Venycia Supermarket

Course Name: Software Requirements
Supervisor name: Bilal Aljabour
Student names: Sokena EL-Fellah
Date: fall 2017/2018

Table of Contents

1.Introduction

This section describes the scope and overview of everything included in this document.

1.1 Purpose:

The purpose of this document is to give a detailed description of the requirements for Venicya supermarket. This document contains the functional, behavioral and non-functional requirements of the project. Venicya supermarket is one of the biggest supermarket franchise in Benghazi, Libya. It opened in October 2010. It provides grocery and general merchandise.

1.2 Identifying the problem:

- The supermarket wants to increase customers, rather than having their customers physically come to the shop, they want another way to increase their customer rate.
- Want to create a website and a mobile application.
- A way to enter all their products into the system in a faster and organized way.
- Customers to search for products.
- Automatically update product stock, when product has been bought.
- For customers to login
- For employee/ manager to update website / application with ease.

1.3 Solution:

- Create a web portal /database for the products creating one for each department, with the information of each product;
    1. ***1.*** Product Name
    2. ***2.*** Product Picture
    3. ***3.*** Product description
- Create mobile application, in which customer will buy from an app store or other services, in which each customer will create an account;
    1. Name
    2. Address
    3. Phone number
    4. Email

1.4 Scope

Venicya supermarket is a website and a mobile application which helps customers order products from their homes.

Main actors of this system:

- Customer
- Administrator/ manager

Main use cases associated:

1. Administrator/ Manager can:
    - Enter new department
    - Enter new product
    - Enter product information
        - Product name
        - Product description
        - Product price
        - Product picture

    - Manage stock
    - Edit user information

2. Customers can:
    - Create account- enter information
        - Name
        - Address
        - Telephone
        - Email address
    - Search for products
    - Choose product
    - Place item in bag/ shopping cart
    - Place order

1.5 Definitions, acronyms, and abbreviations:

User: Someone who interacts with the mobile phone application

Administrator: Is someone who is given specific permission for managing and controlling the system

Web-portal: A web application which present special facilities for the administrator.

Stakeholder: Any person who is involved with the system who is not a developer.

DESC: Description

RAT: Rational

DEP: Dependency

1.6. References:

1. Use cases
2. http://aakashtechsupportdocs.readthedocs.io/en/latest/nonfunc.html
3. http://users.csc.calpoly.edu/~jdalbey/SWE/QA/nonfunctional.html
4. https://gephi.org/users/gephi_srs_document.pdf
5. https://www.slideshare.net/PasinduTennage/sample-software-engineering-feasibility-study-report
6. http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs_example_2010_group2.pdf

2. Feasibility study

After much research and analysis, the team believes that this system is feasible. Through discussion and determination of the functionality and requirements of the essential subsystems, preliminary system designs were made and project costs were estimated. In reference to the given time, resource, and cost restraints, the project has been scaled to an appropriate level in which deliverables are reasonable

2.1 Financial Feasibility:

Being a web application Venicya supermarket will have an associated accommodating cost. Since the system doesn't consist of any multimedia data transfer, bandwidth required for the operation of this application is very low.

The project is financially feasible.

2.2 Technical Feasibility:

The project is a complete web based application. The main technologies and tools that are associated with it are:

- HTML
- CSS
- MySQL
  Each of these techs are freely available and the technical skills required are manageable. Initially the web site will be hosted in a free web hosting space, but for later implementation it will be hosted in a paid web hosting space with a sufficient bandwidth. The project is technically feasible.

2.3 Risk Feasibility:

Risk feasibility can be discussed under several perspectives:

Risk associated with size:

Estimated size of the product with many numbers of stakeholders, it will contain substantial amount of code lines. As the system doesn't contain any multimedia aspects, the file sizes and the finished project size will not exceed a certain amount of MB.

Size of the database created or used by the product:

Database size will not surpass the values given by MySQL. Number of relations and entities are minimized by using best practices of normalization theories.

Business impact risks:

Effect of this product on company revenue:

This project can be implemented either as an individual system, or could not go well with the people of Libya, because is a new type of way of shopping, there is a high risk this might not work as well as expected.

Reasonableness of delivery deadlines:

Being a 5 month project, the project will have several deadlines and deliverables that are arranged successively. Depending on the design and coding cost and effort, the deadlines are quite reasonable.

2.4 Social/ legal Feasibility:

This project will be socially accepted because the whole home delivery feature, has become a very successful business in Libya, the lack of transportation in the country makes it more difficult for the people to go out. This product will help a lot of customers who cannot physically come to the supermarket. This will also increase workers because the supermarket will need people to deliver the orders.

The feasibility study has shown that the project is likely to succeed within its constraints, and this helps us start the requirements analysis and proceed with the project.

3. Overall description

This section will give an overview of the whole system. The system will be explained in its setting to show how the system interacts with other systems and introduce the simple 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.

3.1 Product perspective

This system will consist of two parts: one mobile application and one web portal. The mobile application will be used for customers to buy the supermarkets products and view information about them while the web portal will be used for managing the information about each department, product and the system as a whole. The mobile application will need to be connected to the internet within the mobile phone.

The functionality provided in the website will be embedded into the application in order for the user to be able to use the functions in the application in a flawlessly manner. 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 converse 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 sharing. To escape 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.

3.2 Product functions

With the mobile application, the users will be able to search for certain products. 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 in a list view, depending on what criteria inputted in the search. The list view will have one list item for each product matching the search criteria and show a small part of the products information so the user can identify the product they prefer. The users will be able to either select a certain product as their preferred merchandise or get information about it and what it entails. The web portal will provide functionality to manage the system and the product information. It will also provide information about the system, for example show when there is a new update.

3.3 User characteristics

There are two types of users that interact with the system: users of the mobile application and the administrators. Each of these two types of users have different use of the system so each of them has their own requirements. The mobile application users can only use the application to find the products wanted and to order/purchases their cart. This means that the user has to be able to search for products, choose a selected item from that search and then purchase order. 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 administrators will not use the mobile application but the web portal instead. There they will manage the information about their departments and products, for example a description of an item, contact information and their prices They are managing the overall system so there is no incorrect information within it. The administrator can manage the options for the mobile application users.

3.4 Constraints

The Internet connection is a constraint for the application. Since the application brings 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 applications it may be forced to queue incoming requests and therefor increase the time it takes to gather data.

3.5 Assumptions and dependencies

One assumption about the product is that it will always be used on mobile phones that have enough presentation. If the phone does not have enough memory available for the application, for example the users might have download them with other applications, there may be circumstances where the application does not work as intended or even at all.

4.Elicitation

Elicitation is about collecting the requirements from stakeholders, there are many different types of elicitation, in this process I interviewed the customer, because he is both a user and the customer.

Interview:

Below you will see examples of some of the question that has been asked to the stakeholder:

Q1: Can you please explain to me what your franchise does/is?

A1: We own Venycia Supermarket, we opened it in 2010. Venycia is grocery and general merchandise retailer, in Benghazi we have two open stores, Hadayak and Beloon. We consider ourselves the biggest supermarket franchise in Libya.

Q2: Can you please explain what exactly it is you want?

A2: I want an application and web page, that will allow customers to shop from the comfort of their homes.

Q3: Why exactly do you want this application; do you think this will help with your review?

A3: Yes, we decided to create an application because we want to increase our customer base without opening up another shop. This way we think we can increase our review without spending a big quantity of money.

Q4: What exactly do you want the application to do?

A4: The application will have different departments and each department has its products, which will have a picture and its price for each one. Customer will be able to select their items and place their order. It will be delivered to them in a space of time.

Q5: Will there be delivery?

A5: Yes, there will be a choice for the customer to either come collect their order, or it being delivered to their home.

Q6: Do the customers pay when delivered? How do u want the payment?

A6: Customer will pay on delivery.

Q7: Do you want a log-in system? Does the customer have to have an account and what does the account consist of?

A7: Yes, a customer must register and create an account to place an order. The user account will consist of, Name, Address, Telephone number and Email address.

Q8: How do you want each product to be presented?

A8: Each product will have a picture, and a small description and price.

Q9: Do you want the customer to be able to search for a product?

A9: Yes, the customer must be able to search for a product, and I want them to have a choice to sort out the search results, by alphabetical order, price: lowest- to- highest, highest- to- lowest and most recommended.

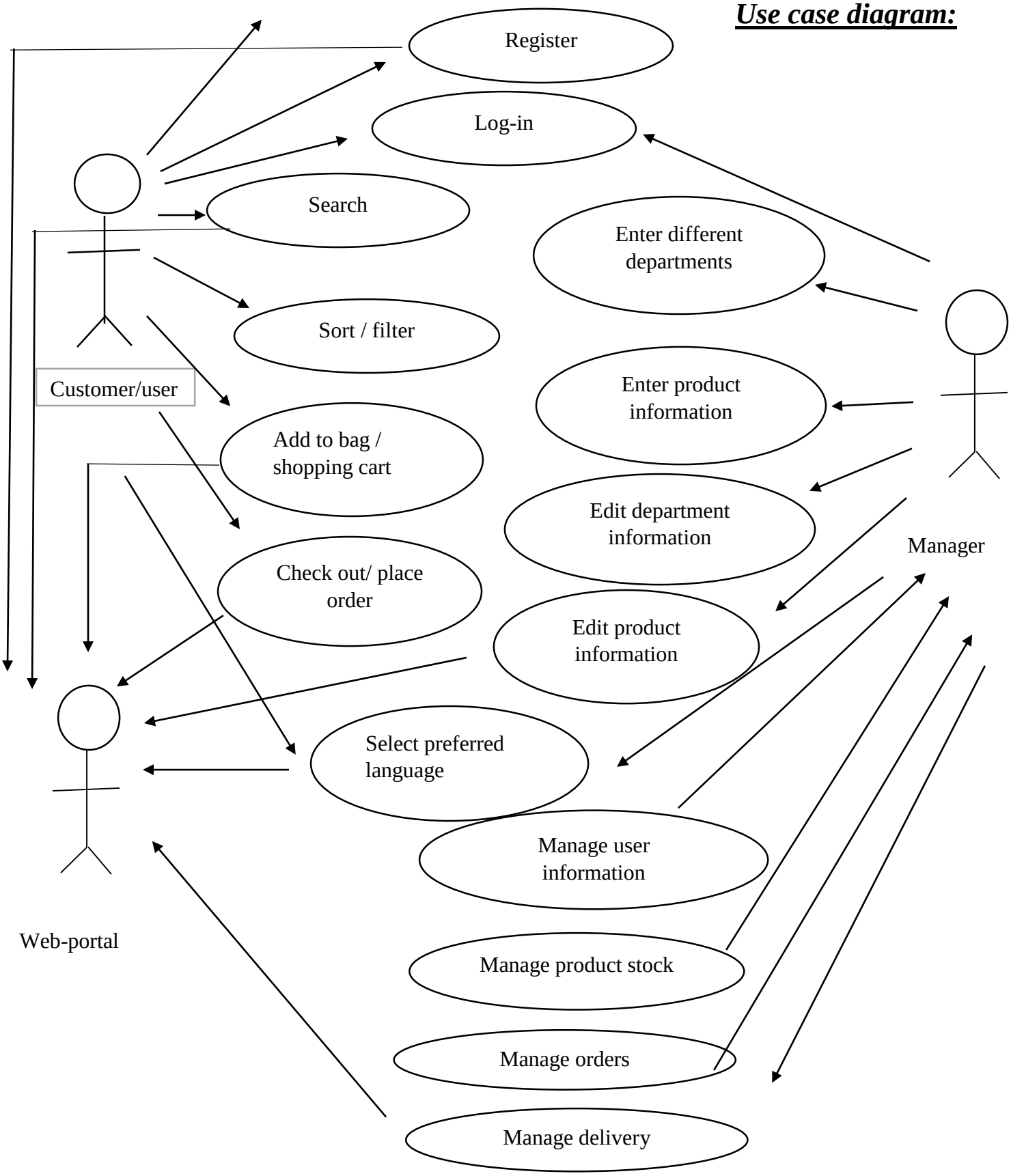
4.1Scenario:

Venicya supermarket is a website and a mobile application which helps customers order products from their homes. It will consist of all products provided within the supermarket, picture of each product with its name, price and a description when selected. The application should be free to download from either a mobile phone app store or similar services. Customers would be able to create their own account, with their information; name, address, and phone number. The customers address is included to determine the location of the closest shop. This will help the delivery to come quicker, unlike not knowing the customer's location. The customer will be able to search for a department or product. The user will also be able to sort out or filter the list of searches, by price, recommendation or alphabetical order. Once customer has found the product they are looking for they then add it to their bag/shopping kart. Once the shopper has completed shopping they then place their order. Once the order has been placed the user will get a notification of the order and so will the manager at the supermarket. A manager uses the web-portal in order to manage the system and keep the information accurate. The manager can, for instance, create a new department, enter new products in the database, entering each products information e.g. product name, description, price etc., verify if products are in stock or in low stocks, or not in stock. The software needs Internet connection to get and display results. All system information is maintained in a database. The application also has the capability of representing both summary and detailed information about the supermarket.

4.1 Use-Case model

In software and systems engineering, a **use case** is a list of action or event steps, typically defining the interactions between a role and a system, to achieve a goal while software requirements specification (**SRS**) is a document that captures complete description about how the system is expected to perform. Like the figure below.

Use case diagram:



Name: **Download Application.**	
Actor:	Customer
Brief Description:	Customer downloads app in order to use the supermarkets online shopping.
Pre-condition:	Customer must own a smart phone. Must have an account on their application store/ other services in order to install the app. Must be connected to the internet.
Post-condition:	Application will be installed on the users mobile phone.
Basic Flow:	1. Customer opens their application store / other services. 2. Customer searches for the app name “Venicya supermarket”. 3. Customer clicks “get app”. 4. Customer agrees to installation, and signs into their app store account. 5. The use case ends.
Alternate A:	1. Customer is not connected to the internet. 2. Customer will get a notification that there is an error due to no internet connection. 3. Customer would have to reinstall app.
Alternate B:	1. Customer does not have an app store account. 2. Customer is unable to download app, unless they create an account.
Alternate C:	1. Customer does not have enough storage on their mobile phone. 2. Customer must clear up enough storage in order to install the application.
DEP:	FR1,FR2

Name: **Register/Create Account.**	
Actor:	Customer
Brief Description:	The user creates a log in and becomes a registered user.
Pre-condition:	Customer must own a smart phone. Customer must have installed the application. Must be connected to the internet.
Post-condition:	User will have an account and will be logged in.
Basic Flow:	This use case starts when the User enters the system that enables him/her to create an account by entering information that is asked for the User's account. 1. The customer enters the required Account information: • Name • Address • Email address • Phone number • Account name • Password 2. The system validates the entered information. 3. The values for the customer's Account information are stored. The system notifies the User that the account has been created. 4. The use case ends
Alternate flow (User Enters Invalid User Account Information):	If throughout the Create Account process, the system determines that the User has entered invalid information, the following occurs: 1.The system points out which of the entered data is invalid and informs the customer with suggestions for entering valid data. 2.The system prompts the customer to re-enter the invalid information. 3.The customer re-enters the information and the system re-validates it. 4.If valid information is entered, the User Account Information is stored. 5.If invalid information is entered, the Entered Information is Invalid alternative flow is executed again. This continues until the User enters valid information.
DEP:	FR3

Name: **Sign-In**	
Actor:	Customer & Manager
Brief Description:	The user logs into their registered account.
Pre-condition:	User must own a smart phone. User must have installed the application. Must be connected to the internet. User must have previously created an account.
Post-condition:	Given the has user created an account and is able to log in.
Basic Flow:	This use case starts when the user has accessed the sign in feature of the system. 1. The user asks the user for their username and password. 2. The user enters his/her username and password. 3. The system then validates the information, making sure the username exists and that the password is valid. 4. The user is signed in. the system displays the home page. 5. The use case ends
Alternate A (new user):	If the user does not have an account, the system will give the user the option to create an account. Which will be displayed by the system. Once the account is created, the user is able to sign in.
Alternate B (user forgot username/ password):	If the user forgets his/her username or password, the system will give the user the option to select forget password/username, and then asks the user to answer security questions. If the user is unable to answer the security question, the system will prompt the user to enter their email address, and an email will be sent to the email address provided when the account was created.
DEP:	FR4,FR5,FR18

Name: **Search**	
Actor:	Customer
Brief Description:	The user logs into their registered account and searches for the product of their choosing.
Pre-condition:	Must be connected to the internet. Customer must have previously created an account.
Post-condition:	Given the user has created an account and is log-in.
Basic Flow:	This use case starts when the User accesses the search features to search for a particular product: 1. User enters the product name/ type. 2. User searches department type. 3. A list of potential products will be shown. 4. The use case ends.
Alternate flow (no product type exist):	If the user enters a product name that is not provided in the system, the system will prompt the user to write a valid product/ department name, because no such name exists within the database.
DEP:	FR6, FR7, FR8, FR11, FR12 FR13

Name: **Sort/ filter**	
Actor:	Customer
Brief Description:	The user sorts/filters their search.
Pre-condition:	Must be connected to the internet. User must be logged in.
Post-condition:	Given the has user is logged in and has searched for a product type, name or department name.
Basic Flow:	This use case starts when the user has searched for a specific product and a list of products has been displayed. 1. User selects sort and chooses how they want the list to be sorted: • Most recommended • Lowest price to highest • Highest price to lowest. • Alphabetical order. 2. User chooses to filter the list: • Brand • Product type • Price • Range 3. The use case ends.
Alternate flow:	None.
DEP:	FR9, FR10 FR14, FR15

Name: **Add to bag/ shopping cart**	
Actor:	Customer
Brief Description:	The user selects their searched products.
Pre-condition:	Must be connected to the internet. User must be logged-in. User must have searched for products.
Post-condition:	Given the user is logged-in and has found their products.
Basic Flow:	This use case starts when the user has searched for their desired products. 1. User has searched for product 2. User selected their product 3. User selects the quantity 4. User then clicks on the add to bag button 5. The use case ends
Alternate A(product out of stock):	If the user selects an item and it is out of stock the product will not be added into their bag, and the system will inform the user that the product is out of stock and will be in stock soon.
DEP:	FR16

Name: **Check out/ Place order**	
Actor:	Customer
Brief Description:	Given the user has selected all the products they require, added them to their bag customer then completes their order.
Pre-condition:	Must be connected to the internet. User must be logged-in. User must have searched for products. User should have their required items in their bag.
Post-condition:	Given the user is logged-in and has found their products and added them to their shopping bag.
Basic Flow:	This use case starts when the user has filled their shopping bag. 1. User completes their shopping 2. User makes sure of the items saved in their shopping bag. 3. User completes their shopping by placing their order. 4. The use case ends
Alternate A(cancels their order):	If the user has decided they want to cancel their order, the system displays a button “cancel order” which the user would have a limited time in which they could do so, once canceled the system informs the administrator and the user that the order has been canceled.
DEP:	FR17

Name: **Select preferred language.**	
Actor:	Customer & Manager
Brief Description:	The user selects their preferred language.
Pre-condition:	Must be connected to the internet. User must be logged-in.
Post-condition:	Given the user has a created an account and is logged-in.
Basic Flow:	This use case starts when the user has logged-in: 1. User has logged-in 2. User goes on app settings 3. User selects languages 4. User select their preferred language, Arabic/English. 5. System changes to the language selected. 6. The use case ends.
Alternate flow:	None.
DEP:	FR23

Name: **Enter department types**	
Actor:	Manager
Brief Description:	The user/manager enters all the different departments.
Pre-condition:	Must be connected to the internet. User must be logged-in.
Post-condition:	Given the user is logged-in and is an administrator..
Basic Flow:	This use case starts when the user is creating the departments: 1. User is logged in 2. User enters the information for each department: • Department name • Department type • What the department entails • Products for the department 3. User edits the departments information when needed 4. The use case ends
Alternate flow(not an administrator):	If the user wants to add/edit departments, and is not an administrator, they are not able to enter and change anything within the system.
DEP:	FR19

Name: **Enter product types**	
Actor:	Manager
Brief Description:	The user/manager enters all the different product types.
Pre-condition:	Must be connected to the internet. User must be logged-in.
Post-condition:	Given the user is logged-in and is an administrator..
Basic Flow:	This use case starts when the user is entering product information: 1. User uploads a picture of the product 2. User uploads the information of the product: • Product name • Product price • Product department • Product brand • Product description • Product in stock or not 3. User must connect the product to the right department. 4. Given that the product information has been uploaded, the user can edit the product information when necessary 5. The use case ends.
Alternate flow:	None
DEP:	FR18, FR20

Name: **Manage user information**	
Actor:	Manager/ Customer
Brief Description:	User will be able to update the User Account Information maintained in the account.
Pre-condition:	Must be connected to the internet. Must have an account. User must be logged-in.
Post-condition:	The data entered is stored in the users account.
Basic Flow:	This use case starts when the User accesses the system feature that enables him/her to update the information that is maintained in the User's account. 1.The system displays the information currently stored for the User. 2.The User enters the chosen information values and requests that the system saves. 3.The system validates the entered information. 4.The values for the Account are stored in the User's account. The system notifies the User that the account has been updated. 5.The use case ends
Alternate flow (User Enters Invalid User Account Information):	If during Modification, the system determines that the User entered invalid information, the following occurs: • The system points out which entered data is invalid and gives the User suggestions for entering valid data. • The system prompts the User to re-enter information. • The User re-enters the information and the system re-validates it. • If valid information is entered, the User Account Information is stored.
DEP:	FR21

Name: **Manage product stock**	
Actor:	Manager/ system
Brief Description:	User will be able to update product stock, which automatically updates the app.
Pre-condition:	Must be connected to the internet. Must have an account. User must be logged-in.
Post-condition:	The database will be updated, every time a product is purchased.
Basic Flow:	This use case starts when the User purchases an item, the database is automatically updated, informs the manager when a product is in need of re-stocking: 1. The system will inform the managers and update his/her database, when a product is purchased 2. The system then automatically reduces the stock. 3. The system notifies the user when a product is low in stock 4. The system notifies the user when a product is out of stock, and needs to be re stocked 5. The user then restocks the product and updates the database 6. The use case ends
Alternate flow (must have internet connection):	If the system and manager are not connected to the internet, it leads to many problems, the database will not be updated and customers could order a product that is not in stock, without knowledge because the system has not been updated.
DEP:	FR22

Name: **Manage orders.**	
Actor:	Manager
Brief Description:	User will be able to manage customers' orders
Pre-condition:	Must be connected to the internet. Must have an account. User must be logged-in.
Post-condition:	Customer's orders will be successfully ordered
Basic Flow:	This use case starts when the customer has completed chopping and placed their order: 1. When order has been placed, the system notifies the user/workers. 2. User then makes sure of the customer's order, and gathers all the items selected by the customer. 3. The system is then updated, and reduces each item to update the sock. 4. Order has been completed, the system notifies the customer and the user and assigns each order with a number. 5. The use case ends.
Alternate flow (mix up orders):	If during the order stage, the system crashes and mixes up the assigned number to a wrong customer this could lead to a lot of problems.
DEP:	

Name: **Manage delivery**	
Actor:	Manager
Brief Description:	Given the user has competed the order successfully, they should be able to deliver the order to the correct customer.
Pre-condition:	Must be connected to the internet. Must have an account. User must be logged-in.
Post-condition:	Order is successfully delivered.
Basic Flow:	This use case starts when the customer's order has been successfully completed: 1. Order is now complete. 2. The system informs the deliverer that order is ready. 3. The deliver then takes order and delivers it to the customer. 4. Once the delivery is complete, the system informs the manager and the customer. 5.The use case ends.
Alternate flow (order goes to the wrong number):	If during the ordering/packaging stage the system gives an order an incorrect number, this could lead to the order being delivered to the wrong customer.
DEP:	

5. 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.

5.1 External interface Requirements

This section offers a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces.

5.1.1 User interfaces

A first time user of the mobile application would see the log-in page when she/he opens the application. If the user is not registered, she/he should be able to do so on the log-in page. If the user is not a first-time user, she/he should be able to see the search page straight when the app is opened. 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 details, address, e-mail address, phone number and password. Also, the user can set the mobile application to his/her preferred language. When a user searches by product type, this view should be the default one. The sorting header allows the user to sort the results according to most recommended, price, product, name, description. Each result item includes information about the product. There is also a filtering option, where the user can choose to filter the results by increasing or decreasing the price or most popular. It will then show you the list of items that meet the search, and will list it the way the user has filtered it. The administrators interact with the system through a web-portal. An administrator should be able to log in to the web-portal where he/she can manage the system by for example editing products, departments, shop information or user information.

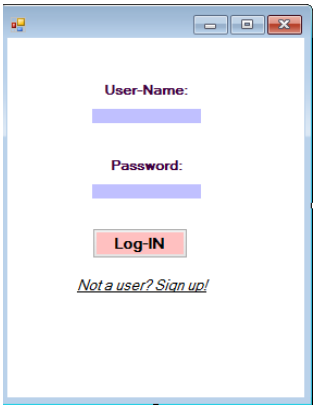


Figure 1- Log-in

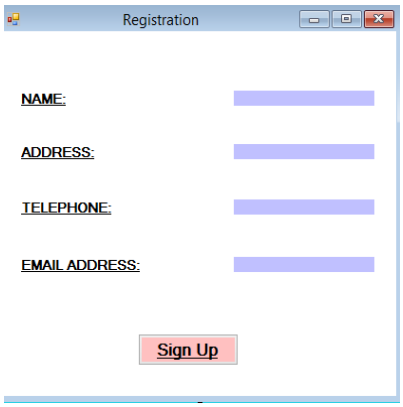


Figure 2 - Registration

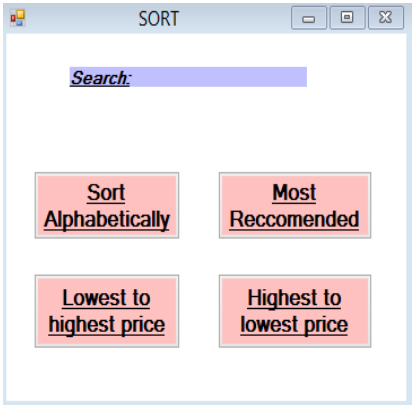


Figure 3- Sorting search

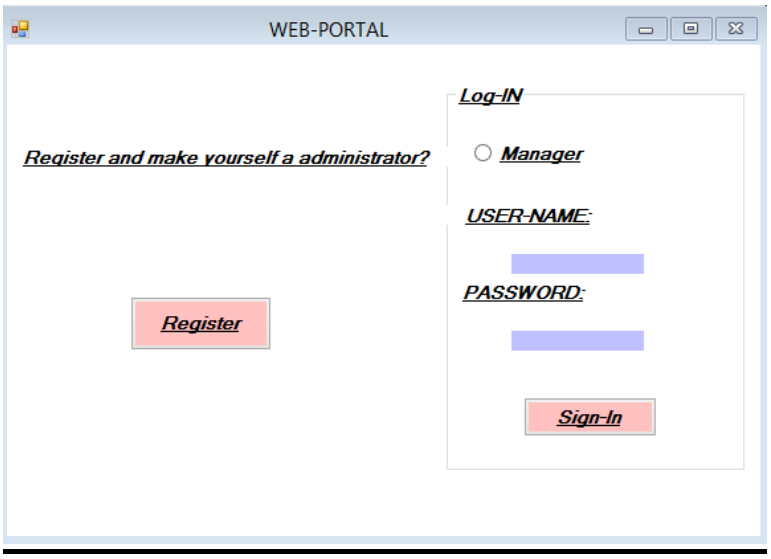


Figure – 4 Web-portal signing in page

5.1.2 Hardware interfaces

The hardware connection to the database server is managed by the underlying operating system on the mobile phone and the web server.

5.1.3 Software interfaces

The mobile application communicates with the internet. 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.

5.1.4 Communications interfaces

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

5.2 Functional requirements

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

The User

ID: FR1

TITLE: Download application

DESC: A user should be able to download the mobile application through either an app 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

ID: FR2

TITLE: New release/ Update

DESC: When a new/updated version or release of software is released, the user should get a notification. 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

ID: FR3

TITLE: User registration

DESC: Given that a user has downloaded the mobile application, then the user should now be able to register. The user must provide user-name, password, name, address, phone number and e-mail address.

RAT: In order for a user to register on the mobile application.

DEP: FR1,10

ID: FR4

TITLE: User log-in

DESC: Known that a user has registered, the user should be able to log in to the app. 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 app.

DEP: FR1, FR3

ID: FR5

TITLE: Retrieve password

DESC: when the user has registered, then the user must be able to retrieve his/her password by email.

RAT: User - retrieve his/her password.

DEP: FR1

ID: FR6

TITLE: Search

DESC: Given that a user is logged in to the app, then the first page that is shown should be the home page which has a search bar. The user should be able to search for a product, according to several search options. The search options are Price, Description and Department. 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 product.

DEP: FR4

ID: FR7

TITLE: Search result.

DESC:

- Search results can be viewed as a list. On the list the relevant product according to the user's input is shown. If the result contains more products than what can be displayed on the screen at one time, the user should be able to scroll through them.
- A specific picture will represent a specific product. By each picture there should be the price and a small description of each item.
- The list should show all the products that match the user's search, with different brands prices and which is the most popular.
- The list view should include a button that, when selected, should display different filtering options in a filtering menu.

RAT: The way results are displayed.

DEP: FR6,11

ID: FR8

TITLE: Selecting the picture link

DESC: A user should be able to select the picture link, which is included on all result items. The link will direct the user to an information page, which includes a picture of the product, the product name, description of the product, department and price.

RAT: In order to show information about the products.

DEP: FR7, FR8

ID: FR9

TITLE: Search by price

DESC: 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

ID: FR10

TITLE: Search by department

DESC: A user should be able to input the department. By default, the most recommended in this department will be displayed. The result is displayed in a list view by default.

RAT: User to search by department.

DEP: FR7

ID: FR11

TITLE: Accepted input for name and department search

DESC: Integers should be accepted as input when a user searches by product name or department. If the system receives an invalid input the user should be informed and prompted to insert an accepted input.

RAT: User to search with valid input.

DEP: FR9, FR10

ID: FR12

TITLE: Search by product brand

DESC: A user should be able to select a brand in a given list as input.

RAT: User to search by product brand.

DEP: FR7, FR6

ID: FR13

TITLE: No match found

DESC: If no match is found the user should be informed but kept on the search page in order to search again.

RAT: User to conduct a new search if no match is found.

DEP: FR5

ID: FR14

TITLE: Sorting results

DESC: When viewing the results in a list, a user should be able to sort the results according to product type, description and price.

- When sorting by product name, the results should be ordered alphabetically.
- When sorting by price the results should be ordered from cheapest to most expensive.
- When sorting by most recommended the results should be ordered in the most recommended to the least recommended
- 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: User to sort results in a list.

DEP: FR7, FR6

ID: FR15

TITLE: Filtering results

DESC: When viewing the results in a list, the user should be able to filter the results in a filtering menu. The filtering options include:

- increasing or decreasing the maximum price
- choosing a product type
- choosing most recommended
- choosing a product brand
- choosing department

When filtering the results, only the existing results shall be affected and a new search query should not be sent.

RAT: User to filter results in a list or a map.

DEP: FR6, FR7, FR8

ID: FR16

Title: Add to bag/ shopping cart.

DESC: When the user has found what they are looking for, they select the product and add it to their bag to save and when done shopping being able to order.

RATE: User to Add products to their bag/ shopping cart.

DEP: FR6, FR7, FR8

ID:FR17:

TITLE: Checkout – Place order

DESC: A user has completed their search/ shopping, they go into their nag to make sure of all the products they have selected, once done they select the button Checkout to finalize their order.

Rate; User to checkout/ place order.

DEP: FR16

TITLE: Mobile application - Profile page

DESC: On the app, the user must have a profile page. On the profile page a user can edit his/her information, which includes the password, e-mail address, name, 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 Arabic and English.

RAT: In order for a user to have a profile page on the mobile application.

DEP: FR1

Manager

ID: FR18

Feature: Manager log in
In order to manager the system:
Manager Should be logged in to the web-portal
Scenario: Successful log-in
Assumed that the manager wants to log in:
When the manager logs in, the user should be logged in as an administrator

ID: FR18

Feature: Manage product types
In order to have a list of product types
A manager
Should be able to manage the product types

Scenario: Add a new product type
Assumed that the manager is logged in
When the manager creates a new product type
Then the new product type should be added to the list of product types

Scenario: Editing an existing product type
Given the manager is logged in
When the manager edits an existing product type
Then the product type will be updated in the list

Scenario: Delete a product type
Assumed the manager is logged in
When the manager deletes a product type
Then the deleted product type should be removed from the list.

ID: FR19

Feature: Manage departments
In order to have a list of departments

A manager
Must be able to manage departments

Scenario: Add a new department
Assumed the manager is logged in
When the manager creates a new department
Then the new department should be added to the list.

Scenario: Editing an existing department
Assumed the manager is logged in
When the manager edits an existing department
Then the department should be updated in the list of departments

Scenario: Delete a department
Given the manager is logged in
When the manager deletes a department
Then the deleted department must be removed from the list of departments.

ID: FR20

Feature: Manage product information
In order to manage product information
A manager
Must be logged in to the web-portal

Scenario: Add product information
Assumed the manager is logged in
When the manager adds product information
Then the information should be added

Scenario: Delete product information
Given the manager is logged in
And information about a product exists
When information has been deleted
Then the information must be deleted

Scenario: Edit product information
Manager is logged in
And information about the product exists
When the manager edits the information
Then the information about the product should be edited

ID: FR21

Feature: Manage users
In order to keep track of the users

A manager
Must be able to manage the users

Scenario: Edit an existing user's information
Assumed the manager is logged in
When the manager edits an existing user
Then the user information must be updated

Scenario: Delete/Inactivate an existing user
Given the manager is logged in
When the manager deletes an existing user
Then the user must be deleted

ID: FR22

Feature: Manage product stock
In order to keep track of product stocks
The manager
Should be able to manage the product stock

Scenario: Add a new restaurant owner
Assumed the manager is logged in
When the manager updates the stock
Then the product stock must be updated

Scenario: Edit existing stock
Given the manager is logged in
When the manager edits existing product stock
Then the product stock should be updated

ID: FR23

Feature: Selecting preferred language on the web-portal
In order to understand the web-portal
the manager
Should be able to select their preferred language

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

Scenario: Select Arabic
Given the manager wants to select a preferred language
When the manager chooses Arabic as a new language
Then the web-portal will show all text in Arabic.

5.3Non-Functional Requirement:

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

5.3.1Performance:

The system must be interactive and the delays involved must be less. So in every action-response of the system, there are no immediate delays. In case of opening app page, of popping error messages and saving the settings or sessions there is delay much below 2 seconds, in case of opening databases, sorting questions and evaluation there are no delays.

The search feature should be prominent and easy to find for the user. The different search options should be evident, simple and easy to understand. 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.

5.3.1 Availability:

If the internet service gets interrupted while sending information to the server, the information can be sent again for verification.

The application is available 24/7, customer can order anytime. If the order is being placed after working hours, the user will be notified that the order will be delivered the next day. The order will be automatically saved in the database, and each number will be sorted by the time the order had been placed.

5.3.2 Security:

The main security concern is for users account hence proper login mechanism should be used to avoid hacking.

Security of the communication between the system and server. The messages should be encrypted for log-in communications, so others cannot get user-name and password from those messages. If an admin tries to log in to the web portal with a non-existing account, then the admin will not be logged in. The admin should be notified about log-in failure.

5.3.4 Portability:

The application should be portable with iOS and Android.

5.3.5 Usability:

The application should be user friendly, anyone with experience with a smart phone will be capable of using this application.