

**Capstone Project Report**

**Report 3 – Software Requirement Specification**

– Ho Chi Minh City, January 2021 –

**Table of Contents**

[I. Project Report 3](#_Toc50989330)

[1. Status Report 3](#_Toc50989331)

[2. Team Involvements 3](#_Toc50989332)

[3. Issues/Suggestions 3](#_Toc50989333)

[II. System Requirement Specification 4](#_Toc50989334)

[1. Overall Description 4](#_Toc50989335)

[1.1 Product Overview 4](#_Toc50989336)

[1.2 Business Rules 5](#_Toc50989337)

[2. User Requirements 6](#_Toc50989338)

[2.1 Overview 6](#_Toc50989339)

[2.2 <<Feature Name 1 – i.e Order Meals>> 7](#_Toc50989340)

[2.3 <<Feature name 2 – i.e: Meal Subscriptions>> 10](#_Toc50989341)

[2.4 <<Next Feature Name..>> 11](#_Toc50989342)

[3. Functional Requirements 12](#_Toc50989343)

[3.1 System Functional Overview 12](#_Toc50989344)

[3.2 <<Feature Name 1>> 14](#_Toc50989345)

[3.3 <<Feature Name 2>> 14](#_Toc50989346)

[4. Non-Functional Requirements 15](#_Toc50989347)

[4.1 External Interfaces 15](#_Toc50989348)

[4.2 Quality Attributes 16](#_Toc50989349)

[5. Other Requirements 18](#_Toc50989350)

[5.1 Appendix1 - Messages List 18](#_Toc50989351)

[5.2 Appendix2 - … 18](#_Toc50989352)

[5.3 … 18](#_Toc50989353)

# I. Project Report

## 1. Status Report

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Work Item** | **Status** | **Notes (Work Item in Details)** |
| 1 |  | Pending |  |
| 2 |  | In Progress |  |
| 3 |  | Completed |  |

## 2. Team Involvements

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task** | **Member** | **Notes (Task Details, etc.)** |
| 1 |  | KienNT |  |
| 2 |  | TuanTV |  |
| 3 |  | AnhLM |  |

## 3. Issues/Suggestions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Issue** | **Status** | **Notes (Solution, Suggestion, etc.)** |
| 1 |  | Pending |  |
| 2 |  | In Progress |  |
| 3 |  | Completed |  |

# II. Software Requirement Specification

## 1. Overall Description

### 1.1 Product Overview

*[This section presents a high-level overview of the product and the environment in which it will be used, the anticipated users, and known constraints, assumptions, and dependencies]*

*[This section Describe the product's context and origin of the product you are developing. Is it the next member of a growing product line, the next version of a mature system, a replacement for an existing application, or an entirely new product? If this SRS defines a component of a larger system, state how this software relates to the overall system and identify major interfaces between the two. Consider including visual models such as a context diagram or ecosystem map to show the product's relationship to other systems or anything else in the universe.*

*The context diagram presents the boundary and connections between the system you’re developing and everything else in the universe. This identifies external entities (or terminators – software, hardware, human components, and other systems) outside the system that interface to it in some way, as well as data, control, and material flows between the terminators and the system.*

*An ecosystem map shows all of the systems related to the system of interest that interact with one another and the nature of those interactions. It represents scope by showing all the systems that interconnect (directly or indirectly) and that therefore might need to be modified to accommodate your new system]*

<<Sample: The Cafeteria Ordering System is a new software system that replaces the current manual and telephone processes for ordering and picking up meals in the Process Impact cafeteria. The context diagram below illustrates the external entities and system interfaces for release 1.0. The system is expected to evolve over several releases, ultimately connecting to the Internet ordering services for several local restaurants and to credit and debit card authorization services.



>>

### 1.2 Business Rules

*[Provide common business rules that you must follow. The information can be provided in the table format as the sample below]*

|  |  |
| --- | --- |
| ID | Rule Definition |
| BR-01 | Delivery time windows are 15 minutes, beginning on each quarter hour. |
| BR-02 | Deliveries must be completed between 10:00 A.M. and 2:00 P.M. local time, inclusive. |
| BR-03 | All meals in a single order must be delivered to the same location. |
| BR-04 | All meals in a single order must be paid for by using the same payment method. |
| BR-11 | If an order is to be delivered, the patron must pay by payroll deduction. |
| BR-12 | Order price is calculated as the sum of each food item price times the quantity of that food item ordered, plus applicable sales tax, plus a delivery charge if a meal is delivered outside the free delivery zone. |
| BR-24 | Only cafeteria employees who are designated as Menu Managers by the Cafeteria Manager can create, modify, or delete cafeteria menus. |
| BR-33 | Network transmissions that involve financial information or personally identifiable information require 256-bit encryption. |
| BR-86 | Only regular employees can register for payroll deduction for any company purchase. |
| BR-88 | An employee can register for payroll deduction payment of cafeteria meals if no more than 40 percent of his gross pay is currently being deducted for other reasons. |

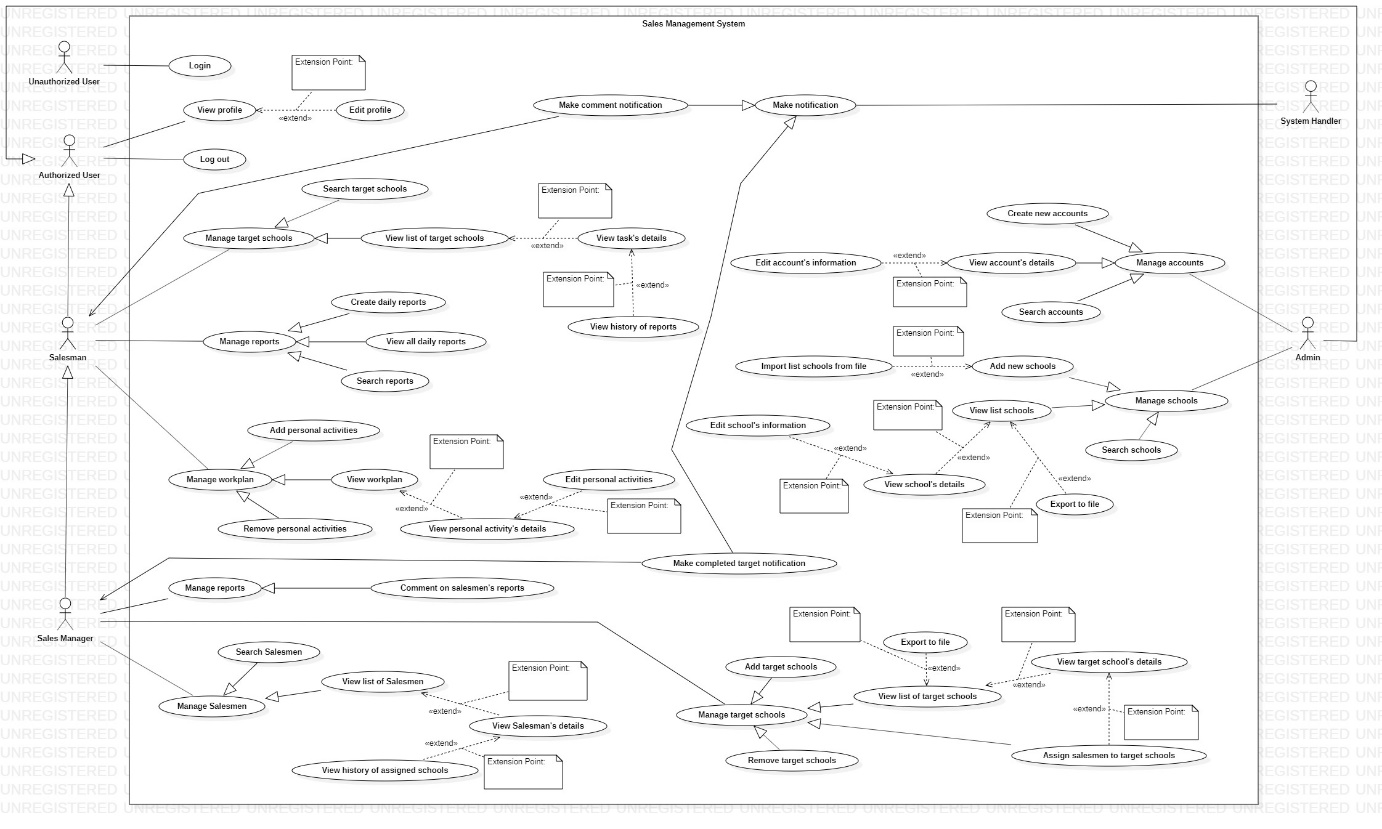
## 2. User Requirements

***(This is optional part)***

### 2.1 Overview

#### Use Case Diagram

*[Provide your use case diagram(s) which is something like the sample below]*



#### System Actors

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | Administrator | Administrators managed Accounts and the schools data. |
| 2 | Salesman | Salesman introduced and marketed Major to schools for cooperation. |
| 3 | Sales Manager | Sales Manager managed Target Schools, Salesmen and their Reports. Sales Manager also marketed as Salesman. |
| 4 | Unauthorized User | The people that have not logged into the system yet. |
| 5 | Authorized User | The people that have logged into the system with the account( Administrator, Salesman, Sales Manager) |
| 6 | <<System>> Handler | The <<System>> Handler deals with internal process. |

#### Use Cases List

*[A use case 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; This part describe the use cases, you can follow the table form as below, in which: the primary actors initiate the use case and derive the main value from it, the secondary actors are the person or system which will participate in completing execution of the use case (participates somehow in the successful execution of the use case)]*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Use Case** | **Primary Actors** | **Secondary Actors** |
| 01 | Login | Unauthorized User |  |
| 02 | View Profile | Authorized User, Salesman, Sales Manager, Administrator |  |
| 03 | Edit Profile | Authorized User, Salesman, Sales Manager, Administrator |  |
| 04 | Logout | Authorized User, Salesman, Sales Manager, Administrator |  |
| 05 | Manage Target Schools  (Search, View) | Salesman |  |
| 06 | Manage Reports  (Create, View, Search) | Salesman |  |
| 07 | Manage Work Plan- Activities  (View, Add New Personal Activity, Edit Personal Activity, Remove Personal Activity) | Salesman |  |
| 08 | Manage Reports  (Comment On Salesmen’s Report) | Sales Manager |  |
| 09 | Manage Target Schools  (View, Add, Edit, Remove, Assign Salesmen) | Sales Manager |  |
| 10 | Manage Salesmen  (Search, View) | Sales Manager |  |
| 11 | Manage accounts  (Create, View, Edit, Search) | Administrator |  |
| 12 | Manage Schools  (Add, View, Edit, Import File) | Administrator |  |
| 13 | Make comment notification | <<System>> Handler | Salesman |
| 14 | Make completed target notification | <<System>> Handler | Sales Manager |

### 2.2 <<Unauthorized User>> Overview Use Case



#### a. Login

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-01 Login** | | |
| Created By: |  | Date Created: | 15/02/2021 |
| Primary Actor: | Unauthorized User | Secondary Actors: | None |
| Description: | This use case allow the user to log into the system to use service of system. | | |
| Trigger: | The user click on the “Login” button on the Login screen. | | |
| Preconditions: | PRE-1. The user have to own an account with username and password.  PRE-2. The user has not logged into system yet.  PRE-3. The user’s device was connected to the internet when logging in. | | |
| Post-conditions: | POST-1. The user successfully log into the system, and redirect the home screen. | | |
| Normal Flow: | **1.0 Login**   1. The user opens login page via web browsers. 2. The user inputs into the username field and the password field on Login form. 3. The user clicks on “Login” button. 4. The system validates login information successfully and allows the user to access the system, redirect home screen base on the role of user. (see **1.0.E1**, **1.0.E2**) | | |
| Alternative Flows: | None | | |
| Exceptions: | **1.0.E1 The system can’t submit the form and display the error message**  1. The username and password are invalid – incorrect format. Return to step 2.  **1.0.E2 The system has failed login credentials and displays the error message**  1. The username and password are not matched the saved data.  1a. The user choose the command to cancel login. The Use case ends.  1b. The user choose the command to login again. Return to step 2. | | |
| Priority: | Medium | | |
| Frequency of Use: | Approximately 200 users, average of 500 usage per day. Peak usage load for this use case is between 6:00 A.M. and 10:00 P.M. local time. | | |
| Business Rules: |  | | |
| Other Information: | None | | |
| Assumptions: | None | | |

### https://f16.photo.talk.zdn.vn/2912878202032592742/42a05042332ec070993f.jpg<<Authorized User >> Overview Use Case

#### View Profile



|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-02 View Profile** | | |
| Created By: |  | Date Created: | 15/02/2021 |
| Primary Actor: | Authorized User | Secondary Actors: | None |
| Description: | This use case allow the users view their profile. | | |
| Trigger: | The User chooses View Detail command. | | |
| Preconditions: | PRE-1. The User has been log in to system. | | |
| Post-conditions: | POST-1. The User information is displayed in details as a form. | | |
| Normal Flow: | **2.0 View Profile**   1. The User clicks on the avatar in the navigation bar. 2. The User chooses View Detail command. 3. The system redirects the “Profile” page. The detail information user is displayed on the screen. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | Usually | | |
| Business Rules: | None | | |
| Other Information: | None | | |
| Assumptions: | None | | |

#### Edit Profile

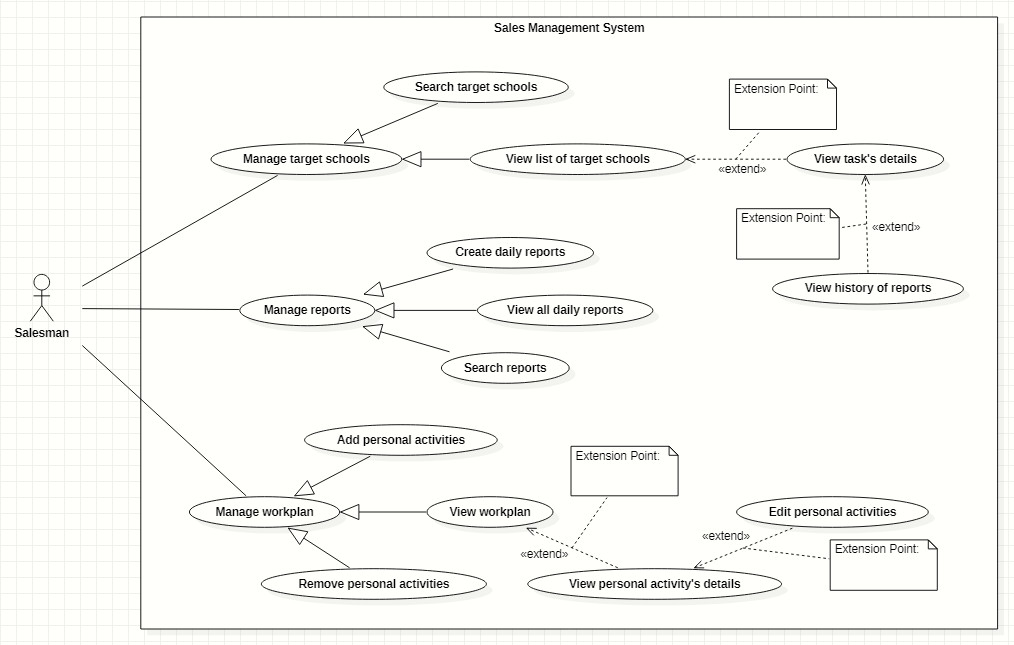


|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-03 Edit Profile** | | |
| Created By: |  | Date Created: | 15/02/2021 |
| Primary Actor: | Authorized User | Secondary Actors: | None |
| Description: | This use case allow the users edit their profile. | | |
| Trigger: | The User chooses Edit Profile command. | | |
| Preconditions: | PRE-1. The User has been log in to system.  PRE-2. The User is on the profile page. | | |
| Post-conditions: | POST-1. The User information is displayed in details as a form. | | |
| Normal Flow: | * 1. **Edit Profile**  1. On Profile screen, the User chooses “edit” button in the information row that his want to edit. 2. The User input new context in this row. 3. The User click “save” button when finish. (see **3.0.E1, 3.0.E2**) | | |
| Alternative Flows: | None | | |
| Exceptions: | **3.0.E1** **The system can’t submit the form and display the error message.**  1. Field’s information is invalid – incorrect format. Return to step 2.  **3.1.E2** **The system has failed edit and displays the error message.**  1. The data field is already existed – unique (email, phone).  1a. The user choose the command to cancel edit. The Use case ends.  1b. The user choose the command to try again. Return to step 2. | | |
| Priority: | Medium | | |
| Frequency of Use: | Usually | | |
| Business Rules: | None | | |
| Other Information: | None | | |

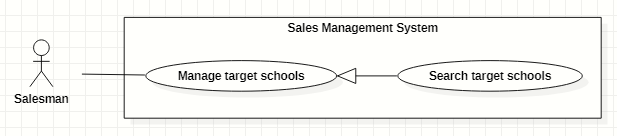
1. ***Logout***

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-04 Logout** | | |
| Created By: |  | Date Created: | 15/02/2021 |
| Primary Actor: | Authorized User | Secondary Actors: | None |
| Description: | This use case allow the users log out of the system. | | |
| Trigger: | The User chooses logout command. | | |
| Preconditions: | PRE-1. The User has been log in to system. | | |
| Post-conditions: | POST-1. The User information is displayed in details as a form. | | |
| Normal Flow: | * 1. **Logout**  1. The User clicks on the avatar in the navigation bar. 2. The User chooses Logout command. 3. The User chooses confirm to logout in confirm dialog.( see **4.1**) 4. The User return Login screen. User’s token was removed and user’s session was destroyed in client. | | |
| Alternative Flows: | * 1. **The User chooses cancel command in confirm logout dialog.**  1. Return user’s working screen. | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | Usually | | |
| Business Rules: | None | | |
| Other Information: | None | | |
| Assumptions: | None | | |

### <<Salesman>> Overview Use Case



1. ***Search Target Schools.***



|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-05 Search Target Schools** | | |
| Created By: |  | Date Created: | 15/02/2021 |
| Primary Actor: | Salesman | Secondary Actors: | None |
| Description: | This use case allow the users search the target schools. | | |
| Trigger: | The User chooses search target school command. | | |
| Preconditions: | PRE-1. The User has been log in to system. | | |
| Post-conditions: | POST-1. The User information is displayed in details as a form. | | |
| Normal Flow: | * 1. **Logout**  1. The User clicks on the avatar in the navigation bar. 2. The User chooses Logout command. 3. The User chooses confirm to logout in confirm dialog.( see **4.1**) 4. The User return Login screen. User’s token was removed and user’s session was destroyed in client. | | |
| Alternative Flows: | * 1. **The User chooses cancel command in confirm logout dialog.**  1. Return user’s working screen. | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | Usually | | |
| Business Rules: | None | | |
| Other Information: | None | | |
| Assumptions: | None | | |

## 3. Functional Requirements

### 3.1 System Functional Overview

#### a. Screen Flow

*[This part show the system screens and the relationship among screens. You can draw the Screens Flow for the system in the form of diagram as below]*



#### b. Screen Details

*[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 | .. |  |  |

#### c. 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 the specific system user role names]*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Screen** | **Role1** | **Role2** | **Role3** | **Role4** | **RoleX** |
| <<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 |  |  |  |  |  |
| … |  |  |  |  |  |

In which:

* Role1: <<role1 description>>
* Role2: <<role2 description>>
* …

#### d. 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 | … |  |  |

#### e. Entity Relationship Diagram

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



**Entities List**

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

### 3.2 <<Feature Name 1>>

#### a. <<Function Name 1>>

*[A function can be a screen or a non-screen function (listed in the part 5.1 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: mockup prototype of the screen, sample below is for Manage Products screen*

**

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

*]*

#### b. <<Function Name 2>>

…

### 3.3 <<Feature Name 2>>

…

## 4. Non-Functional Requirements

### 4.1 External Interfaces

*[This section provides information to ensure that the system will communicate properly with users and with external hardware or software elements.]*

#### a. User Interfaces

*[Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.]*

UI-1: The Cafeteria Ordering System screen displays shall conform to the Process Impact Internet Application User Interface Standard, Version 2.0 [3].

UI-2: The system shall provide a help link from each displayed webpage to explain how to use that page.

UI-3: The webpages shall permit complete navigation and food item selection by using the keyboard alone, in addition to using mouse and keyboard combinations.

#### b. Software Interfaces

*[Describe the connections between this product and other software components (identified by name and version), including other applications, databases, operating systems, tools, libraries, websites, and integrated commercial components. State the purpose, formats, and contents of the messages, data, and control values exchanged between the software components. Specify the mappings of input and output data between the systems and any translations that need to be made for the data to get from one system to the other. Describe the services needed by or from external software components and the nature of the inte-component communications. Identify data that will be exchanged between or shared across software components. Specify non-functional requirements affecting the interface, such as service levels for responses times and frequencies, or security controls and restrictions.]*

SI-1: Cafeteria Inventory System

SI-1.1: The COS shall transmit the quantities of food items ordered to the Cafeteria Inventory System through a programmatic interface.

SI-1.2: The COS shall poll the Cafeteria Inventory System to determine whether a requested food item is available.

SI-1.3: When the Cafeteria Inventory System notifies the COS that a specific food item is no longer available, the COS shall remove that food item from the menu for the current date.

SI-2: Payroll System

The COS shall communicate with the Payroll System through a programmatic interface for the following operations:

SI-2.1: To allow a Patron to register and unregister for payroll deduction.

SI-2.2: To inquire whether a Patron is registered for payroll deduction.

SI-2.3: To inquire whether a Patron is eligible to register for payroll deduction.

SI-2.4: To submit a payment request for a purchased meal.

SI-2.5: To reverse all or part of a previous charge because a patron rejected a meal or wasn’t satisfied with it, or because the meal was not delivered per the confirmed delivery instructions.

#### c. Hardware Interfaces

*[Describe the characteristics of each interface between the software and hardware (if any) components of the system. This description might include the supported device types, the data and control interactions between the software and the hardware, and the communication protocols to be used. List the inputs and outputs, their formats, their valid values or ranges, and any timing issues developers need to be aware of. If this information is extensive, consider creating a separate interface specification document]*

No hardware interfaces have been identified.

#### d. Communications Interfaces

*[State the requirements for any communication functions the product will use, including e-mail, Web browser, network protocols, and electronic forms. Define any pertinent message formatting. Specify communication security or encryption issues, data transfer rates, handshaking, and synchronization mechanisms. State any constraints around these interfaces, such as whether e-mail attachments are acceptable or not.]*

CI-1: The COS shall send an email or text message (based on user account settings) to the Patron to confirm acceptance of an order, price, and delivery instructions.

CI-2: The COS shall send an email or text message (based on user account settings) to the Patron to report any problems with the meal order or delivery.

### 4.2 Quality Attributes

*[List all the required system characteristics (quality attributes) specification. Some of the possible attributes are provided with the guide/descriptions are mentioned here]*

#### a. Usability

*[This section includes all those requirements that affect usability. For example, specify the required training time for a normal users and a power user to become productive at particular operations specify measurable task times for typical tasks or base the new system’s usability requirements on other systems that the users know and like specify requirement to conform to common usability standards, such as IBM’s CUA standards Microsoft’s GUI standards]*

#### b. Reliability

*[Requirements for reliability of the system should be specified here. Some suggestions follow:*

*Availability—specify the percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, and so on.*

*Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.*

*Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?*

*Accuracy—specifies precision (resolution) and accuracy (by some known standard) that is required in the system’s output.*

*Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand lines of code (bugs/KLOC) or bugs per function-point( bugs/function-point).*

*Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or a complete inability to use certain parts of the system’s functionality.]*

#### c. Performance

*[The system’s performance characteristics are outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.*

*Response time for a transaction (average, maximum)*

*Throughput, for example, transactions per second*

*Capacity, for example, the number of customers or transactions the system can accommodate*

*Degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)*

*Resource utilization, such as memory, disk, communications, and so forth.]*

#### d. Dependability

*[Software dependability includes a range of characteristics including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system]*

##### d1. Security

*[Specify any requirements regarding security or privacy issues that restrict access to or use of the product. These could refer to physical, data, or software security. Security requirements often originate in business rules, so identify any security or privacy policies or regulations to which the product must conform. If these are documented in a business rules repository, just refer to them.]*

##### d2. Safety

*[Specify requirements that are concerned with possible loss, damage, or harm that could result from use of the product. Define any safeguards or actions that must be taken, as well as potentially dangerous actions that must be prevented. Identify any safety certifications, policies, or regulations to which the product must conform.]*

#### e. Supportability

*[This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, and maintenance utilities.]*

#### f. Design Constraints

*[This section indicates any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, and so on.]*

#### g. Support Documents

*[Describes the requirements, if any, for o-line user documentation, help systems, help about notices, and so forth.]*

#### h. Purchased Components

*[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility and interoperability or interface standards.]*

## 5. Other Requirements

*[Examples are: legal, regulatory or financial compliance, and standards requirements; requirements for product installation, configuration, startup, and shutdown; and logging, monitoring and audit trail requirements. Instead of just combining these all under "Other," add any new sections to the template that are pertinent to your project. Omit this section if all your requirements are accommodated in other sections. ]*

### 5.1 Appendix1 - Messages List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Message code** | **Message Type** | **Context** | **Content** |
| 1 | MSG01 | In line | There is not any search result | *No search result.* |
| 2 | MSG02 | In red, under the text box | Input-required fields are empty | *The \* field is required.* |
| 3 | MSG03 | Toast message | Updating asset(s) information successfully | *Update asset(s) successfully.* |
| 4 | MSG04 | Toast message | Adding new asset successfully | *Add asset successfully.* |
| 5 | MSG05 | Toast message | Confirming email of asset hand-over is sent successfully | *A confirmation email has been sent to {email\_address}.* |
| 6 | MSG06 | Toast message | Resetting asset information successfully | *Return asset(s) successfully.* |
| 7 | MSG07 | Toast message | Deleting asset information successfully | *Delete asset(s) successfully.* |
| 8 | MSG08 | In red, under the text box | Input value length > max length | *Exceed max length of {max\_length}.* |
| 9 | MSG09 | In line | Username or password is not correct when clicking sign-in | *Incorrrect user name or password. Please check again.* |

### 5.2 Appendix2 - …

### 5.3 …