Software Requirements Specification

for

Smart App

Release 1.0

Version 1.0 approved

Prepared by Group 5

Process Impact

July 18, 2021

Table of Contents

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Project Scope and Product Features 1

1.4 References 1

2. Overall Description 1

2.1 Product Perspective 1

2.2 User Classes and Characteristics 2

2.3 Operating Environment 3

2.4 Design and Implementation Constraints 3

2.5 Assumptions and Dependencies 3

3. System Features 3

3.1 Order Meals from Cafeteria 3

3.1.1 Description 3

3.1.2 Functional Requirements 3

3.2 Order Meals from Restaurants 5

3.3 Create, View, Modify, and Delete Meal Subscriptions 5

3.4 Create, View, Modify, and Delete Cafeteria Menus 5

4. Data Requirements 6

4.1 Logical Data Model 6

4.2 Data Dictionary 6

4.3 Reports 9

4.3.1 Ordered Meal History Report 9

4.4 Data Integrity, Retention, and Disposal 9

5. External Interface Requirements 10

5.1 User Interfaces 10

5.2 Software Interfaces 10

5.3 Hardware Interfaces 10

5.4 Communications Interfaces 10

6. Quality Attributes 11

6.1 Usability Requirements 11

6.2 Performance Requirements 11

6.3 Security Requirements 11

6.4 Safety Requirements 11

6.5 Availability Requirements 11

6.6 Robustness Requirements 11

Appendix A: Analysis Models 12

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Tăng Minh Tín | 7/15/21 | Initial draft | 1.0 draft 1 |
| Trương Nhật Nam | 7/16/21 | Adding and editing content | 1.0 draft 2 |
| Phạm Quốc Nghị | 7/18/21 | Baseline following changes after inspection | 1.0 approved |

# Introduction

## Purpose

This SRS describes the functional and non-functional requirements for software release 1.0 of the Smart App. This document is intended to be used by the members of the project team who will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are committed for release 1.0.

## Document Conventions

No special typographical conventions are used in this SRS.

## Project Scope and Product Features

The Smart App allows users to use their smartphone to view detailed information about the phone such as description, price, brand, configuration and has pictures taken from the product for users to choose from. The most suitable product for us. With user-friendly features when using the app such as viewing product information, adding products to the cart, choosing the quantity, which color you want to buy, making payment...

## References

None

# Overall Description

## Product Perspective

The Smart App system is a new software system that replaces the manual processes and uses the current phone to order and payment products. The context diagram in Figure 1 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 stores and to credit and debit card authorization services.

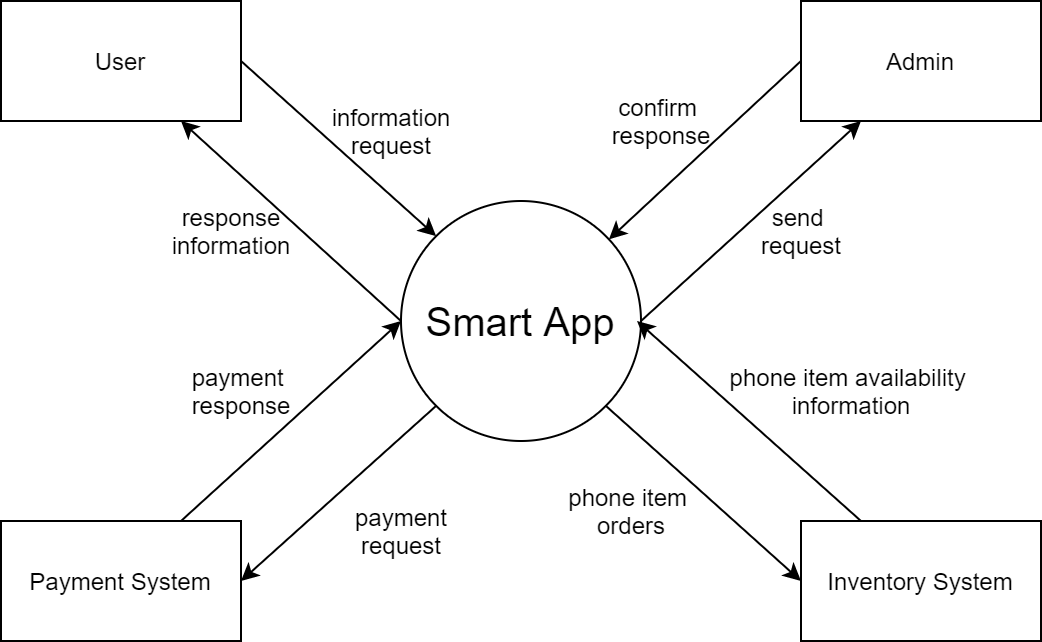


Figure 1. Context diagram for release 1.0 of the Cafeteria Ordering System.

## User Classes and Characteristics

|  |  |
| --- | --- |
| User | A user is someone who wants to order and pay for the company's phone products. There are about 1000 potential customers, of which about 600 customers are expected to use Smart App on average 5 times per week. Patrons will sometimes order many products or refer friends and relatives to the Smart App system. It is estimated that about 70% of orders will be made by the company's app, 30% of orders will be placed in the store. |
| Admin | Admin is the person who will receive information about orders from Smart App, to view and confirm these orders. Most admins will be trained on how to use hardware and software for Smart App. |

## Operating Environment

OE-1: The Smart App shall operate correctly with the following Android versions 21 through 30.

OE-2: The Smart App Server shall operate on a server running the current corporate-approved versions of Red Hat Linux and Apache HTTP Server.

## Design and Implementation Constraints

CO-1: The system’s design, code, and maintenance documents will be attached when submitting this file.

CO-2: The system shall use the current corporate standard MySQL database engine.

## Assumptions and Dependencies

AS-1: Smart App's server must work 24/7 to be ready to serve customers when accessing the system at any time.

DE-1: The operation of the Smart App depends on the changes made in the payment system to accept payment requests for mobile products when placing an order with the Smart App.

DE-2: The operation of the Smart App depends on the changes made in the inventory system to update the stock status of the products when the Smart App allows the acceptance of orders from the customer.

# System Features

## Order Meals from Cafeteria

### Description

A cafeteria Patron whose identity has been verified may order meals either to be delivered to a specified company location or to be picked up in the cafeteria. A Patron may cancel or change a meal order if it has not yet been prepared. Priority = High.

### Functional Requirements

|  |
| --- |
| **Order.Place: Placing a meal order**  .Register: The COS shall confirm that the Patron is registered for payroll deduction.  .No: If the Patron is not registered for payroll deduction, the COS shall give the Patron options to register now and continue placing an order, to place an order for pickup in the cafeteria (but not for delivery), or to exit.  .Date: The COS shall prompt the Patron for the meal date (see BR-8).  .Cutoff: If the meal date is the current date and the current time is after the order cutoff time, the COS shall inform the patron that it’s too late to place an order for today. The Patron can either change the meal date or cancel the order. |
| **Order.Deliver: Delivery or pickup**  .Select: The Patron shall specify whether the order is to be picked up or delivered.  .Location: If the order is to be delivered and there are still available delivery times for the meal date, the Patron shall provide a valid delivery location.  .Notimes: The COS shall notify the Patron if there are no available delivery times for the meal date. The Patron shall either cancel the order or indicate that he will pick up the order in the cafeteria.  .Times: The COS shall display the remaining available delivery times for the meal date. The COS shall allow the Patron to request one of the delivery times shown, to change the order to be picked up in the cafeteria, or to cancel the order. |
| **Order.Menu: Viewing a menu**  .Date: The COS shall display a menu for the date that the Patron specified.  .Available: The menu for the specified date shall display only those food items for which at least one unit is available in the cafeteria’s inventory and which may be delivered. |
| **Order.Units: Ordering multiple meals and multiple food items**  .Multiple: The COS shall permit the user to order multiple identical meals, up to the fewest available units of any menu item in the order.  .TooMany: If the Patron orders more units of a menu item than are presently in the cafeteria’s inventory, the COS shall inform the Patron of the maximum number of units of that food item that he can order. |
| **Order.Confirm: Confirming an order**  .Display: When the Patron indicates that he does not wish to order any more food items, the COS shall display the food items ordered, the individual food item prices, and the payment amount calculated per BR-12.  .Prompt: The COS shall prompt the Patron to confirm the meal order.  .Response: The Patron can confirm, edit, or cancel the order.  .More: The COS shall let the Patron order additional meals for the same or for a different date. BR-3 and BR-4 pertain to multiple meals in a single order. |
| **Order.Pay: Meal order payment**  .Method: When the Patron indicates that he is done placing orders, the COS shall ask the user to select a payment method.  .Deliver: See BR-11.  .Pickup: If the meal is to be picked up in the cafeteria, the Patron shall choose to pay by payroll deduction or by cash at the time of pickup.  .Deduct: If the Patron selected payroll deduction, the COS shall issue a payment request to the Payroll System.  .OK: If the payment request is accepted, the COS shall display a message confirming acceptance of the order with a transaction number.  .NG: If the payment request is rejected, the COS shall display the reason for the rejection. The Patron shall either cancel the order, or change the payment method to cash and request to pick up the order at the cafeteria. |
| **Order.Done: When the Patron has confirmed the order, the COS shall do the following as a single transaction.**  .Store: Assign the next available meal order number to the meal and store the meal order with a status of “Accepted.”  .Inventory: Send a message to the Cafeteria Inventory System with the number of units of each food item in the order.  .Menu: Update the menu for the current order’s order date to reflect any items that are now out of stock in the cafeteria inventory.  .Times: Update the remaining available delivery times for the date of this order.  .Patron: Send an email message or text message to the Patron with the meal order and meal payment information.  .Cafeteria: Send an email message to the Cafeteria Staff with the meal order information.  .Failure: If any step of Order.Done fails, the COS shall roll back the transaction and notify the user that the order was unsuccessful, along with the reason for failure. |

[Note: Functional requirements for reordering a meal and for changing and canceling meal orders are not provided in this example.]

## Order Meals from Restaurants

[Details are not provided in this example. Quite a lot of the functionality described under 3.1 Order Meals from Cafeteria could likely be reused, so this section should just specify the additional functionality that addresses the restaurant interface.]

## Create, View, Modify, and Delete Meal Subscriptions

[Details are not provided in this example.]

## Create, View, Modify, and Delete Cafeteria Menus

[Details are not provided in this example.]

# Data Requirements

## Logical Data Model

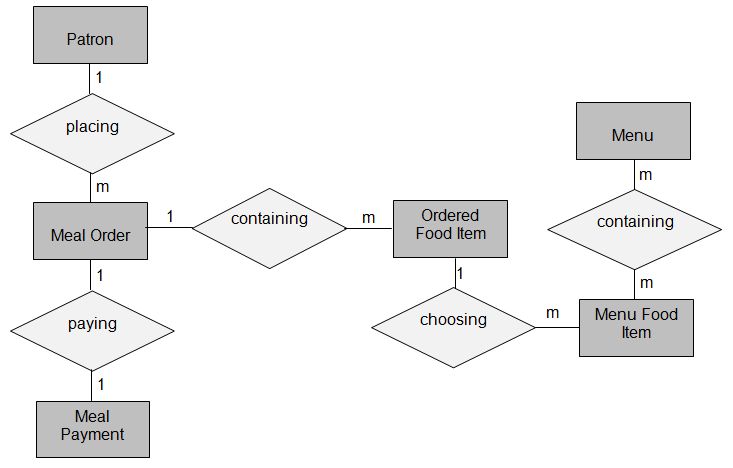


Figure . Partial data model for release 1.0 of the Cafeteria Ordering System.

## Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Element | Description | Composition or Data Type | Length | Values |
| delivery instruction | where and to whom a meal is to be delivered, if it isn't being picked up in the cafeteria | patron name  +patron phone number  +meal date  +delivery location  +delivery time window |  |  |
| delivery location | building and room to which an ordered meal is to be delivered | alphanumeric | 50 | hyphens and commas permitted |
| delivery time window | beginning time of a 15-minute range on the meal date during which an ordered meal is to be delivered; | time | hh:mm | local time; hh = 0-23 inclusive; mm = 00, 15, 30, or 45 |
| employee ID | company ID number of the employee who placed a meal order | integer | 6 |  |
| food item description | ext description of a food item on a menu | alphabetic | 100 |  |
| food item price | pre-tax cost of a single unit of a menu food item | numeric, dollars and cents | dd.cc |  |
| meal date | the date the meal is to be delivered or picked up | date, MM/DD/YYYY | 10 | default = current date if the current time is before the order cutoff time, else the next day; cannot be prior to current date |
| meal order | details about a meal a Patron ordered | meal order number  + order date  + meal date  + 1:m{ordered food item}  + delivery instruction  + meal order status |  |  |
| meal order number | unique ID that COS assigns to each accepted meal order | integer | 7 | initial value is 1 |
| meal order status | status of a meal order that a Patron initiated | alphabetic | 16 | incomplete, accepted, prepared, pending delivery, delivered, canceled |
| meal payment | information about a payment COS accepted for a meal | payment amount  + payment method  + transaction number |  |  |
| menu | list of food items available for purchase on a specific date | menu date  + 1:m{menu food item} |  |  |
| menu date | the date for which a specific menu is available | date, MM/DD/YYYY | 10 |  |
| menu food item | description of a menu item | food item description  + food item price |  |  |
| order cutoff time | the time of day before which all meal orders for that date must be placed | time, HH:MM | 5 |  |
| order date | the date on which a patron placed a meal order | date, MM/DD/YYYY | 10 |  |
| ordered food item | one menu food item that a Patron requested as part of a meal order | menu food item  + quantity ordered |  |  |
| patron | a Process Impact employee who is authorized to order a meal | patron name  + employee ID  + patron phone number  + patron location  + patron email |  |  |
| patron email | email address of the employee who placed a meal order | alphanumeric | 50 |  |
| patron location | building and room numbers of the employee who placed a meal order | alphanumeric | 50 | hyphens and commas permitted |
| patron name | name of the employee who placed a meal order | alphabetic | 30 |  |
| patron phone number | telephone number of the employee who placed a meal order | AAA-EEE-NNNN xXXXX for area code (A), exchange (E), number (N), and extension (X) | 18 |  |
| payment amount | total price of an order in dollars and cents, calculated per BR-12 | numeric, dollars and cents | dddd.cc |  |
| payment method | how the Patron is paying for a meal he ordered | alphabetic | 16 | payroll deduction, cash, credit card, debit card |
| quantity ordered | the number of units of each food item that the Patron is ordering in a single meal order | integer | 4 | default = 1; maximum = quantity presently in inventory |
| transaction number | unique sequence number that COS assigns to each payment transaction | integer | 12 |  |

## Reports

### Ordered Meal History Report

|  |  |
| --- | --- |
| Report ID: | COS-RPT-1 |
| Report Title: | Ordered Meal History |
| Report Purpose: | Patron wants to see a list of all meals that he had previously ordered from the Process Impact cafeteria or local restaurants over a specified time period up to six months prior to the current date, so he can reorder a particular meal he liked. |
| Priority: | Medium |
| Report Users: | Patrons |
| Data Sources: | Database of previously placed meal orders |
| Frequency and Disposition; | Report is generated on demand by a Patron. Data in the report is static. Report is displayed on user's web browser screen on a computer, tablet, or smartphone. It can be printed if the display device permits printing. |
| Latency: | Complete report must be displayed to Patron within 3 seconds after it is requested. |
| Visual Layout: | Landscape mode |
| Header and Footer: | Report header shall contain the report title, Patron's name, and date range specified. If printed, report footer shall show the page number. |
| Report Body: | Fields shown and column headings:   * Order Number * Meal Date * Ordered From ("Cafeteria" or restaurant name) * Items ordered (list all items in the meal order, their quantity, and their prices) * Total Food Price * Tax * Delivery Charge * Total Price (sum of food item prices, tax, and delivery charge)   Selection Criteria: date range specified by Patron, inclusive of end points  Sort Criteria: reverse chronological order |
| End-of-Report Indicator: | None |
| Interactivity: | Patron can drill down to see ingredients and nutritional information for each item in the order |
| Security Access Restrictions: | A Patron may retrieve only his own meal order history |

[Note: Other COS reports are not provided in this example.]

## Data Integrity, Retention, and Disposal

DI-1: The COS shall retain Individual Patron meal orders for 6 months following the meal's delivery date.

DI-2: The COS shall retain menus for one year following the menu date.

# External Interface Requirements

## User Interfaces

UI-1: The Smart App screen displays shall conform to the Process Impact Mobile Application User Interface Standard.

UI-2: The system provides a full range of functions for users to choose and buy products from the app.

UI-3: Interface beautiful and easy to use there are many new and different phone models for people to choose and buy them. Software Interfaces.

## Software Interfaces

SI-1: Product sales system.

SI-1.1: Simple sales interface easy to use.

SI-1.2: The system will continuously update new items and best-selling items to let everyone know and refer.

SI-1.3: The system will automatically notify if the product is out of stock by clicking to view product information when the user clicks on it.

SI-2: Calculation system.

SI-2.1: Every month the system will automatically calculate the total sales and total phones sold on the app.

SI-2.2: The system will check the list of best-selling products and the most viewed and liked products.

SI-2.3: The system will check in 1 month how many downloads and account registrations on the app.

SI-2.4: Calculate the total number of shoppers on the app.

## Hardware Interfaces

No hardware interfaces have been identified.

## Communications Interfaces

CI-1: When the user has successfully placed an order, on the app, someone will call the phone number recorded in the order to confirm the acceptance of the order, fully check the buyer's information and delivery method.

CI-2: App will take over calls or messages from users when there is any problem related to purchase and delivery.

# Quality Attributes

## Usability Requirements

USE-1: The system will allow the user to return the item when the product received it is defective or any damage.

USE-2: 95% of new users can successfully purchase without errors on the first try.

## Performance Requirements

PER-1: The system will accommodate a total of 1000 shoppers at the same time without any problems on the app.

PER-2: The app is suitable for many phone lines and phones with andorid 3 and above can download and use.

PER-3: The system will record all notifications and messages of the user within 24 hours and will respond to the message as soon as possible to the customer.

## Security Requirements

SEC-1: All network transactions that involve financial information or personally identifiable information shall be encrypted.

SEC-2: Users must login to the system to purchase products on the app, except for viewing products.

SEC-3: The system will allow the Client to view only the orders they have placed.

## Safety Requirements

SAF-1: Users can view any product on the app based on a list of products and will display which items are out of stock, in stock and on promotion for easy selection.

## Availability Requirements

AVL-1: App shall be available at least 98% of the time between 5:00 A.M. and midnight local time and at least 90% of the time between midnight and 5:00 A.M. local time, excluding scheduled maintenance windows.

## Robustness Requirements

ROB-1: If the connection between the user and the app is broken prior to a new order being either confirmed or terminated, the app shall enable the user to recover an incomplete order and continue working on it.