	0
	Sarah	0	-2	-2	0
	Average	**0,4**	**-1,8**	**-2**	**0,2**
	Sum				**-3,2**
	Rank of Req.		**37**		
		User	Restaurant owner	Administrator	Customer
FR15	Magnus	1	-1	-2	1
	Elmira	1	0	-2	0
	Niclas	1	-2	-2	-2
	Faegheh	1	-2	-2	0
	Sarah	1	-1	-2	1
	Average	**1**	**-1,2**	**-2**	**0**
	Sum				**-2,2**
	Rank of Req.		**24**		
		User	Restaurant owner	Administrator	Customer
FR16	Magnus	1	-1	-2	1
	Elmira	1	0	-2	0
	Niclas	1	-2	-2	-2
	Faegheh	1	-2	-2	-1
	Sarah	1	-2	-2	1
	Average	**1**	**-1,4**	**-2**	**-0,2**
	Sum				**-2,6**
	Rank of Req.		**31**		
		User	Restaurant owner	Administrator	Customer
FR17	Magnus	1	-1	-2	1
	Elmira	1	0	-2	-1
	Niclas	1	-2	-2	-2
	Faegheh	0	-2	-2	-1
	Sarah	1	-1	-2	0
	Average	**0,8**	**-1,2**	**-2**	**-0,6**
	Sum				**-3**
	Rank of Req.		**35**		
		User	Restaurant owner	Administrator	Customer
FR18	Magnus	2	-2	-2	1

	Elmira	1	-1	-2	1
	Niclas	2	-2	-2	0
	Faegheh	1	-2	-2	-1
	Sarah	1	-2	-2	0
	Average	**1,4**	**-1,8**	**-2**	**0,2**
	Sum				**-2,2**
	Rank of Req.		**25**		
		User	Restaurant owner	Administrator	Customer
FR19	Magnus	1	-2	-2	1
	Elmira	1	-2	-2	0
	Niclas	1	-2	-2	-2
	Faegheh	1	-2	-2	-1
	Sarah	1	-2	-2	0
	Average	**1**	**-2**	**-2**	**-0,4**
	Sum				**-3,4**
	Rank of Req.		**38**		
		User	Restaurant owner	Administrator	Customer
FR20	Magnus	1	-2	-2	1
	Elmira	2	-1	-2	1
	Niclas	1	-2	-2	-2
	Faegheh	1	-2	-2	0
	Sarah	1	-2	-2	0
	Average	**1,2**	**-1,8**	**-2**	**0**
	Sum				**-2,6**
	Rank of Req.		**32**		
		User	Restaurant owner	Administrator	Customer
FR21	Magnus	0	-2	-2	0
	Elmira	1	-2	-2	0
	Niclas	0	-2	-2	-1
	Faegheh	-1	-2	-2	-1
	Sarah	0	-2	-2	0
	Average	**0**	**-2**	**-2**	**-0,4**
	Sum				**-4,4**
	Rank of Req.		**43**		
		User	Restaurant owner	Administrator	Customer
FR22	Magnus	-2	2	-2	2
	Elmira	-2	2	-2	2
	Niclas	-2	2	-2	1
	Faegheh	-2	2	-2	2
	Sarah	-2	2	-2	1
	Average	**-2**	**2**	**-2**	**1,6**
	Sum				**-0,4**
	Rank of Req.		**6**		
		User	Restaurant owner	Administrator	Customer

FR23	Magnus	-2	1	-2	2
	Elmira	-2	2	-2	1
	Niclas	-2	2	-2	1
	Faegheh	-2	2	-2	1
	Sarah	-2	2	-2	1
	Average	**-2**	**1,8**	**-2**	**1,2**
	Sum				**-1**
	Rank of Req.		**14**		
		User	Restaurant owner	Administrator	Customer
FR25	Magnus	-2	1	-2	1
	Elmira	-2	1	-2	0
	Niclas	-2	1	-2	-2
	Faegheh	-2	0	-2	-1
	Sarah	-1	1	-2	0
	Average	**-1,8**	**0,8**	**-2**	**-0,4**
	Sum				**-3,4**
	Rank of Req.		**39**		
		User	Restaurant owner	Administrator	Customer
FR26	Magnus	-2	-2	2	1
	Elmira	-2	-2	1	0
	Niclas	-2	-2	2	1
	Faegheh	-2	-2	2	1
	Sarah	-2	-2	2	1
	Average	**-2**	**-2**	**1,8**	**0,8**
	Sum				**-1,4**
	Rank of Req.		**16**		
		User	Restaurant owner	Administrator	Customer
FR28	Magnus	-2	0	1	1
	Elmira	-2	1	0	1
	Niclas	-1	1	2	0
	Faegheh	-2	-1	2	1
	Sarah	-1	0	1	1
	Average	**-1,6**	**0,2**	**1,2**	**0,8**
	Sum				**0,6**
	Rank of Req.		**2**		
		User	Restaurant owner	Administrator	Customer
FR29	Magnus	-2	0	1	1
	Elmira	-2	1	0	1
	Niclas	-1	1	2	0
	Faegheh	-2	-1	2	1
	Sarah	-1	0	1	1
	Average	**-1,6**	**0,2**	**1,2**	**0,8**
	Sum				**0,6**
	Rank of Req.		**3**		

		User	Restaurant owner	Administrator	Customer
FR30	Magnus	-2	0	1	1
	Elmira	-1	0	1	1
	Niclas	-2	0	2	0
	Faegheh	-2	1	2	1
	Sarah	-1	1	2	1
	Average	**-1,6**	**0,4**	**1,6**	**0,8**
	Sum				**1,2**
	Rank of Req.		**1**		
		User	Restaurant owner	Administrator	Customer
FR31	Magnus	0	-2	1	1
	Elmira	0	-2	0	1
	Niclas	0	-2	2	0
	Faegheh	0	-2	1	0
	Sarah	0	-2	1	1
	Average	**0**	**-2**	**1**	**0,6**
	Sum				**-0,4**
	Rank of Req.		**7**		
		User	Restaurant owner	Administrator	Customer
FR32	Magnus	-2	0	1	1
	Elmira	-2	0	0	1
	Niclas	-2	0	2	0
	Faegheh	-2	1	1	1
	Sarah	-2	0	1	1
	Average	**-2**	**0,2**	**1**	**0,8**
	Sum				**0**
	Rank of Req.		**4**		
		User	Restaurant owner	Administrator	Customer
FR33	Magnus	-2	-2	0	0
	Elmira	-2	-2	0	0
	Niclas	-2	-2	0	-2
	Faegheh	-2	-2	0	-1
	Sarah	-2	-2	0	0
	Average	**-2**	**-2**	**0**	**-0,6**
	Sum				**-4,6**
	Rank of Req.		**44**		
		User	Restaurant owner	Administrator	Customer
QR1	Magnus	2	-2	-2	2
	Elmira	2	-2	-2	2
	Niclas	2	-2	-2	0
	Faegheh	2	-2	-2	1
	Sarah	2	-2	-2	1
	Average	**2**	**-2**	**-2**	**1,2**
	Sum				**-0,8**
	Rank of		**10**		

	Req.				
QR2		User	Restaurant owner	Administrator	Customer
	Magnus	2	-2	-2	2
	Elmira	2	-2	-2	2
	Niclas	2	-2	-2	0
	Faegheh	2	-2	-2	1
	Sarah	2	-2	-2	1
	Average	**2**	**-2**	**-2**	**1,2**
	Sum				**-0,8**
	Rank of Req.	**11**			
QR3		User	Restaurant owner	Administrator	Customer
	Magnus	1	-2	-2	1
	Elmira	1	-1	-2	1
	Niclas	1	-1	-2	0
	Faegheh	0	-1	-2	-1
	Sarah	1	-1	-2	1
	Average	**0,8**	**-1,2**	**-2**	**0,4**
	Sum				**-2**
	Rank of Req.	**21**			
QR4		User	Restaurant owner	Administrator	Customer
	Magnus	1	-2	-2	1
	Elmira	1	-1	-2	1
	Niclas	1	-1	-2	0
	Faegheh	1	-1	-2	0
	Sarah	1	-1	-2	0
	Average	**1**	**-1,2**	**-2**	**0,4**
	Sum				**-1,8**
	Rank of Req.	**19**			
QR5		User	Restaurant owner	Administrator	Customer
	Magnus	1	-1	-2	1
	Elmira	1	-1	-2	1
	Niclas	1	0	-2	-2
	Faegheh	1	1	-2	-1
	Sarah	0	-1	-2	0
	Average	**0,8**	**-0,4**	**-2**	**-0,2**
	Sum				**-1,8**
	Rank of Req.	**20**			
QR8		User	Restaurant owner	Administrator	Customer
	Magnus	1	-2	-2	1
	Elmira	1	-2	-2	0
	Niclas	1	-2	-2	2
	Faegheh	1	-2	-2	0
	Sarah	0	-1	-1	0
	Average	**0,8**	**-1,8**	**-1,8**	**0,6**
	Sum				**-2,2**

	Rank of Req.	**26**			
QR10		User	Restaurant owner	Administrator	Customer
	Magnus	-1	-2	-2	-1
	Elmira	-1	-2	-2	0
	Niclas	-1	-2	-2	-1
	Faegheh	-1	-2	-2	-1
	Sarah	-1	-2	-2	0
	Average	**-1**	**-2**	**-2**	**-0,6**
	Sum				**-5,6**
	Rank of Req.	**45**			
QR11		User	Restaurant owner	Administrator	Customer
	Magnus	-1	-2	-2	-1
	Elmira	0	-2	-2	0
	Niclas	-1	-2	-2	-1
	Faegheh	-1	-2	-2	-1
	Sarah	-1		-2	-1
	Average	**-0,8**	**-2**	**-2**	**-0,8**
	Sum				**-5,6**
	Rank of Req.	**46**			
QR13		User	Restaurant owner	Administrator	Customer
	Magnus	-2	2	-1	2
	Elmira	-2	1	-2	1
	Niclas	-2	2	-2	2
	Faegheh	-2	2	-2	1
	Sarah	-2	1	-2	
	Average	**-2**	**1,6**	**-1,8**	**1,5**
	Sum				**-0,7**
	Rank of Req.	**9**			
QR14		User	Restaurant owner	Administrator	Customer
	Magnus	-2	-2	2	2
	Elmira	-2	-2	1	1
	Niclas	-2	-2	2	2
	Faegheh	-2	-2	2	2
	Sarah	-2	-2	1	1
	Average	**-2**	**-2**	**1,6**	**1,6**
	Sum				**-0,8**
	Rank of Req.	**12**			
QR15		User	Restaurant owner	Administrator	Customer
	Magnus	-2	0	-2	-1
	Elmira	-2	0	-2	2
	Niclas	-2	1	-2	2
	Faegheh	-2	0	-2	2
	Sarah	-2	1	-2	1
	Average	**-2**	**0,4**	**-2**	**1,2**

	Sum				**-2,4**
	Rank of Req.			**30**	
QR16		User	Restaurant owner	Administrator	Customer
	Magnus	-2	-2	0	-1
	Elmira	-2	-1	0	2
	Niclas	-2	-2	1	2
	Faegheh	-2	-2	1	0
	Sarah	-2	-2	0	0
	Average	**-2**	**-1,8**	**0,4**	**0,6**
	Sum				**-2,8**
	Rank of Req.			**33**	
QR17		User	Restaurant owner	Administrator	Customer
	Magnus	2	-2	-2	0
	Elmira	1	-2	-2	2
	Niclas	1	-2	-2	2
	Faegheh	0	-2	-2	1
	Sarah	0	-2	-2	1
	Average	**0,8**	**-2**	**-2**	**1,2**
	Sum				**-2**
	Rank of Req.			**22**	
QR18		User	Restaurant owner	Administrator	Customer
	Magnus	-2	2	-2	0
	Elmira	-2	1	-2	2
	Niclas	-2	1	-2	2
	Faegheh	-2	0	-2	1
	Sarah	-2	0	-2	1
	Average	**-2**	**0,8**	**-2**	**1,2**
	Sum				**-2**
	Rank of Req.			**23**	
QR19		User	Restaurant owner	Administrator	Customer
	Magnus	-2	-2	-2	2
	Elmira	-1	-1	-2	2
	Niclas	-2	-2	-2	1
	Faegheh	-2	-2	-2	2
	Sarah	-1	-1	-2	1
	Average	**-1,6**	**-1,6**	**-2**	**1,6**
	Sum				**-3,6**
	Rank of Req.			**40**	
QR20		User	Restaurant owner	Administrator	Customer
	Magnus	0	-2	-2	2
	Elmira	0	-2	-2	2
	Niclas	1	-2	-2	1
	Faegheh	1	-2	-2	1
	Sarah	-1	-1	-2	1

	Average	**0,2**	**-1,8**	**-2**	**1,4**
	Sum				**-2,2**
	Rank of Req.		**27**		
QR21		User	Restaurant owner	Administrator	Customer
	Magnus	-2	-2	-2	2
	Elmira	-1	-1	-1	2
	Niclas	-2	-2	-2	2
	Faegheh	1	-2	-2	2
	Sarah	-1	-1	-1	2
	Average	**-1**	**-1,6**	**-1,6**	**2**
	Sum				**-2,2**
	Rank of Req.		**28**		
QR23		User	Restaurant owner	Administrator	Customer
	Magnus	1	-2	-2	1
	Elmira	1	0	-2	2
	Niclas	1	-2	-2	1
	Faegheh	1	-2	-2	1
	Sarah	1	-1	-2	1
	Average	**1**	**-1,4**	**-2**	**1,2**
	Sum				**-1,2**
	Rank of Req.		**15**		

Table 8 – Result of five-way priority scheme prioritization

Priority	Requirement ID
11	FR30
12	FR28
13	FR29
14	FR32
15	FR11
16	FR22
17	FR31
18	FR9
19	QR13
20	QR1
21	QR2
22	QR14
23	FR1
24	FR23
25	QR23
26	FR26
27	FR13
28	FR12
29	QR4
30	QR5

31	QR3
32	QR17
33	QR18
34	FR15
35	FR18
36	QR8
37	QR20
38	QR21
39	FR3
40	QR15
41	FR16
42	FR20
43	QR16
44	FR2
45	FR17
46	FR10
47	FR14
48	FR19
49	FR25
50	QR19
51	FR4
52	FR5
53	FR21
54	FR33
55	QR10
56	QR11

Appendix IV: Release Plan

Table 9 – Release plan

RE:	Dependencies	Description	Motivation	Release	Duration
FR1	-	Download mobile application	This requirement makes the application available for users and is therefore an important requirement to include in the first release.	1	80
FR2	FR1	Download and notify users of new releases	The download of the new versions is important for users to be able to receive the future release of the application and will therefore be included in the first release.	1	2
FR3	-	User registration	For the user to be able to use the application, the user has to registrant. Consequently, this requirement needs to be met in the first release.	1	4
FR4	FR1, FR3	User Log-in	For the user to be able to use the application, the user has to log in. Consequently, this requirement needs to be met in the first release.	1	2
FR5	FR1	Retrieve Password	For the user to be able to receive a forgotten password will have to wait to the second release. This is not vital for the application and was therefore not included in the first release.	2	1
FR6	FR4	Search	The search feature is one of the most important and vital part of the system. It's part of the basic goal of the program and should therefore be included in the first release.	1	2
FR7	FR6, QR22, QR23	Search result in a map view	The ability to show the search result in a map view is part of the basic goal of the program and should therefore be included in the first release.	1	1
FR8	FR6, QR22	Search result in a list view	The ability to show the search result in a list view is part of the basic goal	2	2

			of the program. We have decided to put this one in the second release and only include the map result view in the first release.		
FR9	FR7, FR8	Navigation to restaurant	To make the first release interesting and useful for the users, this is included in the first release.	1	2
FR10	FR7, FR8	Switch result view	The switch between the result views will be implemented after the result list is implemented. This is not in some way vital for the application and can be discarded if the project gets delayed or overruns the budget.	3	1
FR11	FR7, FR8	Selecting the information link	This will be added as an additional feature in the second release to make the application more attractive for the users.	2	1
FR12	FR8	Search by price	This will be added as an additional feature in the third release to make the application more attractive for the users.	2	1
FR13	FR7	Search by destination	This is the search function that will be included in the first release.	1	1
FR14	FR12, FR13	Accepted input for price and destination search	This is a requirement that will make the application more stable. It will be added in the third release.	3	1
FR15	FR7	Search by restaurant type	This will be added as an additional feature to the second release to make the application more attractive for the users.	3	3
FR16	FR7	Search by specific dish	This will be added as an additional feature in the third release to make the application more attractive for the users.	3	3
FR17	FR6	Free-text search	This will be added as an additional feature to the first release to make the application more attractive for the	2	3

			users.		
FR18	FR5	No search match found	This is a requirement that will make the application more stable and make the user experience better. It is not vital and will be added in the third release.	3	1
FR19	FR8	Sorting results	This will be added as an additional feature to the first release to make the application more attractive for the users.	2	2
FR20	FR7, FR8	Filtering results	This will be added as an additional feature to the first release to make the application more attractive for the users.	2	4
FR21	FR1	Profile page	This will be added as an additional feature in the third release to make the application more attractive for the users.	3	1
FR22	-	Create account - restaurant owner	The restaurant owner is a vital part of the system and must be included in the first release.	1	2
FR23	FR27	Log-in - restaurant owner	This should be included in the first release because the restaurant owner needs to be able to log in so hi/she can manage the restaurant information.	1	2
FR24	FR23, FR25	Manage info - restaurant owner	The ability for the restaurant owner to manage the information about the restaurant is very important, if there is no information there is nothing to search for.	1	2
FR25	-	Selecting preferred language on the web-portal - restaurant owner	This will be added as an additional feature in the third release to make the application more attractive for the users. It is not vital for the application and will therefore be implemented in the third release.	3	3

FR26	-	Log-in - admin	This should be included in the first release because the administrator needs to be able to log in so she/hi can manage the system.	1	2
FR27	FR26, FR22	Verify restaurant owner	It's important for the admin to be able to verify the restaurant owner's application for registration.	1	2
FR28	FR26	Manage rest. types - administrator	The type search will be added in the second release.	2	1
FR29	FR26	Manage rest. dished - administrator	The dish search will be added in the second release.	2	1
FR30	FR26	Manage rest. info - administrator	The info link will be added in the second release.	2	1
FR31	FR26	Manage users - administrator	The administrator needs to be able to handle users during the first release.	1	1
FR32	FR26	Selecting preferred language on the web-portal - administrator	The administrator needs to be able to handle restaurant owners during the first release.	1	1
FR33	FR26	Language selection on portal	This will be added as an additional feature in the third release to make the application more attractive for the users. It is not vital for the application and will therefore be implemented in the third release.	3	2
QR1	-	Prominent search feature	This is a very high-prioritized requirement. Consequently, this will be included in the first release.	1	1
QR2	-	The ease of usage of the search feature	This is a very high-prioritized requirement. Consequently, this will be included in the first release.	1	1

QR3	-	The ease of usage of the result in the list view	This is a high-prioritized requirement. But we have decided to put this in the second release in favor for more vital and higher prioritized requirements.	2	1
QR4	-	The ease of usage of the result in map view	This is a high-prioritized requirement. But we have decided to put this in the second release in favor for more vital and higher prioritized requirements.	2	1
QR5	-	The ease of usage of the information link	This will be considered when all the mandatory requirements for the system have been implemented.	3	1
QR6	-	System response time	This will be considered and continuously improved during the whole process. We can only discharge the requirement when all functions are implemented.	3	*
QR7	-	System availability	This will be considered and continuously improved during the whole process.	1	*
QR8	QR22, QR23, FR14, FR15, FR16, FR17	System dependability	This will be considered and continuously improved during the whole process. We can only discharge the requirement when all functions are implemented.	3	*
QR9	-	System reliability	This will be considered and continuously improved during the whole process.	1	*
QR10	-	Application hard drive space usage	This will be considered and continuously improved during the whole process. We can only discharge the requirement when all functions are implemented.	3	*
QR11	-	Application memory usage	This will be considered and continuously improved during the whole process. We can only	3	*

			discharge the requirement when all functions are implemented.		
QR12	-	Communication security	As the system grows, and with respect for the users this need to be included.	2	4
QR13	FR22	Non-existing account security for rest. Owners	This need to be included in the first release to enhance the safety of the system.	1	3
QR14	-	Non-existing account security for the administrators	This needs to be included in the first release to enhance the safety of the system.	1	3
QR15	FR23	Log in security for rest. Owners	This needs to be included in the first release to prevent illegitimate attempts to use the restaurants owners account.	1	3
QR16	-	Log in security for administrators	This need to be included in the first release to prevent illegitimate attempts to use the administrator account.	1	2
QR17	FR3	User account creation security	This needs to be included in the first release to resolve conflict between users with the same name.	1	2
QR18	FR22	Restaurant owner account creation security	This needs to be included in the first release to resolve conflict between restaurant owners with the same name.	1	2
QR19	-	Application extendibility	This will be considered and continuously improved during development.	1	*
QR20	-	Application portability	This will be considered and continuously improved during the whole process.	1	50
QR21	-	Application testability	The test environment for the system will continuously be built as the	1	30

			system expands.		
QR22	-	Internet connection	Internet connection is mandatory for the application to work and is therefore included in the first release.	1	*
QR23	-	GPS connection	GPS connection is mandatory for the application to be able to show the result and is therefore included in the first release.	1	2

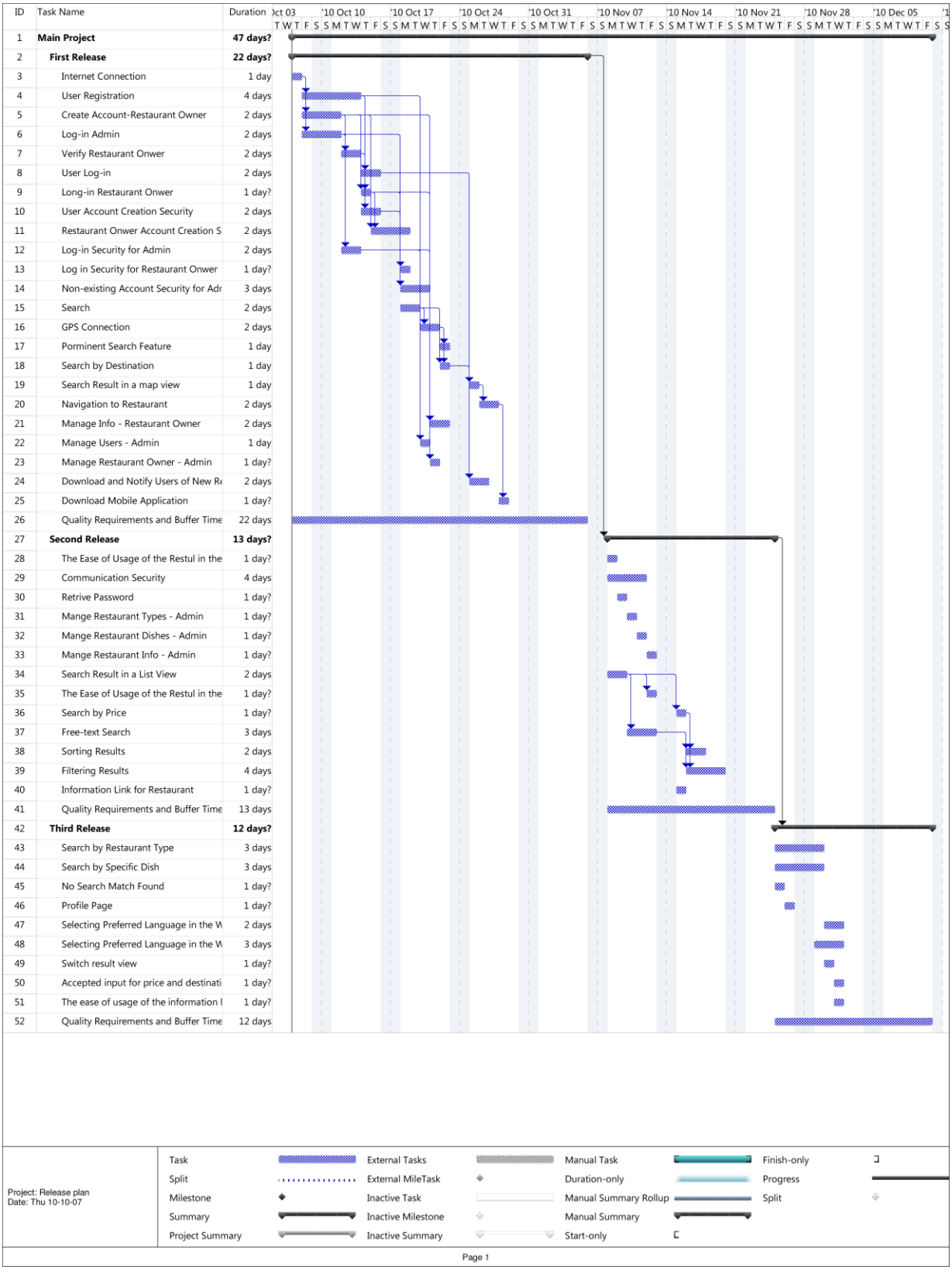

Figure 11 – Release plan schedule

Appendix V: I-star

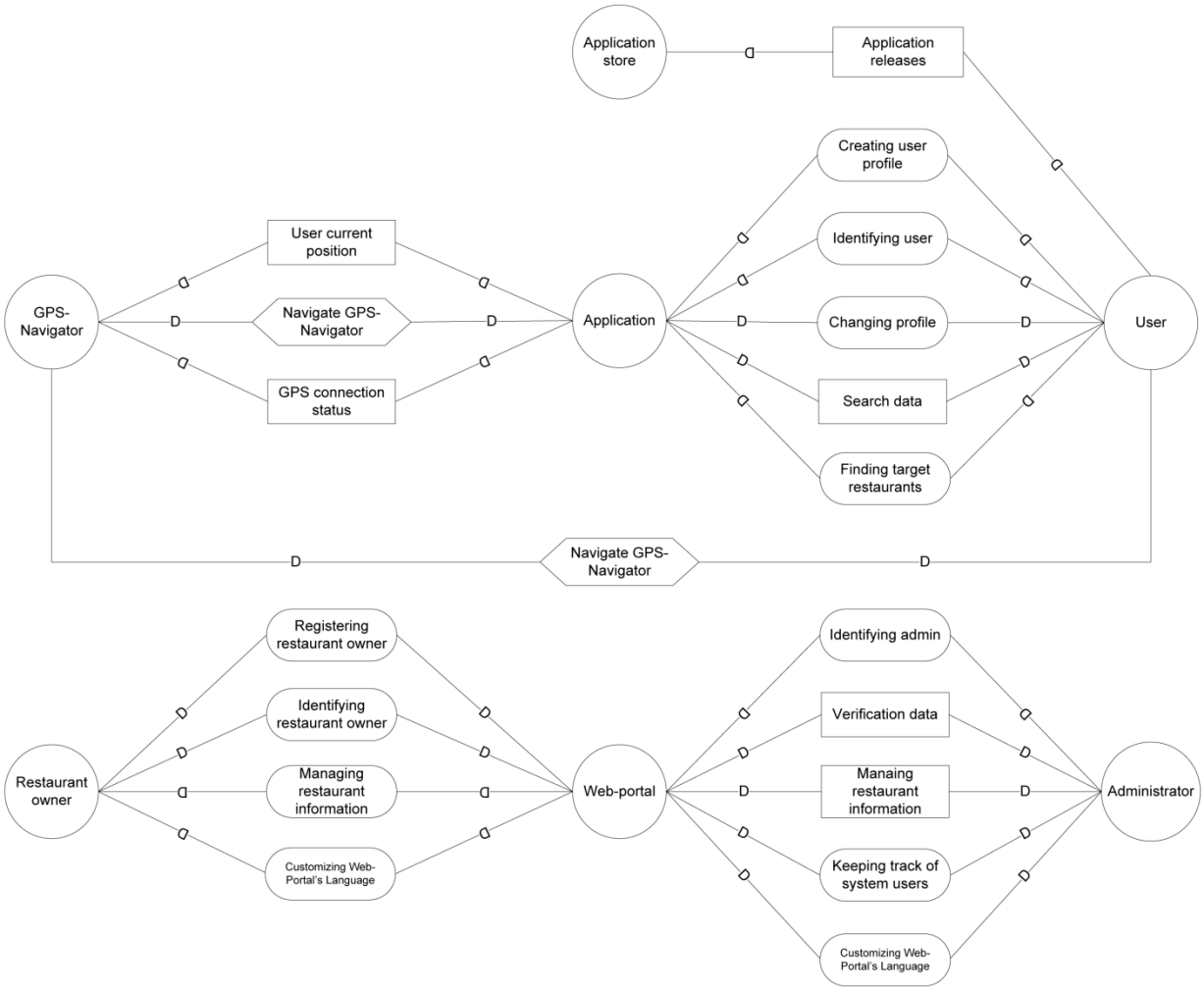

Figure 12 – SD diagram

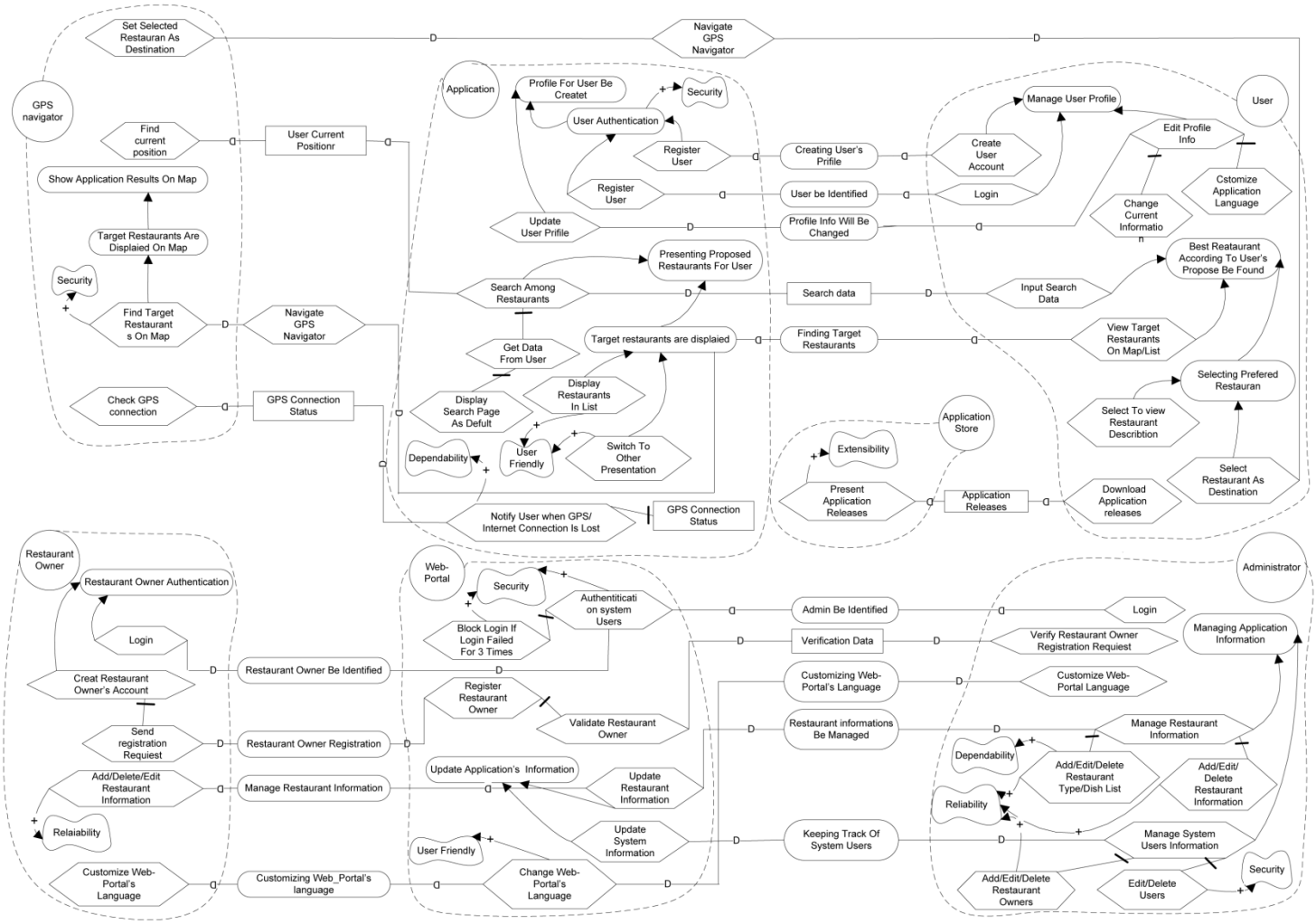

Figure 13 – SR diagram