

**SOFTWARE REQUIREMENT SPECIFICATION**

**<<Project Name – Project Code>>**

– Hanoi, August 2019 –

# Record of Changes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | A\* M, D | In charge | Change Description |
| V1.0 | 15/2 |  |  |  |
| V1.0 | 16/2 |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

\*A - Added M - Modified D - Deleted

Table of Contents

[Record of Changes 2](#_Toc135038521)

[I. Product Overview 5](#_Toc135038522)

[1.1. Product Vision 5](#_Toc135038523)

[1.2. Product Context 5](#_Toc135038524)

[1.3. Major Features 6](#_Toc135038525)

[1.4. User Requirements 6](#_Toc135038526)

[1.4.1 Actors List 6](#_Toc135038527)

[1.4.2 Use Cases 7](#_Toc135038528)

[1.5. Assumptions & Dependencies 8](#_Toc135038529)

[1.6. Limitations and Exclusions 8](#_Toc135038530)

[1.7. Business Rules 8](#_Toc135038531)

[II. Use Case Specifications 9](#_Toc135038532)

[2.1. Order Meals Feature 9](#_Toc135038533)

[2.1.1 Order a Meal 9](#_Toc135038534)

[2.1.2 Change Meal Order 11](#_Toc135038535)

[2.1.3 Cancel Meal Order 11](#_Toc135038536)

[2.2. Meal Subscriptions Feature 11](#_Toc135038537)

[2.2.1 Register for Payroll Deduction 11](#_Toc135038538)

[2.2.2 <<Next Use Case Name..>> 12](#_Toc135038539)

[2.3. <<Next Feature Name..>> 12](#_Toc135038540)

[2.3.1 <<Use Case Name>> 12](#_Toc135038541)

[2.3.2 … 12](#_Toc135038542)

[III. Functional Requirements 13](#_Toc135038543)

[3.1. System Functional Overview 13](#_Toc135038544)

[3.1.1 Screens Flow 13](#_Toc135038545)

[3.1.2 Screen Descriptions 13](#_Toc135038546)

[3.1.3 Screen Authorization 13](#_Toc135038547)

[3.1.4 n-Screen Functions 14](#_Toc135038548)

[3.2. Data model 14](#_Toc135038549)

[3.2.1 Entity class diagram 14](#_Toc135038550)

[3.2.2 Entity Description 14](#_Toc135038551)

[3.3. <<Feature Name 1>> 15](#_Toc135038552)

[3.3.1 <<Function Name 1>> 15](#_Toc135038553)

[3.2 <<Function Name 2>> 15](#_Toc135038554)

[3.3 <<Feature Name 2>> 15](#_Toc135038555)

[IV. Non-Functional Requirements 16](#_Toc135038556)

[4.1. External Interfaces 16](#_Toc135038557)

[4.2. Quality Attributes 16](#_Toc135038558)

[4.2.1 Usability 16](#_Toc135038559)

[4.2.2 Reliability 16](#_Toc135038560)

[4.2.3 Performance 16](#_Toc135038561)

[4.2.4 … 16](#_Toc135038562)

# I. Product Overview

## 1.1. Product Vision

*[Write a concise vision statement that summarizes the purpose and intent of the new product and describes what the world will be like when it includes the product. The vision statement should reflect a balanced view that will satisfy the needs of diverse customers as well as those of the developing organization. It may be somewhat idealistic, but it should be grounded in the realities of existing or anticipated customer markets, enterprise architectures, organizational strategic directions, and cost and resource limitations]*

<<Sample

For employees who want to order meals from the company cafeteria or from local restaurants on-line, the Cafeteria Ordering System is an Internet-based and smartphone-enabled application that will accept individual or group meal orders, process payments, and trigger delivery of the prepared meals to a designated location on the Process Impact campus. Unlike the current telephone and manual ordering processes, employees who use the Cafeteria Ordering System will not have to go to the cafeteria to get their meals, which will save them time and will increase the food choices available to them.

>>

## 1.2. Product Context

*[Gives the* ***context diagram****, describe the diagram elements (might be in the form of the relevant* ***events list****) here. 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]*

<<Sample

The connections between the COS with the external entities are as described in the below diagram



In which:

* Element description 1
* Element description 2
* …

>>

## 1.3. Major Features

*[Include a numbered list of the major features of the new product, emphasizing those features that distinguish it from previous or competing products. Specific user requirements and functional requirements may be traced back to these features]*

<<Sample:

FE-01: Order and pay for meals from the cafeteria menu to be picked up or delivered.

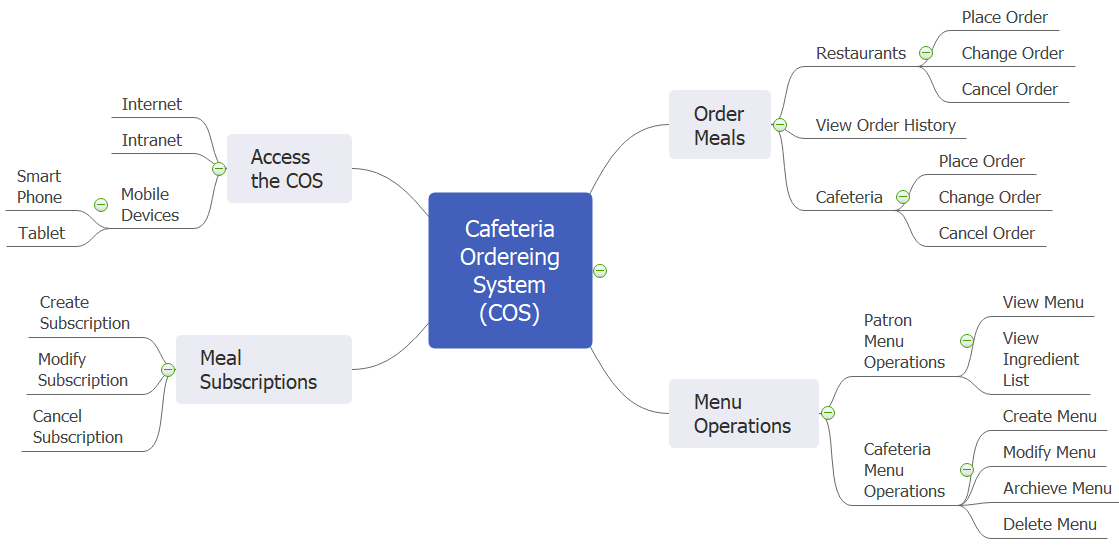
FE-02: Order and pay for meals from local restaurants to be delivered.

FE-03: Create, view, modify, delete, and archive cafeteria menus.

FE-04: View ingredient lists and nutritional information for cafeteria menu items.

FE-05: Create, view, modify, and cancel meal subscriptions for standing or recurring meal orders, or for daily special meals.

FE-06: Provide system access through corporate intranet, smartphone, tablet, and outside Internet access by authorized employees



>>

## 1.4. User Requirements

### 1.4.1 Actors List

*[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 | … |  |

### 1.4.2 Use Cases

*[Give the use case diagram(s) and the description on each use case here]*

<<Sample

Diagram 1

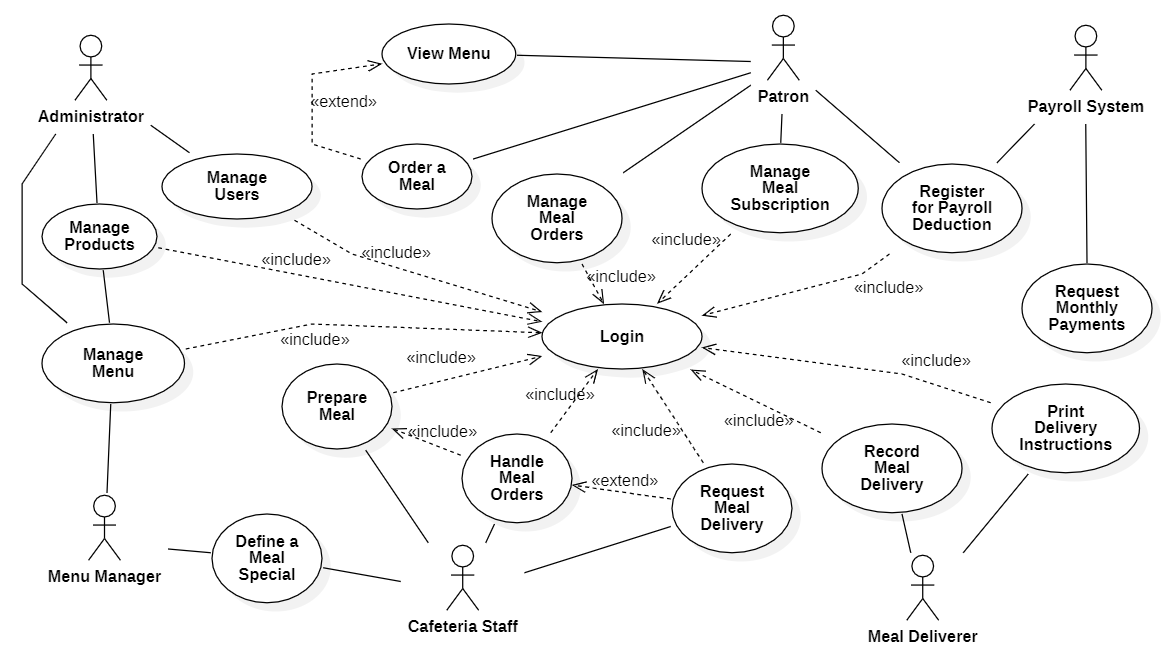


Diagram 2

…

In which

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Feature** | **Use Case** | **Description** |
| UC-01 | Order Meals | Order a Meal | <<Use case description>> |
| 02 | Order Meals | Change Meal Order | <<Use case description>> |
| 03 | Order Meals | Cancel Meal Order | <<Use case description>> |
| 04 | Meal Subscriptions | Register for Payroll Deduction | <<Use case description>> |
| 05 | Meal Subscriptions | Unregister for Payroll Deduction | <<Use case description>> |
| 06 | Meal Subscriptions | Manage Meal Subscription | <<Use case description>> |
| 07 | Menu Operations | View Menu | <<Use case description>> |
| 08 | Menu Operations | Create a Menu | <<Use case description>> |
| 09 | Menu Operations | Modify a Menu | <<Use case description>> |
| 10 | Menu Operations | Delete a Menu | <<Use case description>> |
| 11 | Menu Operations | Archive Menus | <<Use case description>> |
| 12 | Menu Operations | Define a Meal Special | <<Use case description>> |
| 13 | Meal Preparations | Prepare Meal | <<Use case description>> |
| 14 | Meal Preparations | Generate a Payment Request | <<Use case description>> |
| 15 | Meal Preparations | Request Meal Delivery | <<Use case description>> |
| 16 | Meal Preparations | Generate System Usage Reports | <<Use case description>> |
| 17 | Meal Delivery | Record Meal Delivery | <<Use case description>> |
| 18 | Meal Delivery | Print Delivery Instructions | <<Use case description>> |

>>

## 1.5. Assumptions & Dependencies

*<<Record any assumptions that were made when conceiving the project and writing this vision and scope document. Note any major dependencies the project must rely upon for success, such as specific technologies, third-party vendors, development partners, or other business relationships.>>*

<<Sample:

AS-1: Systems with appropriate user interfaces will be available for cafeteria employees to process the expected volume of meals ordered.

AS-2: Cafeteria staff and vehicles will be available to deliver all meals for specified delivery time slots within 15 minutes of the requested delivery time.

DE-1: If a restaurant has its own on-line ordering system, the Cafeteria Ordering System must be able to communicate with it bi-directionally.

>>

## 1.6. Limitations and Exclusions

*[Identify any product features or characteristics that a stakeholder might anticipate, but which are not planned to be included in the new product]*

## 1.7. Business Rules

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

<<Sample

|  |  |  |
| --- | --- | --- |
| **ID** | **Category** | **Rule Definition** |
| BR-01 | Constraints | Delivery time windows are 15 minutes, beginning on each quarter hour. |
| BR-02 | Constraints | Deliveries must be completed between 10:00 A.M. and 2:00 P.M. local time, inclusive. |
| BR-03 | Facts | All meals in a single order must be delivered to the same location. |
| BR-04 | Facts | All meals in a single order must be paid for by using the same payment method. |
| BR-11 | Constraints | If an order is to be delivered, the patron must pay by payroll deduction. |
| BR-12 | Computations | 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. |

>>

# II. Use Case Specifications

## 2.1. Order Meals Feature

### 2.1.1 Order a Meal

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-01 Order a Meal** | | |
| Created By: | Prithvi Raj | Date Created: | 10/4/13 |
| Primary Actor: | Patron | Secondary Actors: | Cafeteria Inventory System |
| Description: | A Patron accesses the Cafeteria Ordering System from the corporate intranet or from home, views the menu for a specific date if desired, selects food items, and places an order for a meal to be delivered to a specified location within a specified 15-minute time window. | | |
| Trigger: | A Patron indicates that he wants to order a meal | | |
| Preconditions: | PRE-1. Patron is logged into COS.  PRE-2. Patron is registered for meal payments by payroll deduction. | | |
| Postconditions: | POST-1. Meal order is stored in COS with a status of “Accepted”.  POST-2. Inventory of available food items is updated to reflect items in this order.  POST-3. Remaining delivery capacity for the requested time window is updated. | | |
| Normal Flow: | **1.0 Order a Single Meal**   1. Patron asks to view menu for a specific date. (see 1.0.E1, 1.0.E2) 2. COS displays menu of available food items and the daily special. 3. Patron selects one or more food items from menu. (see 1.1) 4. Patron indicates that meal order is complete. (see 1.2) 5. COS displays ordered menu items, individual prices, and total price, including taxes and delivery charge. 6. Patron either confirms meal order (continue normal flow) or requests to modify meal order (return to step 2). 7. COS displays available delivery times for the delivery date. 8. Patron selects a delivery time and specifies the delivery location. 9. Patron specifies payment method. 10. COS confirms acceptance of the order. 11. COS sends Patron an email message confirming order details, price, and delivery instructions. 12. COS stores order, sends food item information to Cafeteria Inventory System, and updates available delivery times. | | |
| Alternative Flows: | **1.1 Order multiple identical meals**   1. Patron requests a specified number of identical meals. (see 1.1.E1) 2. Return to step 4 of normal flow.   **1.2 Order multiple meals**   1. Patron asks to order another meal. 2. Return to step 1 of normal flow. | | |
| Exceptions: | **1.0.E1 Requested date is today and current time is after today’s order cutoff time**  1. COS informs Patron that it’s too late to place an order for today.  2a. If Patron cancels the meal ordering process, then COS terminates use case.  2b. Else if Patron requests another date, then COS restarts use case.  **1.0.E2 No delivery times left**  1. COS informs Patron that no delivery times are available for the meal date.  2a. If Patron cancels the meal ordering process, then COS terminates use case.  2b. Else if Patron requests to pick the order up at the cafeteria, then continue with normal flow, but skip steps 7 and 8.  **1.1.E1 Insufficient inventory to fulfill multiple meal order**  1. COS informs Patron of the maximum number of identical meals he can order, based on current available inventory.  2a. If Patron modifies number of meals ordered, then Return to step 4 of normal flow.  2b. Else if Patron cancels the meal ordering process, then COS terminates use case. | | |
| Priority: | High | | |
| Frequency of Use: | Approximately 300 users, average of one usage per day. Peak usage load for this use case is between 9:00 A.M. and 10:00 A.M. local time. | | |
| Business Rules: | BR-1, BR-2, BR-3, BR-4, BR-11, BR-12, BR-33 | | |
| Other Information: | 1. Patron shall be able to cancel the meal ordering process at any time prior to confirming it. 2. Patron shall be able to view all meals he ordered within the previous six months and repeat one of those meals as the new order, provided that all food items are available on the menu for the requested delivery date. (Priority = M) 3. The default date is the current date if the Patron is using the system before today’s order cutoff time. Otherwise, the default date is the next day that the cafeteria is open. | | |
| Assumptions: | Assume that 15 percent of Patrons will order the daily special (source: previous 6 months of cafeteria data). | | |

### 2.1.2 Change Meal Order

*<<Use Case Description in the same format as above>>*

### 2.1.3 Cancel Meal Order

*<<Use Case Description in the same format as above>>*

## 2.2. Meal Subscriptions Feature

### 2.2.1 Register for Payroll Deduction

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-05 Register for Payroll Deduction** | | |
| Created By: | Nancy Anderson | Date Created: | 9/15/13 |
| Primary Actor: | Patron | Secondary Actors: | Payroll System |
| Description: | Cafeteria patrons who use the COS and have meals delivered must be registered for payroll deduction. For noncash purchases made through the COS, the cafeteria will issue a payment request to the Payroll System, which will deduct the meal costs from the next scheduled employee payday direct deposit. | | |
| Trigger: | Patron requests to register for payroll deduction, or Patron says yes when COS asks if he wants to register | | |
| Preconditions: | PRE-1. Patron is logged into COS. | | |
| Postconditions: | POST-2. Patron is registered for payroll deduction. | | |
| Normal Flow: | **5.0 Register for Payroll Deduction**   1. COS asks Payroll System if Patron is eligible to register for payroll deduction. 2. Payroll System confirms that Patron is eligible to register for payroll deduction. 3. COS asks Patron to confirm his desire to register for payroll deduction. 4. If so, COS asks Payroll System to establish payroll deduction for Patron. 5. Payroll System confirms that payroll deduction is established. 6. COS informs Patron that payroll deduction is established. | | |
| Alternative Flows: | None | | |
| Exceptions: | 5.0.E1 Patron is not eligible for payroll deduction  5.0.E2 Patron is already enrolled for payroll deduction | | |
| Priority: | High | | |
| Frequency of Use: |  | | |
| Business Rules: | BR-86 and BR-88 govern an employee’s eligibility to enroll for payroll deduction. | | |
| Other Information: | Expect high frequency of executing this use case within first 2 weeks after system is released. | | |
| Assumptions: |  | | |

### 2.2.2 <<Next Use Case Name..>>

*<<Use Case Description in the same format as above>>*

## 2.3. <<Next Feature Name..>>

### 2.3.1 <<Use Case Name>>

*<<Use Case Description in the same format as above>>*

### 2.3.2 …

# III. 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 Role-Name1, Role-Name2,… 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 |  |  |  |  |
| … |  |  |  |  |

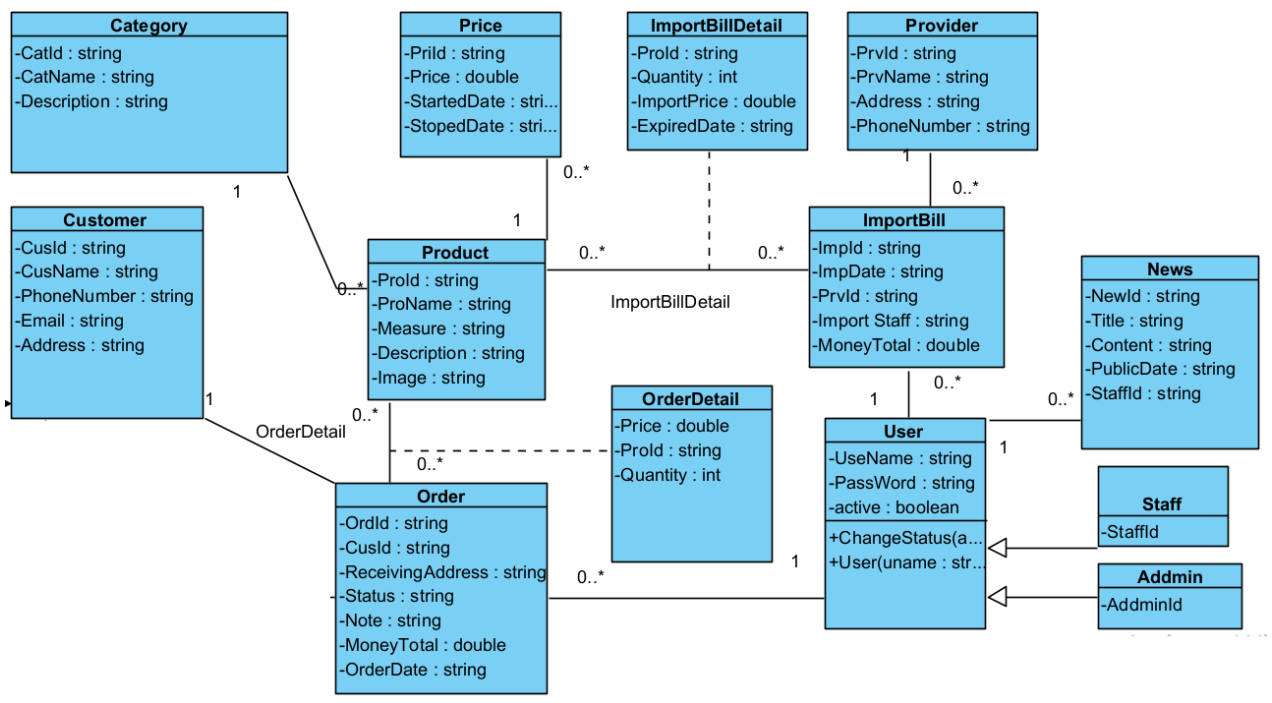
### 3.1.4 n-Screen Functions

*[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 | … |  |  |

### 3.2. Data model

### 3.2.1 Entity class diagram



### 3.2.2 Entity Description

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

## 3.3. <<Feature Name 1>>

### 3.3.1 <<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 <<Function Name 2>>

…

### 3.3 <<Feature Name 2>>

…

# IV. 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/system elements.]*

## 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]*

### 4.2.1 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]*

### 4.2.2 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.]*

### 4.2.3 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*

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

### 4.2.4 …