SOFTWARE REQUIREMENTS SPECIFICATION

for

< Cascade - Contactless Cafeteria Ordering System >

Version 1.0 approved.

Prepared by

19Z210 - Deepti Ravi Kumar

21.08.2021

TABLE OF CONTENTS

Revision History

Date	Description	Author	Comments
21.08.2021	Initial Draft	Deepti Ravi Kumar	Initial Draft of the document

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to describe the software specification requirements for release 1.0 of the Cafeteria Ordering System (COS). This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are high priority and committed for release 1.0. The document contains the functional behavior and non-functional requirements of the system project. The document also contains the guidelines for system engineers and programmers to start working and accomplish the project on a given time frame.

1.2 Document Conventions

The format of this Software Requirement Specifications for Canteen Ordering System is simple. In general, this document follows the IEEE formatting requirements. Bold face and indentation are used on general topics and on specific points of interest. The rest of the document will be written using the standard font, Arial italics font size 11, or 12 throughout the document for text. Document text should be single spaced and maintain the 1" margins found in this template. For Section and Subsection titles please follow the template.

1.3 Intended Audience and Reading Suggestions

The intended readers of this document are the developers of the application, testers, cafeteria staff, managers, and coordinators.
In case of any suggested changes on the requirements listed on this document should be included in the last version of it so it can be a reference to developing and validating teams.

1.4 Product Scope

The software product is named Cafeteria Ordering System. Keeping track of the cafeteria menu, attendance, and consumption is a difficult task. Manual and paper-based procedures are cumbersome and prone to errors, resulting in inaccuracy and waste of time and resources. Therefore, an automated canteen management system is essential.

This project attempts to build a canteen management system that analyzes food consumption and waste by item, offers customers a convenient mobile ordering system, and includes an AI-based attendance system. Meals for breakfast, lunch, and dinner, as well as special days and events, may all be planned. This project enables menu item monitoring, quick transactions, and the avoidance of accounting errors. Customers can also provide feedback using the feedback function, which allows the canteen manager to analyze and make necessary adjustments to the system.

References

1.4.1 IEEE 830-1998 standard for writing SRS document.

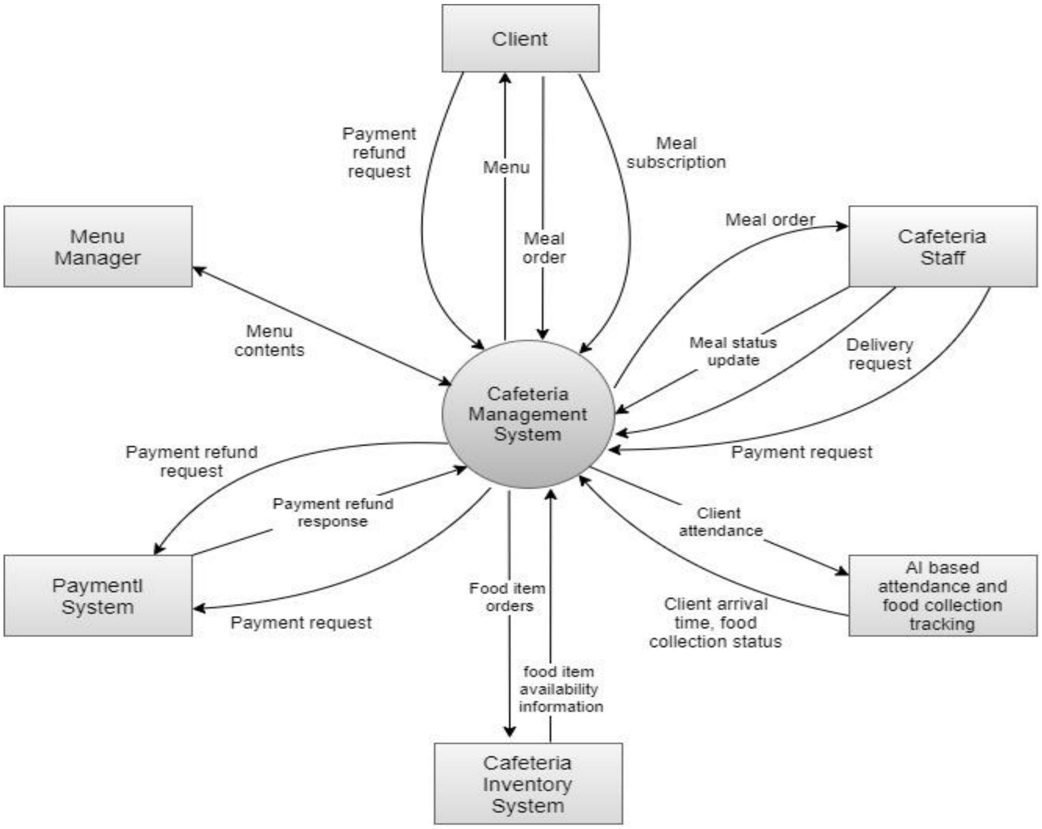
1.4.2 IEE Std 1233, 1998 Edition, IEEE Guide for Developing System Requirements Specifications

2 OVERALL DESCRIPTION

2.1 Product Perspective

The Cafeteria Ordering System assists the canteen manager in managing the canteen more efficiently and effectively by automating meal ordering, billing, and inventory control. The AI-based attendance and food collection tracking can help to ensure effective food distribution, as well as penalty and refund decisions.

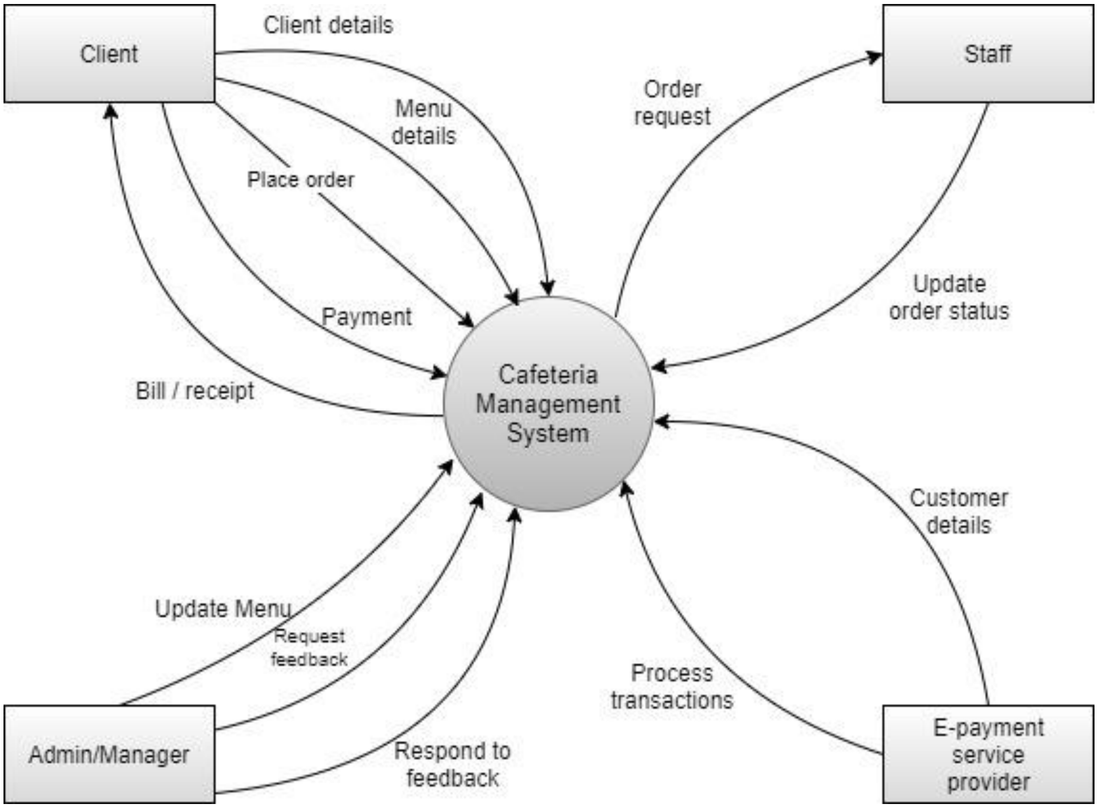
The system analyses transactions and saves the generated data, which the manager may use to produce reports and make informed business choices for the canteen. The management, for example, can determine whether extra canteen staff is necessary based on the number of customers for a specific time frame. In addition, one can easily calculate daily expenses and profits. The system will evolve over several releases, eventually connecting to the college's online ordering system and credit and debit card authorization services.



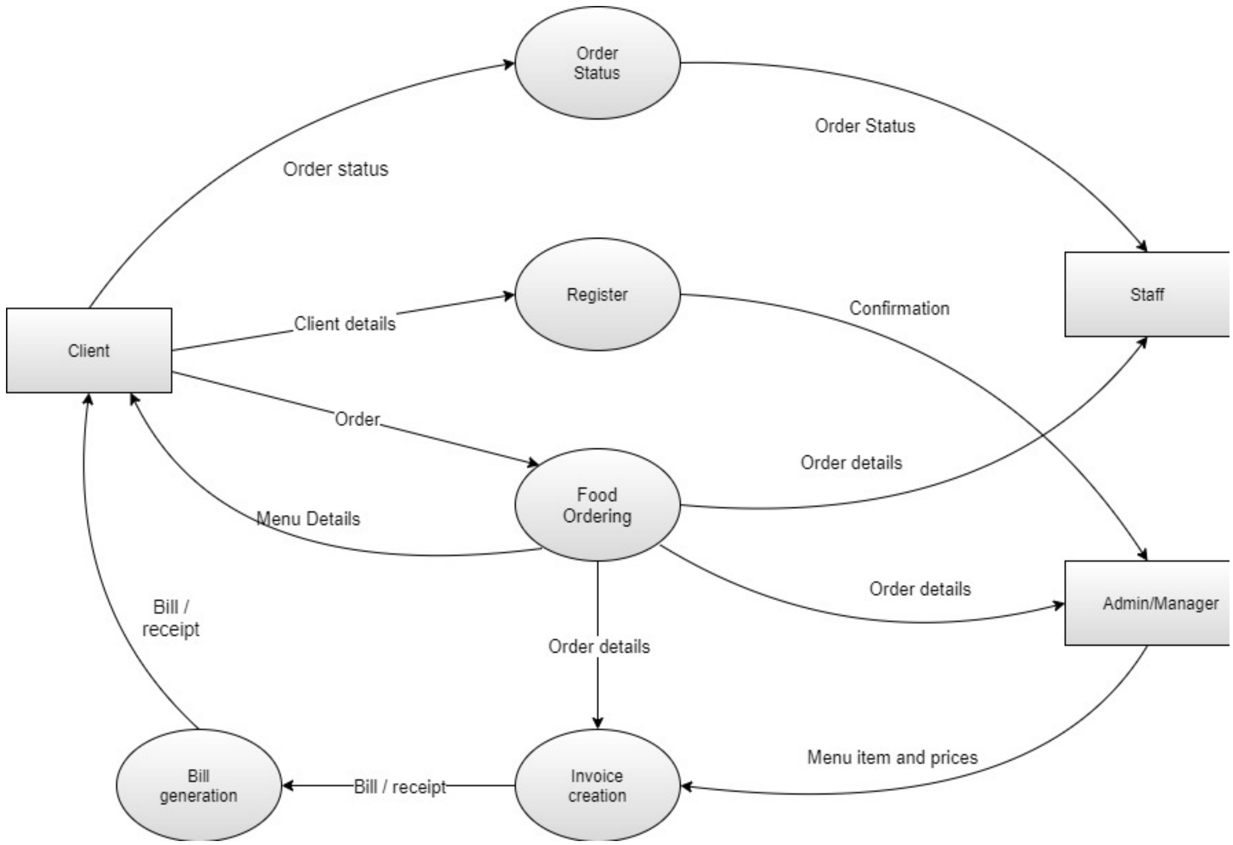
2.2 Product Functions

All of the functions will be performed in the order given below,

- *Food order via application*
- *Confirm order*
- *Online Payment*
- *Serve food*
- *Available goods*
- *Required goods*
- *Customer information*
- *Customer review*



Data Flow Diagram – Level 0



Data Flow Diagram – Level 1

2.3 User Classes and Characteristics

Three active actors and a cooperative system are involved in the project.

2.3.1 Customer Class

Customers can access the system using their smartphones to place orders. The customer accesses the online payment portal to complete the payment transaction. After a customer has paid for his order, the canteen staff checks the order and sends a confirmation. When the food is ready, the client will receive a notification. The customer can now go and collect the food at the counter.

2.3.2 Cafeteria Manager

In addition to adding and removing items from the menu, the admin can also edit the price, count the total earnings and expenditures, and ask customers for feedback.

2.3.3 Cafeteria Staff

Admin's job is to manage the inventory and other information related to menu. Moreover, he gets a notification whenever a particular order is complete, or some customer asks for help through the system

2.4 Operating Environment

2.4.1 Hardware environment

Hardware requirements include Jetson Nano or Raspberry Pi and Pi-camera for the AI-based attendance system and order collection tracking.

2.4.2 Software environment

The application will utilize Linux as the Operating system and Google Cloud computing for Firebase utilization.

2.5 Design and Implementation Constraints

i. The core system and its user interfaces are compatible with smartphones.
ii. The number of hardware component sets required is dependent on the frame of focus of each camera and the total area of the cafeteria.

2.6 User Documentation

The application provides the users with a clear and descriptive guideline for an appealing user experience. A wide range of hover texts are available to help navigate the user through the application. The application will include an FAQ or similar page that describes the application's terms, its functionality, and any other information deemed relevant to users. In case of further queries, the user shall contact the support team.

2.7 Assumptions and Dependencies

The assumption is that the canteen is open for breakfast, lunch, and dinner every working business day in which clients are expected to be in the campus.

The application however has two dependencies:
i. The operation of the system depends on changes being made in System to accept payment requests for meals ordered with the application.

ii. *The operation of the application depends on changes being made in the Canteen Inventory System to update the availability of food items as the orders are accepted.*

iii. *The application uses Firebase database for online storage of information like orders and menu items that needs to be in working state. If the Firebase interface changes the application needs to be adjusted accordingly.*

3 EXTERNAL INTERFACE REQUIREMENTS

3.1 User Interfaces

3.1.1 Customer Interface

The customer interface will contain three screens. All three screen will have a consistent layout.

3.1.1.1 Place Order

In this screen, system shows a list of cards (UI Elements) of dishes. Each dish will have an image, its price per serving.

3.1.1.2 Timer and Edit/Cancel Order

After confirming the order, the user will be shown a timer screen. In this screen customer will be shown "Edit Order" and "Cancel Order" buttons and a timer which shows the completion time of the order. There will also be a button to request for bill.

3.1.1.3 Feedback

In feedback screen, a button for "Request Bill" will be shown. Beneath this button we will display a form which will have different multiple-choice questions and a submit feedback button.

3.1.2 Cafeteria staff Interface

In head chef interface, system will show all the current orders in detail i.e., all the dishes of a particular order. In each order, there is a button which will be used to mark that dish cooked. Moreover, when customer wants to remove a dish from his order, system will show head chef a notification to approve the removal of the dish.

3.1.3 Manager Interface

As Admin is authorized to perform CRUD operations on Staff Members, Menu Items, and Inventory Items. He'll be having three different screens for Staff Members, Menu Items, and Inventory.

3.2 Hardware Interfaces

Our system can interact with a hardware device directly. We have to connect our system to a camera that would be used for client attendance using face recognition and order collection confirmation using object detection. Clients will require a smartphone too place an order on the application. Moreover, the central screen in kitchen which will be displaying the status of order queues

3.3 Software Interfaces

- *For Database services system shall use to Firebase latest version.*
- *System will run on android version above or equal to android 11*
- *System shall use Picamera and Jetson Nano or Raspberry Pi*

3.4 Communications Interfaces

Cascade is an android application, and it will communicate with Firebase (which is a storage server provided by Google for android developers). Firebase uses HTTP protocol for communication, so our device will follow HTTP protocol when connecting to Firebase.

4 SYSTEM FEATURES

4.1 Place Order

4.1.1 Description and Priority

The system will give customers the ability to place their orders using our product. It will display a list of available and unavailable dishes in the menu where unavailable dishes will be grayed out. Customer will be able to select multiple dishes and their quantity for a particular order.
Priority: high

4.1.2 Stimulus/Response sequences

When user enters the order activity/page, initially system displays a list of available and unavailable dishes along with their prices.

i. **Stimulus:**
Customer taps on an available dish.
Response:
System shows a popup having name of the dish and price per serving. Also, it contains a text box for the customer to enter the quantity, Next button, and a Cancel button.

ii. **Stimulus:**
Customer enters the quantity and press Next button.
Response:
System closes the popup, shows quantity selected and total price of that dish.

iii. **Stimulus:**
Customer taps on confirm order button at the bottom
Response:
System closes the order screen and displays a timer along with a "Cancel Order" button and "Edit Order" button

4.1.3 Functional Requirements

i. *The system will show a list of cards (UI element) of dishes. Each card will have a picture of the dish. Below the dish it shows the price in Rupees per serving.*

ii. *The system must show all available and unavailable dishes to the Customer.*

iii. *Tap on any of the displayed dish will result in a popup for quantity and a green mark after quantity has been selected.*

iv. *The popup for quantity input will not allow the user to enter letters, negative numbers, or any invalid characters.*

v. *After completing the order, the system will display a timer "Time to complete the order" and it is the total time required to serve the dish keeping in view the previously queued orders. Moreover, it also shows a cancel order button.*

vi. *Unavailable dishes must be displayed but their operations must be disabled.*

4.2 Staff Order Queue

4.2.1 Description and Priority

Whenever a new order is placed by the Customer, the dishes in the orders are classified into categories. There is a centralized screen in the kitchen which displays queues for each chef. Each item in the queue is labeled with the name of the dish.
Priority: high

4.2.2 Stimulus/Response sequences

Stimulus:
Customer taps the "Confirm Order" button in "Place Order screen".
Response:
Displays the dishes on kitchen screen in order of queue.

4.2.3 Functional Requirements

System will classify the dishes in the order according to category and add this dish in a queue on the kitchen screen.

4.3 Edit Order

4.3.1 Description and Priority

Customer can edit the order any time before the serving. In editing mode, the customer can change the quantity of the of the food ordered, add, and remove dishes from the order.
Priority: high

4.3.2 Stimulus/Response sequences

The timer screen shows two buttons "Cancel Order" and "Edit Order" button

i. **Stimulus:**
Customer taps on "Edit Order" button.
Response:
System shows the previous menu screen where selected dishes are already marked.

ii. **Stimulus:**
Customer taps on any of the selected dish.
Response:
System opens a popup with previous quantity pre-filled. This popup will also contain a button "Remove Dish".

iii. **Stimulus:**
Customer enters new quantity and press "OK"
Response:
System shows an error "Cannot edit <Name> dish" or System closes the popup and new quantity will be displayed on that dish in the list.

iv. **Stimulus:**
Customer taps on "Remove Dish"
Response:
System responds with "Dish <Name> removed" or "Dish <Name> cannot be removed"

v. **Stimulus:**
Customer taps on any new dish which was not previously selected
Response:
Stimuli/Responses of "Place Order" feature will be followed.

4.3.3 Functional Requirements

i. *System must allow the Customer to increase, decrease or even remove the dish from the order any time before serving.*
ii. *System must remove the dish or decrease quantity of the dish with the approval of head chef.*

4.4 Cancel Order

4.4.1 Description and Priority

Our system will also provide an option to cancel the current order. When the customer taps on the "Cancel Order" button. Customer can cancel the order at any time before serving.
Priority: high

4.4.2 Stimulus/Response sequences

Stimulus:
Customer taps on the "Cancel Order" button

Response:
System responds with a popup "Order canceled successfully" or "Order cannot be cancelled"

4.4.3 Functional Requirements

i. *System must allow the customer to cancel order at any time before serving.*
ii. *In cancel order, all the dishes will be presented for approval to the staff. Only approved dishes will be dropped.*

4.5 Mark Dish as Cooked

4.5.1 Description and Priority

The staff can mark the dish of a particular order complete once it is prepared.
Priority: high

4.5.2 Stimulus/Response sequences

The system will show a list of current orders in earliest first order on the screen. Moreover, it also shows a list of dishes for each order. Alongside of each dish there is a button saying, "Marked Cooked".

Stimulus:
Head chef taps on the "Mark Cooked" button on a dish in an order.
Response:
System changes that button to a green tick.

4.5.3 Functional Requirements

i. *System must send a notification to the staff once all the dishes of a particular order has been marked "cooked".*
ii. *System must replace the timer screen with a new screen having feedback and request bill options.*

4.6 Customer Feedback

4.6.1 Description and Priority

The system will give customers the ability to give a feedback for the food or overall services. In the feedback screen there are multiple choice questions each having two options "Satisfactory" and "Unsatisfactory". At the end there is a submit button.
Priority: high

4.6.2 Stimulus/Response sequences

Stimulus:
The customer taps on request bill option
Response:
The system shows a feedback screen with multiple choice questions and a submit button.

4.6.3 Functional Requirements

i. *System must show the feedback screen to the user.*
ii. *System must display multiple choice questions for feedback.*

4.7 Add/Edit/Delete Staff Members

4.7.1 Description and Priority

The system gives ability to the admin to add, edit and delete staff members. Using this feature an admin can add chefs, waiters, managers.
Priority: high

4.7.2 Stimulus/Response sequences

Admin/Manage screen shows a grid of staff members. There is a button at the top of grid which says Add Member. In the grid after every entry there is a "Edit" and "Remove" button.

i. **Stimulus:**
Admin taps on "Add Staff" button
Response:
System opens another screen with a form

ii. **Stimulus:**
Admin fills the information and hit submit
Response:
System responds with "<Staff Member> added successfully"

iii. **Stimulus:**
Admin taps on edit button
Response:
System opens a screen with a form prefilled with the existing values.

iv. **Stimulus:**
Admin edits the information and hit submit
Response:
System responds with "<Staff Member> edited successfully"

v. **Stimulus:**
Admin taps on remove button on a particular row
Response:
Responds with a "<Staff Name> removed successfully"

4.7.3 Functional Requirements

i. Admin should be able to add all necessary information about the staff member
ii. System must give admin the ability to edit information about all staff members
iii. System must give admin the ability to remove staff members.

4.8 Add/Edit/Delete Menu Items

4.8.1 Description and Priority

The system gives ability to the admin to add, edit and delete staff members. Using this feature an admin can add chefs, waiters, managers.
Priority: high

4.8.2 Stimulus/response sequences

Admin screen shows all the previously added dishes. It also shows a "Add Dish" button along with "Edit" and "Remove" with all the available dishes

i. **Stimulus:**
Admin taps on "Add Dish" button
Response:
System opens another screen with a form

ii. **Stimulus:**
Admin fills the information and hit submit
Response:
System responds with "<Dish> added successfully"

iii. **Stimulus:**
Admin taps on edit button
Response:
System opens a screen with a form prefilled with the existing values.

iv. **Stimulus:**
Admin edits the information and hit submit
Response:
System responds with "<Dish Member> edited successfully"

v. **Stimulus:**
Admin taps on remove button on a particular row
Response:
Responds with a "<Dish> removed successfully"

4.8.3 Functional Requirements

i. *Admin should be able to add all necessary information about the staff member*
ii. *System must give admin the ability to edit information about all staff members*
iii. *System must give admin the ability to remove staff members.*

5 NON-FUNCTIONAL REQUIREMENTS

5.1 Performance Requirements

- *The system shall accommodate 400 users during the peak usage time window of 8:00am to 10:00am local time, with an estimated average session duration of 8 minutes.*
- *All the pages generated by the system shall be fully downloadable in no more than 10 seconds over a 40KBps modem connection.*
- *Responses to queries shall take no longer than 7 seconds to load onto the screen after the user submits the query.*
- *The system shall display confirmation messages to users within 4 seconds after the user submits information to the system.*

5.2 Safety Requirements

No safety requirements have been identified.

5.3 Security Requirements

There is a need for a proper and encrypted login authentication for head chef and admin as employee sensitive information as well as inventory should be protected from hacking. Information transmission should be securely transmitted to Firebase without any changes in information to avoid disturbances in orders and billing

5.4 Software Quality Attributes

5.4.1 Adaptability:

There can be a change in the menu and information stored in the database about employees and inventory.

5.4.2 Availability:

The system is up and running for most of the time and server is not down for more than a few minutes to avoid inconvenience of the customers.

5.4.3 Correctness:

The bill generated by the application must be accurate and the orders placed should exactly be the same which the user has selected.

5.4.4 Flexibility:

If need arises in the future, software can be modified to change the requirements.

5.4.5 Interoperability:

The data is transferred from the customer's end to the kitchen and then staff is assigned orders. This way data is transferred from one part of the system to another.

5.4.6 Maintainability:

Software can be easily repaired if a fault occurs.

5.4.7 Portability:

Software can be easily installed on devices and would run smoothly according to the requirement.

5.4.8 Reliability:

No matter how many orders are placed, system must give the correct results.

5.4.9 Reusability:

Current version can be used in the future versions with more functionality added.

5.4.10 Robustness:

Software must have checks to ensure that the items that are not available in the menu cannot be selected and the emails, phone numbers added are all valid.

5.4.11 Testability:

All the requirements are fulfilled, response time is low, and all functions are working perfectly.

5.4.12 Usability:

Interface of the software must be easy to use. It would not be complex since managers, chefs have a view, so interface should be simple.

5.5 Business Rules

- *Manager's interface contains the view of tables that are free, and manager can just view and doesn't provide any input to the system.*
- *Once the bill is paid, manager can mark the order as paid.*
- *Admin has access to perform add, delete, update operations on the database for menu, inventory, employees, and no other person can modify the data in the database.*
- *Customers can place order from the list of available items and can update order and pay bill.*
- *Staffs are assigned orders and can update the queues and has an additional functionality of load balance.*

6 OTHER REQUIREMENTS

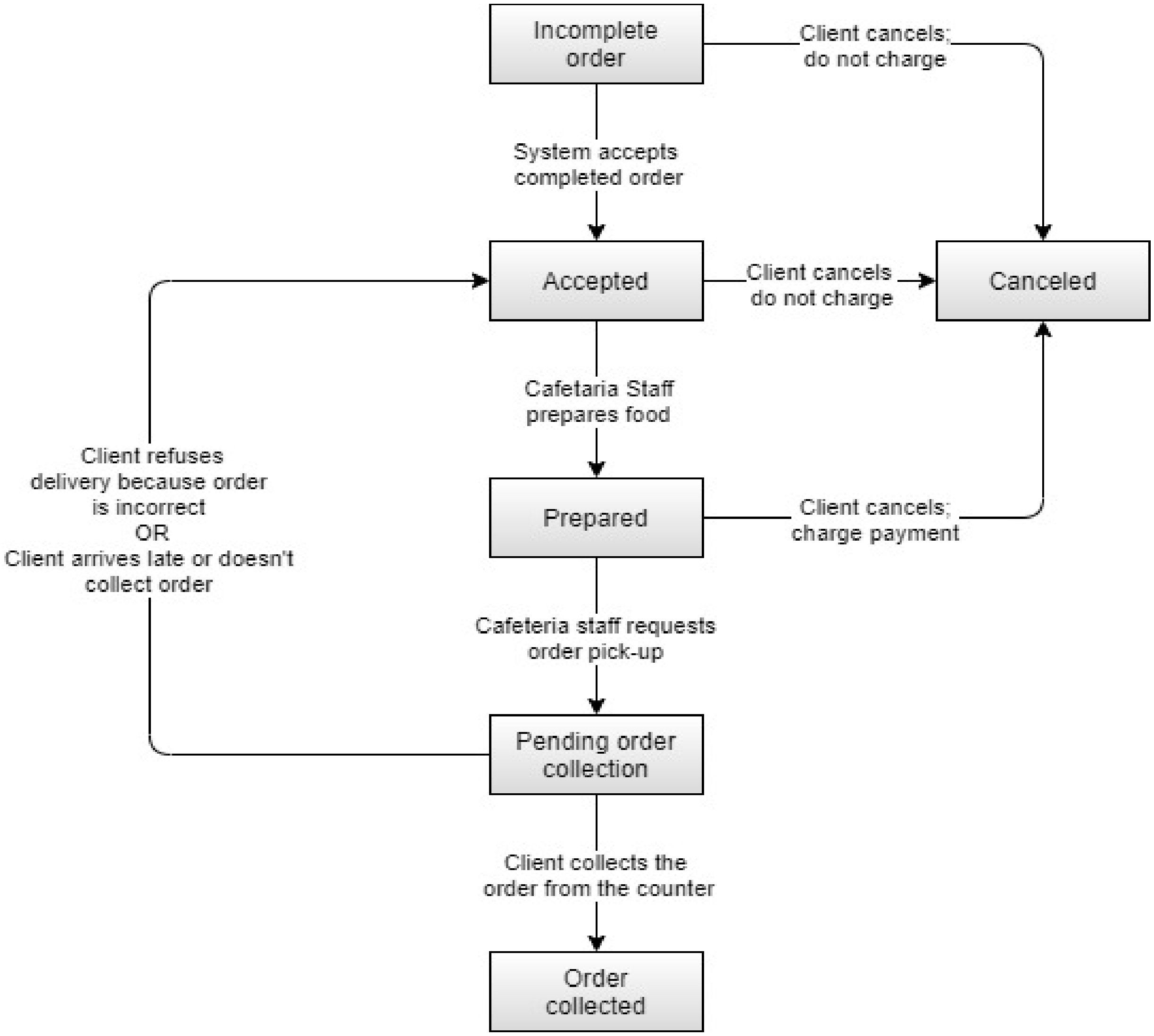
Appendix A: Glossary

This part involves acronyms and abbreviations used in the SRS.

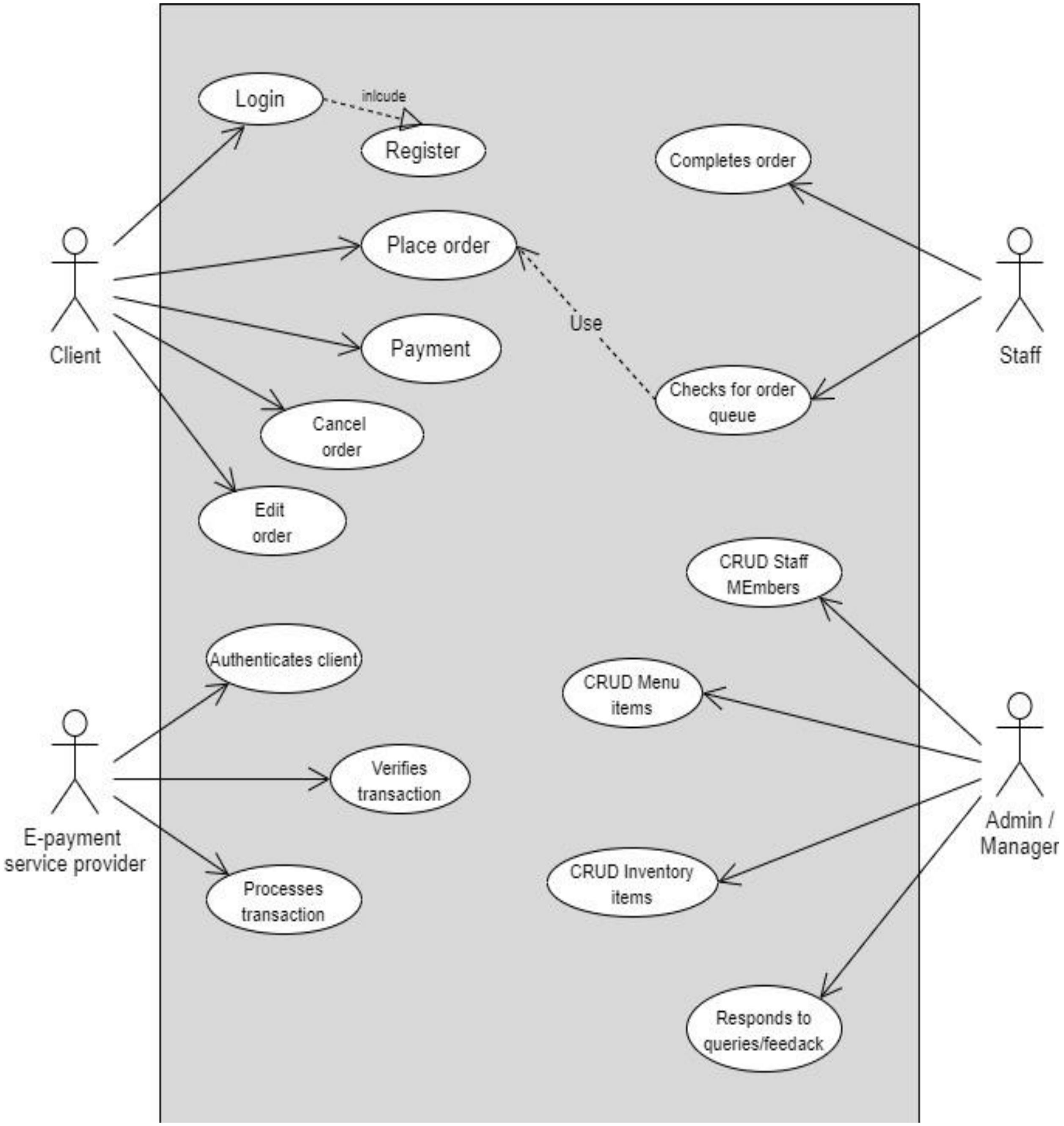
- *SRS: Software Requirements Specification*
- *AI: Artificial Intelligence*
- *DESC: Description*
- *ERD: Entity-relationship diagrams*
- *Graphic Depicting - small icons that correspond to the weather.*
- *CRUD - In computer programming, create, read, update, and delete (CRUD) are the four basic functions of persistent storage. Alternate words are sometimes used when defining the four basic functions of CRUD, such as retrieve instead of reading, modify instead of update, or destroy instead of deleting.*

Appendix B: Analysis Models

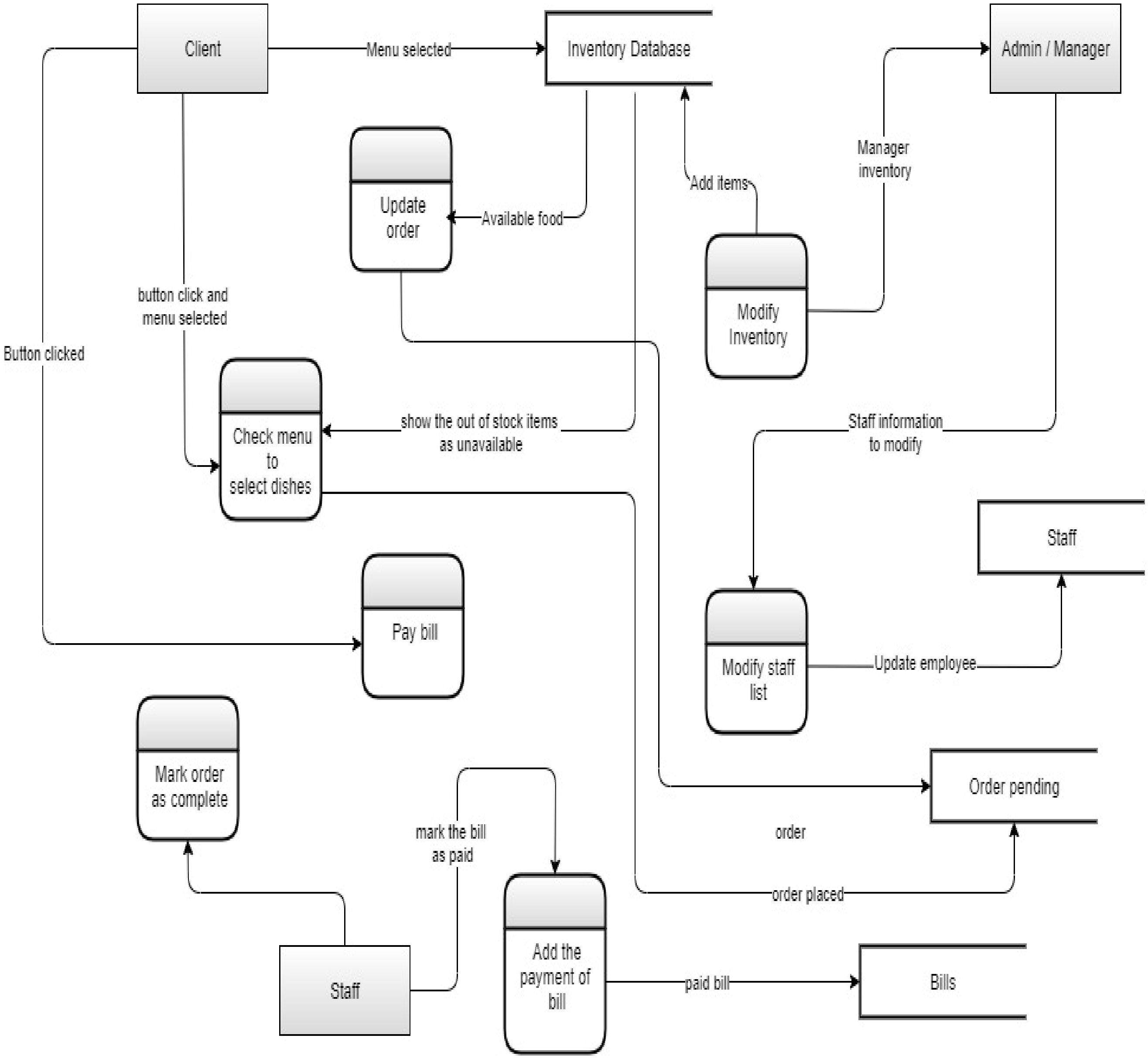
- *State Transition Diagram*


Incomplete order
Client cancels; do not charge
System accepts completed order
Accepted
Client cancels do not charge
Canceled
Cafetaria Staff prepares food
Client refuses delivery because order is incorrect
OR
Client arrives late or doesn't collect order
Prepared
Client cancels; charge payment
Cafeteria staff requests order pick-up
Pending order collection
Client collects the order from the counter
Order collected

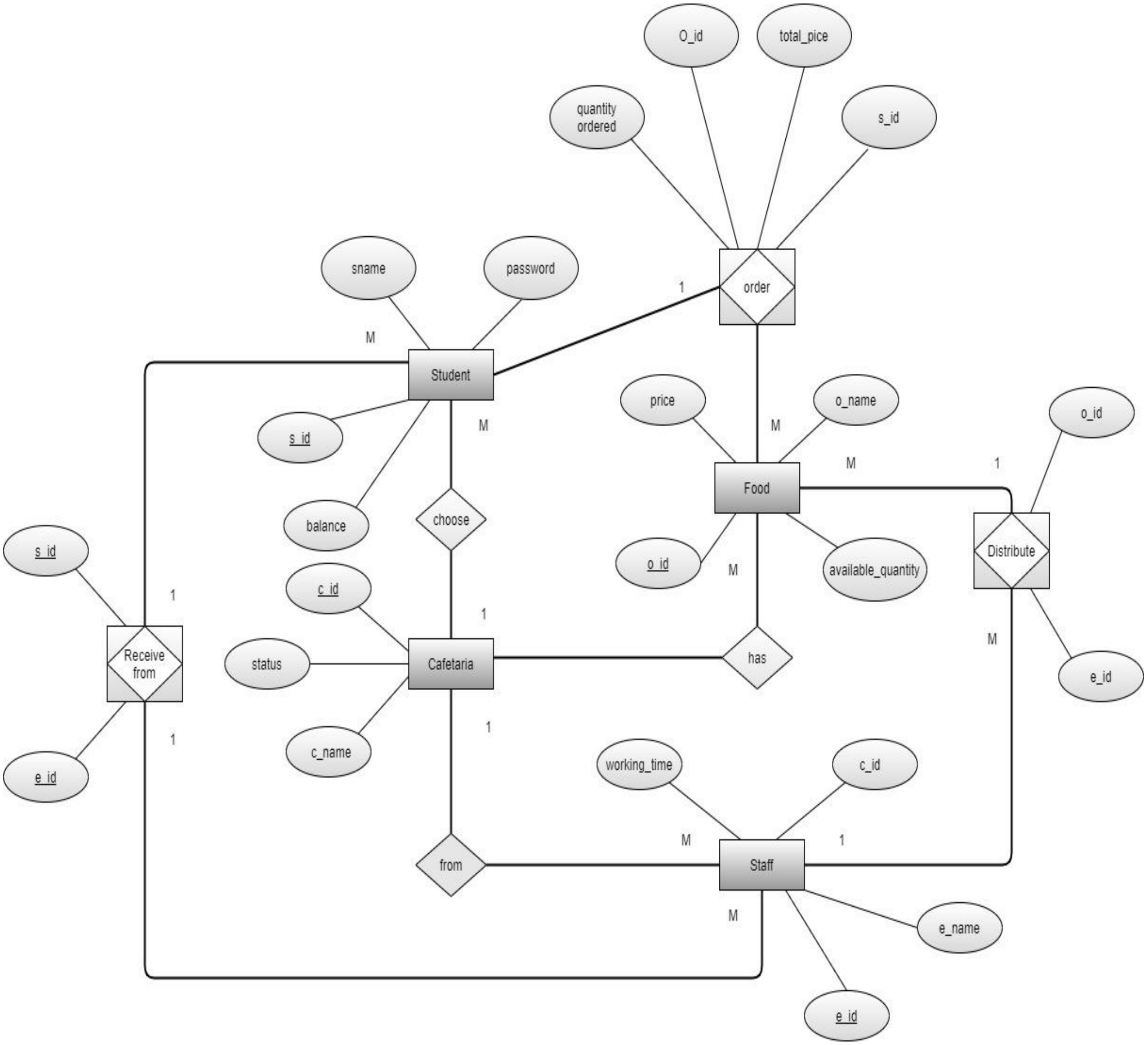
- *Use case Diagram*



- *Data Flow Diagram-Level 2*



- *E-R diagram*


O_id
total_pice
quantity ordered
s_id
sname
password
1
order
M
Student
M
M
s_id
price
o_name
o_id
M
1
balance
choose
Food
Distribute
s_id
c_id
o_id
M
available_quantity
1
1
Receive from
status
Cafetaria
has
M
e_id
1
1
c_name
working_time
c_id
e_id
M
1
from
Staff
M
e_name
e_id

Appendix C: To Be Determined List

Weekly sales report and tracking most ordered dish and prioritizing its inventory stocking feature (restock the items that are most ordered often) is yet to be determined by the client and may need further meetings for elaboration.

Adding POS (point of sale) features to the application is yet to be determined as well.