

**Capstone Project Report**

**Report 3 – Software Requirement Specification**

– Hanoi, August 2019 –

**Table of Contents**

[I. Project Report 3](#_Toc72138553)

[1. Status Report 3](#_Toc72138554)

[2. Team Involvements 3](#_Toc72138555)

[3. Issues/Suggestions 3](#_Toc72138556)

[II. Software Requirement Specification 4](#_Toc72138557)

[1. Overall Description 4](#_Toc72138558)

[1.1 Product Overview 4](#_Toc72138559)

[1.2 Business Rules 5](#_Toc72138560)

[2. User Requirements 6](#_Toc72138561)

[2.1 System Actors 6](#_Toc72138562)

[2.2 Use Cases 6](#_Toc72138563)

[3. Functional Requirements 8](#_Toc72138564)

[3.1 System Functional Overview 8](#_Toc72138565)

[3.2 <<Feature Name 1>> 10](#_Toc72138566)

[3.3 <<Feature Name 2>> 10](#_Toc72138567)

[4. Non-Functional Requirements 11](#_Toc72138568)

[4.1 External Interfaces 11](#_Toc72138569)

[4.2 Quality Attributes 12](#_Toc72138570)

[5. Other Requirements 14](#_Toc72138571)

[5.1 Appendix1 - Messages List 14](#_Toc72138572)

[5.2 Appendix2 - … 14](#_Toc72138573)

[5.3 … 14](#_Toc72138574)

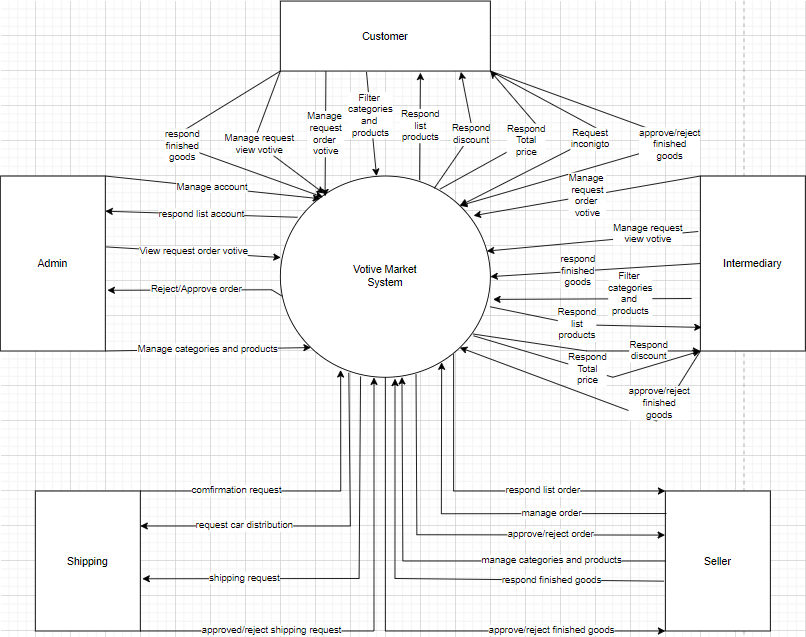
# I. Record of Changes

|  |  |  |  |
| --- | --- | --- | --- |
| Date | A\* M, D | In charge | Change Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

# II. Software Requirement Specification

## 1. Product Overview



## 2. User Requirements

### 2.1 Actors

*[An actor is a person (or sometimes another software system or a hardware device) that interacts with the system to perform a use case. Following are some questions you might ask to help user representatives identify actors*

* *Who (or what) is notified when something occurs within the system?*
* *Who (or what) provides information or services to the system?*
* *Who (or what) helps the system respond to and complete a task?*

*This part gives the description of system actors, you can follow the table form as below]*

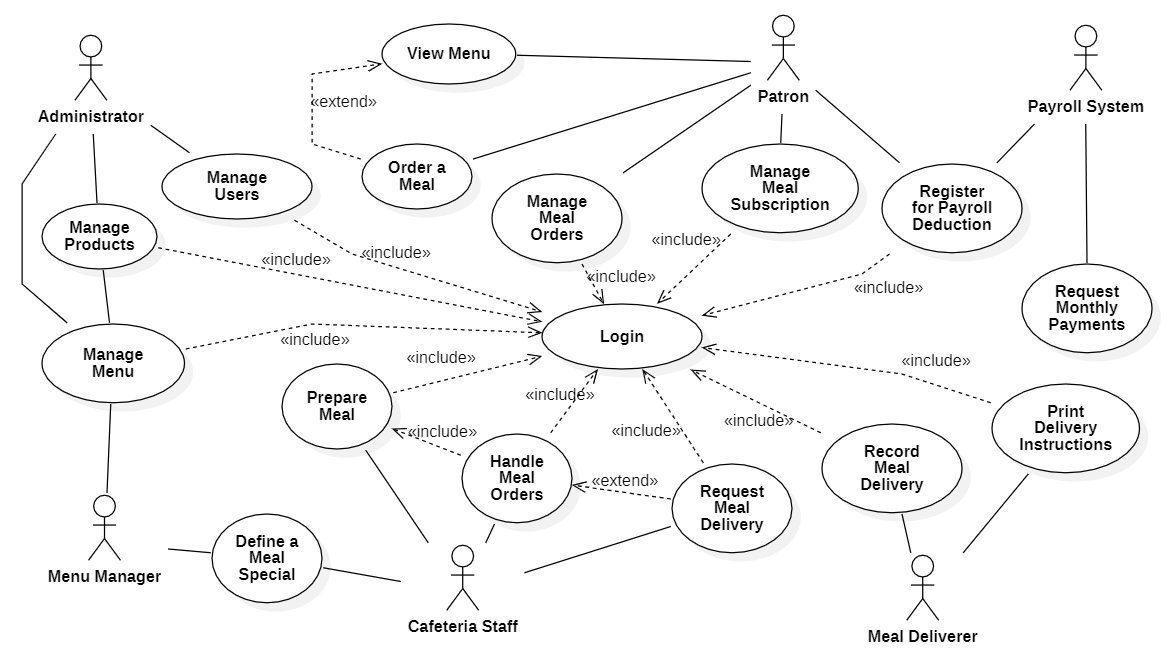
|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | Administrator |  |
| 2 | Menu Manager |  |
| 3 | … |  |

### 2.2 Use Cases

*[A use case (UC) describes a sequence of interactions between a system and an external actor that results in the actor being able to achieve some outcome of value. The names of use cases are always written in the form of a verb followed by an object. Select strong, descriptive names to make it evident from the name that the use case will deliver something valuable for some user]*

#### 2.2.1 Diagram(s)

*[Provide the UC diagram(s) to show the actor-UCs and UC-UC relationships like the sample below. You can have multiple UC diagrams for the system]*



#### 2.2.2 Descriptions

*This part describes the use cases & their main flow (the list of the user actions and corresponding system responses that will take place during execution of the use case under normal, expected conditions), you can follow the table form as below]*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Use Case** | **Actors** | **Use Case Description & Main Flow** |
| 01 | View Menu | Patron |  |
| 02 | Order a Meal | Patron |  |
| 03 | … |  |  |

## 3. Functional Requirements

### 3.1 System Functional Overview

*[Provide functionality overview of software system: screen flow, screen descriptions, system user roles, screen authorization, non-screen functions, ERD]*

#### 3.1.1 Screens Flow

*[This part shows the system screens and the relationship among screens. You can draw the Screens Flow for the system in the form of diagram as below. Please note that beside the normal flat screen, we might have the oval notation for pop-up screen (Import Order) or a screen with multiple information tab (Order Details), etc. You may also use text or background format for different visuality purpose]*



#### 3.1.2 Screen Descriptions

*[Provide the descriptions for the screens in the Screens Flow above]*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **Screen** | **Description** |
| 1 | Order Meals | Create Order | <<Screen Brief description>> |
| 2 | Order Meals | Change Order |  |
| 3 | .. |  |  |

#### 3.1.3 Screen Authorization

*[Provide the system roles authorization to the system features (down to screens, and event to the screen activities if applicable) in the table form as below – replace Role1, Role2,… with your specific system user role names]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen** | **Role-Name1** | **Role-Name2** | **Role-Name3** | **…** |
| <<Screen Name1>> | X |  | X | X |
| <<Screen Activity>> |  |  | X | X |
| <<Screen Name2>> | X |  | X |  |
| Query All Data | X |  |  |  |
| Query Own Data |  |  | X |  |
| Query Managed Data |  |  | X |  |
| Add New Data |  |  | X | X |
| Update All Data |  |  |  | X |
| Update Own Data |  |  |  | X |
| Update Managed Data |  |  |  | X |
| Delete Data |  |  |  |  |
| … |  |  |  |  |

#### 3.1.4 Non-Screen Functions

*[Provide the descriptions for the non-screen system functions, i.e batch/cron job, service, API, etc.]*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **System Function** | **Description** |
| 1 | <<Feature Name>> | <<Function Name1>> | <<Function Name1 Description>> |
| 2 | … |  |  |

#### 3.1.5 Entity Relationship Diagram

*[Provide the entity relationship diagram and the entity descriptions in the table format as below]*



**Entities Description**

|  |  |  |
| --- | --- | --- |
| **#** | **Entity** | **Description** |
| 1 | User |  |
| 2 | Meal |  |
| 3 | Meal Subscription |  |
| 4 | … |  |

### 3.2 <<Feature Name 1>>

#### 3.2.1 <<Function Name 1>>

*[A function can be a screen or a non-screen function (listed in the part 3.1.5 above). In this part, you need to provide the details on the related function, focus on mentioning below information*

* *Function trigger: how this function is triggered (navigation path, a timing frequency, etc.*
* *Function description: actors/roles, purpose, interface, data processing, etc.*
* *Screen layout: mock-up prototype of the screen, sample below is for Manage Products screen*

**

* *Function Details: provide explanation for the data, validation, business rules, functionalities (for both normal cases and abnormal cases), etc. of the function so that the reader can image how it work.*

*]*

#### 3.2.2 <<Function Name 2>>

…

### 3.3 <<Feature Name 2>>

…

## 4. Non-Functional Requirements

4.1 External Interfaces

#### 4.1.1.User interfaces

**UI-1**: The web application display shall conform to the user and provide a friendly interface.

**UI-2**: The font size is easy to distinguish where is the title, where is the content.

**UI-3**: Consistent interface: font size, font, color, the logo must be in sync.

**UI-4**: The web application will support responsive screens on mobile, tablet, and desktop Quality Attributes

#### 4.2.1 Usability

**USA-1**: The steps and flows are short and easy to understand so that users can interact with the application without training.

**USA-2**: The user interface is clean, friendly, and consistent from the layout, component to the

button, text field.

**USA-3**: The input action is flexible with both keyboard and mouse.

**USA-4**: Error and warning messages are simple.

#### 4.2.2 Reliability

#### 4.2.3 Performance

* Reduce as less as possible query to database in loop
* Resource utilization: use very few computer resources such as ram, memory, CPU.

#### 4.2.4 Dependence

##### 4.2.4.1. Security

- Only the Admin account has access to system management.

- The system does not allow users who are not FPT University students to register for an account on the system but with other roles as recruiters and sellers.

- The recruiter role will not be able to sell and the seller role will not be able to post job vacancies.

- The system uses spring security to classify the user type for the system.

##### 4.2.4.2 Safety

- The password of the account is encrypted since creating a moment.

- The password should be hidden in the client.

#### 4.2.5 Supportability

- The account's password must be MD5 encrypted

- Named according to the Coding Convention

#### 4.2.6 Design Constraints

The back-end programming language is .NET

Front-end programming language is ReactJS

#### Using Spring MVC

## 5. Requirement Appendix

### 5.1 Business Rules

|  |  |
| --- | --- |
| **ID** | **Description** |
|  | The email address must be valid |
|  | Password must be at least 8 characters in length |
|  | Password must be encrypted |
|  | When registering or changing a password, the user must enter the new password twice |
|  | A guest cannot register with a username and email that has already been registered |
|  | User cannot update their email once it has been registered |
|  | When changing a password, the new password must not be the same as the current password |
|  | Format of date must be “dd/mm/yyyy” |
|  | When an account is registered, the user must choose their roles |
|  | Email is unique and cannot be duplicated in the database |
|  | Format of money must be “10.000” |
|  | Format DateTime of comment restaurant must be “dd-MMM-yyyy HH:mm:ss” |
|  | User must provide their email and password when logging into the website |
|  | When updating the profile, the phone number is unique and cannot be duplicated in the database |
|  | The deadline of request when creating or updating must be in the future |
|  | After creating or re-opening a request to find a job or recruitment, the status of the request is PENDING |
|  | Students can repost a request find a job when the request’s status is COMPLETE or EXPIRED |
|  | A recruiter can repost request recruitment when the request’s status is COMPLETE or EXPIRED |
|  | Students can only apply for job postings when the job posting's status is APPROVED |
|  | A recruiter can only hire workers when the find job posting's status is APPROVED |
|  | When deleting posts that are not " APPROVED" status, the posts on the homepage are not deleted |
|  | When add, update or report, the blanks must be filled in completely. |

### 5.2 Common Requirements

### 5.3 Application Messages List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Message code** | **Message Type** | **Context** | **Content** |
|  | MSG01 | In red, Toast Message | Wrong account | Email hoặc mật khẩu không chính xác. |
|  | MSG02 | In green, Toast Message | Logout success | Đăng xuất thành công. |
|  | MSG03 | Toast Message | Notify actions | Bạn phải hoàn thành ít nhất một đơn hàng trước khi đánh giá cửa hàng. |
|  | MSG04 | Toast Message | Notify actions | Không thẻ xóa thẻ này vì thẻ này đang được sử dụng. |
|  | MSG05 | Toast Message | Notify actions | Cập nhật mật khẩu mới thành công! |
|  | MSG06 | Toast Message | Notify actions | Có lỗi xảy ra. Vui lòng thử lại. |
|  | MSG07 | Toast Message | Notify actions | Mật khẩu cũ không đúng! |
|  | MSG08 | Toast Message | Notify actions | Mật khẩu xác nhận không giống với mật khẩu mới! |