Institute of Information Technology,
University of Dhaka

IIT
University of Dhaka

CARS Cafeteria
Management System

Software Requirement Specification & Analysis [SE-406]

Submitted By
GROUP-1
Dip Saha (BSSE 1001)
MD Akram Khan (BSSE 1007)
MD Sabbir Hossain (BSSE1014)
Ashraful Gafur (BSSE 1021)
Junaid Mansur Ifti (BSSE 1027)
Mehedi Hasan Arif (BSSE1036)

Submitted To
Dr. Kazi Muheymin-Us-Sakib
Professor,
Institute of Information Technology,
University of Dhaka .

Submission Date: 11th November, 2019

Contents

INTRODUCTION

This chapter is a part of our software requirement specification for the

project "Cars Cafeteria Management System" . In this chapter we will focus on the

intended audience for this project.

PURPOSE

This document briefly describes the Software Requirement Analysis of

Cars Cafeteria Management System. It contains the functional, non-functional and the supporting requirements and establishes a requirement's baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered and organized by topics. The SRS serves as an official means of communicating user requirements to the developer and provides a common reference point for both the developer team and the stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

INTENDED AUDIENCE

- This SRS report is intended for several audiences including the users(Teachers & Officers) , admin(Cafeteria's manager), project managers, developers and testers.
- The users and admin will use this SRS to verify that the developer team has created a product that is acceptable to the customer.
- The project managers of the developer team will use this SRS to plan milestones and a delivery date, and ensure that the developing team is on track during development of the system.

- The designers will use this SRS as a basis for creating the system's design.The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs.
- The developers will use this SRS as a basis for developing the system's functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created a software that will fulfill all of the customer's documented requirements.
- The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfills the requirements documented in this SRS. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

CONCLUSION

This analysis of the audience helped us to focus on the users who will be using our analysis. This overall document will help each and every person related to this project to have a better idea about the project.

INCEPTION OF CCMS

In this chapter, the Inception part of the SRS will be discussed briefly.

INTRODUCTION

CARS Cafeteria is a Cafeteria for teacher and Officers located at Mokarram Bhaban ,University of Dhaka.Teaches and Officers from different departments have their lunch in this Cafeteria and different types of events or parties are also held here by booking the whole cafeteria in the evening hours.Several employees work under one manager in this Cafeteria.

INCEPTION OF CARS Cafeteria Management System

At the beginning of our project, we entered the inception stage. This stage

includes, how the project will be started and their scope and limitations. The main goal of this phase is to identify the requirements, demand and

establish some sort of mutual understanding between the software team and the stakeholders of the Cars Cafeteria. In order to make this phase effective we took the following steps:

- Identifying the client of our project
- Icebreaking
- Identifying the stakeholders of the CARS Cafeteria
- Identifying the multiple viewpoints of stakeholders

IDENTIFY THE CLIENT OF OUR PROJECT

Admin of the CARS Cafeteria can be identified as a client of our project.Teachers and Officers can be viewed as stakeholders.

ICEBREAKING

Icebreaking refers to the fact that to diminish the communication barrier

between two persons. It is a crucial part since it denotes the

acceptation of our proposal. We started this phase by talking with the stakeholders with context free languages. Their behavior, respond to our question impacted the whole system.

IDENTIFYING THE STAKEHOLDERS OF THE CCMS

Stakeholder refers to any person or group who will be affected directly or

indirectly by the system. Stakeholders include end-users who interact with

the system and everyone else in an organization who may be affected by its installation. The CARS Cafeteria have limited number of stakeholders.They are:

- Teachers
- Officers
- Manager of CARS Cafeteria

IDENTIFYING THE MULTIPLE VIEWPOINTS OF THE STAKEHOLDER

Different stakeholders expect different benefits from the system as every

person has his own point of view. So, we have to recognize the requirements

from multiple viewpoints. Different viewpoints of the stakeholders about the

expected software are given below:

CARS Cafeteria's Viewpoint:

- First and foremost, a really friendly user interface
- Desktop and Mobile,both platform based software if affordable
- Provide signal when any grocery item is out of stock
- Store information about the employees and the teachers/officers
- Calculate total buy and sell amount of a month and show the whole cost and benefits periodically
- Easy Reservation Management
- Secured and Automated Transaction System directly connected to the Cafeteria's account
- Easy Transaction History Storing
- Automated Memo Giving System

Users' Viewpoint:

- Easy and Fast Interface
- Mobile Platform Based Software
- Availability of Food Delivery directly to office rooms
- Secured Online payment
- Cash On Delivery System
- Easy Reservation System to book for any official ceremony

CONCLUSION

The primary goal of this project is to model and design a software for the teachers and officers of the University(Specially science faculties) to ease lunch order system and For Managers to easily manage multiple orders and reservations. For these reasons,The software will be as

simple as a teacher can easily be able to use this and the managers can maintain it without any annoyance. The software will be designed in such a way as it takes very little time to manage. To make this software project successful, collaboration with stakeholders was a main priority that what they want, how the software will work, how it can be more convenient, how it will save time and energy, etc.

ELICITATION OF CCMS

We have seen Question and Answer (Q& A) approach in the previous chapter, where the inception phase of requirement engineering has been described. The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. The collaborative working approach of the stakeholders is required to elicit the requirements. We have finished the following tasks for eliciting requirements-

- Collaborative Requirements Gathering
- Quality Function Deployment

- Usage Scenarios(Story)

COLLABORATIVE REQUIREMENTS GATHERING

We have met with many stakeholders in the Inception phase such as the manager,teachers and officers. These meetings created an indecisive state

for us to elicit the requirements. To solve this problem, we have met with

the stakeholders (who are acting a vital role in the whole process) few times to elicit the requirements.

Quality Function Deployment

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software. Ultimately the goal of QFD is to translate subjective quality criteria into objective ones that can be quantified and measured and which can then be used to design and manufacture the product. It is a methodology that concentrates on

maximizing customer satisfaction from the software engineering process. So, we have followed this methodology to identify the requirements for the

project. The requirements, which are given below, are identified successfully by the QFD.

Normal requirements:

Normal requirements are generally the objectives and goals that are stated for a product or system during meetings with the customer. The presence of these requirements fulfills customers' satisfaction. These are the normal requirements for our project.

- Users will create an account by providing their credentials.
- A predefined account will be given to system admin.
- System will verify the user's credentials from database.
- Users can update his/her profile.
- Users can recover their password if forgotten.
- Users must be logged in before doing any operation.
- Users can order food online.
- A memo will be provided after order confirmation.
- Order details will be added into virtual cart.
- Users can reserve table(s) for lunch through online.
- Users can book whole cafeteria after lunch through online.
- If any user reserves cafe which is already booked, he will be assigned into queue. First queued user will be notified after every booking cancellation.
- Users can pay bills through "SSL Commerz" and cash on delivery.
- If any users cancel order or reservation, he will be refunded.
- Estimated time of food delivery will be prompted in the user's display.
- Staff info, user info, every transaction details will be stored into admin database.
- A replica of user info, order memo and transaction history will be stored in user database.
- If a user updates his/her info, it will be updated in user database first and then the same update will be replaced into Admin Database automatically.
- Admin can manage everyday menu.
- Inventory management.
- If a food is out of stock it will be shown as stocked out.

Expected requirements

These requirements are intrinsic to the product or system and may be so elementary that the customer does not explicitly state them. Their absence

will be a cause for significant dissatisfaction. Below the expected

requirements for our project are briefly described-

- The system will be secured.
- Delivery time for every delivery will be estimated by GPS
- Several users can request for tables and cafeteria at the same time.
- Responsiveness of the System will be expeditious.
- Transaction history of Non Registered users will also be recorded .
- Interactive and attractive graphical user interface.

Exciting requirements

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present. Following are some exciting requirements of our project:-

- If a user order certain items frequently, it will be prompted in the user's homepage.
- All relevant food items according to user's taste will be suggested.
- After analyzing the orders of previous one month from the users, the system will show the mostly ordered food items in the admin's homepage.

Usage Scenario:

CARS Cafeteria Management System

CCMS (Cars Cafeteria Management System) is an automated system where CARS cafeteria will be managed. This is intended to ease both the Management (administrator) and Teachers/Officers (Users) to interact, make orders, reservations etc with each other more conveniently.

1) Account Management:

1.1) Create Account:

User Perspective: A user must create an account to enter into the system. He/she needs to provide the following information to create account-

- Full name
- Mobile number
- Email address
- Teacher/Officer id
- Password
- Department name
- Room number & Location

After providing the information, admin will verify and send a confirmation code to the provided mobile number. By inputting this code, account will be created.
Admin Perspective: An account for the administrator will be given to the cafeteria management with a predefined username and password.

1.2) Verification:

System will verify the user's credentials from teachers' database of Dhaka University.

1.3) Update Account:

User can update his/her profile. He/she can change his/her following information-

- Email
- Mobile number
- Password
- Room number

1.4) Password Recovery:

A user can recover his/her password if forgotten, by using his/her email or mobile phone number. User can click on "Forget Password" Button and choose from two options-"1.Recover Through Email, 2.Recover Through Mobile Number"
Through Email: A recovery link will be sent to user's email, if user clicks on "Recover Through Email" button. User will then input a new password and his/her password will be updated in Database.
Through Mobile: An OTP will also be sent to the user's mobile number , if a user clicks on "Recover Through Mobile" button. User will have to input the OTP within 1 minute and then he/she will have to input a new password. His/her password will be updated in Database.

1.5) Log in:

A user can log into the system by using his/her registered email-id or phone number and password.
An administrator can login with the predefined username and password given by the system authorities.

2) Ordering:

To order, user must be logged in to the system. There will be daily menu provided by the cafeteria management which will include pricing and stock alert(if a food is out of stock it will be shown as stocked out).User can choose to order food from the menu according to his/her taste. He/she can also select quantity. After choosing, the food will be added to virtual cart and the amount of bill to be paid will be shown. He/she can pay the bill using any authorized payment system (Described in the following section "Payment") or he/she can also avail Cash-On Delivery (COD) service. If the user takes "Cash-on Delivery" service he/she must confirm that before payment and will have to pay some extra charge for delivery.
If user chooses to pay using online payment system, his/her order will be confirmed after payment. Else if user chooses COD service, his/her order will be confirmed instantly.
After order confirmation, A memo will be given to the user.
User can only give orders for food before 12.30 p.m. No order will be taken after this certain time.
User can also cancel order after payment, but it must have to be before 12.30 p.m. He/she will be refunded if cancelled before time.

3) Reservation:

For reservation, user must log in. Then he/she can choose one or multiple table numbers to book. After booking tables, he/she must provide the customized menu and quantity of people. User can also book the whole Cafeteria after lunch hour for any big occasions or ceremonies.

User must book before 48 hours for an event. Payment will be calculated based on the venue rate and customized food menu rate. Advance payment of 30% must be done within 2 hours of reservation confirmation. He/she will be notified through email and sms about confirmation of reservation.
To cancel the reservation, he/she must notify it before 36 hours to get refunded.

If any user wants to reserve tables which are already booked, the user will be assigned into the booking queue. If the user who reserves in the first place cancels the reservation in time, the queued user next to him/her will be notified and the table(s) will be reopened for booking.

Each reservation has a reservation id.

4) Payment:

Online Payment: External Sub-System "SSL Commerz" will be integrated for online payment method. Cafeteria's Transaction accounts will be added to SSL Commerz and users will just have to pay by logging into their account using this system.
Notification email and sms will be sent to user after every transaction by SSL Commerz.It will also be automatically added to admin database by SSL Commerz system.
On-Site Payment: User can also pay on spot if he/she goes to the cafeteria by himself/herself. After payment he can choose any table (not reserved). A memo will be provided including order number.
Cash-On Delivery: User can pay cash to the delivery man if he chooses Cash On Delivery.
For cancellation before a certain time, they will be refunded within 3-4 working hours.If a user fails to pay within the specific time, his/her reservation will be cancelled automatically.

5) Memo :

After confirmation of each order, a memo will be prompted to the user including the following information-

- Order id
- Food item with price
- Quantity
- Total price

In case of office delivery, a printed copy of memo will be provided to the user through the delivery man.
1-SSLCOMMERZ uses industry standard Secure Sockets Layer (SSL) technology which is used worldwide for securing data encryption. It is also PCI DSS v3.2 compliant which is the highest grade of recognition of Data Security compliance in the Payment Card Industry
In case of on spot payment, a printed copy of memo will be given to the user hand to hand after payment.

6) Delivery :

User perspective: Users can request for room delivery inside the university campus. For this service, he/she has to pay a delivery charge (25/-). Estimated time of food delivery will be prompted in the user's display.

Admin perspective: Admin can deliver the order in two ways-

- By staff
- By "Pathao Food Service"

System will track the location from where the user has requested, through GPS. If requested location is within 200m radius of CARS Cafeteria, then Staff will be sent to deliver foods.
Otherwise, foods will be sent by "Pathao Food Service."After confirming the delivery, the staff or "Pathao" riders will communicate with the user and deliver the order.

7) Database:

Admin Database: Staff info and User info will be stored in admin database. Every reservation details will also be stored in admin database.
Delivery staff numbers will be given input everyday and will be stored in database. . If a staff goes for room delivery, the availability count for the staff will be automatically deducted by admin input. After each office delivery, when staff will come back to the cafeteria, admin will update the database again.
Every order memo will be stored distinctly in the database. Admin will be able to see the sum of transactions happened in a day.After every order paid,either through delivery-man or pathao or on the spot ,the transaction detail will be added into Admin Database.

User Database: Every order memo and transaction history will be stored in User Database. A replica of user info from admin database will also be stored in User Database for security consolidation. If a user updates his/her info, it will be updated in user database first and then the same update will be replaced into Admin Database automatically.

8) Additional Features:

- If a user order certain items frequently, it will be prompted in the user's homepage.

- All relevant food items according to user's taste will be suggested.

- After analyzing the orders of previous months from the users, the system will show the mostly ordered food items in the admin's homepage.

9) Administrative Management:

9.1) Menu:

The menu will be fixed for a day. If any change is required, then admin will update the menu. The availability of food items will be displayed when the user wants to order.

After every order is confirmed (Online/Offline), menu will be updated and the quantity of items confirmed will be deducted from the availability count. If any order is cancelled, the availability of the food items will also be updated.

9.2) Inventory Management:

Administrator will input count of his/her stored grocery items. He/she will update counts everyday. By this, he/she will be able to track which item is going to be stocked out or which item is abundantly stocked.

Use Case Diagram

DEFINITION OF USE CASE

A Use Case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders. In essence, a Use Case tells a stylized story about how an end user interacts with the system under a specific set of circumstances. A Use Case diagram simply describes a story using corresponding actors

who perform important roles in the story and makes the story understandable for the users. The first step in writing a Use Case is to define that set of "actors" that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using system.

Primary Actor

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

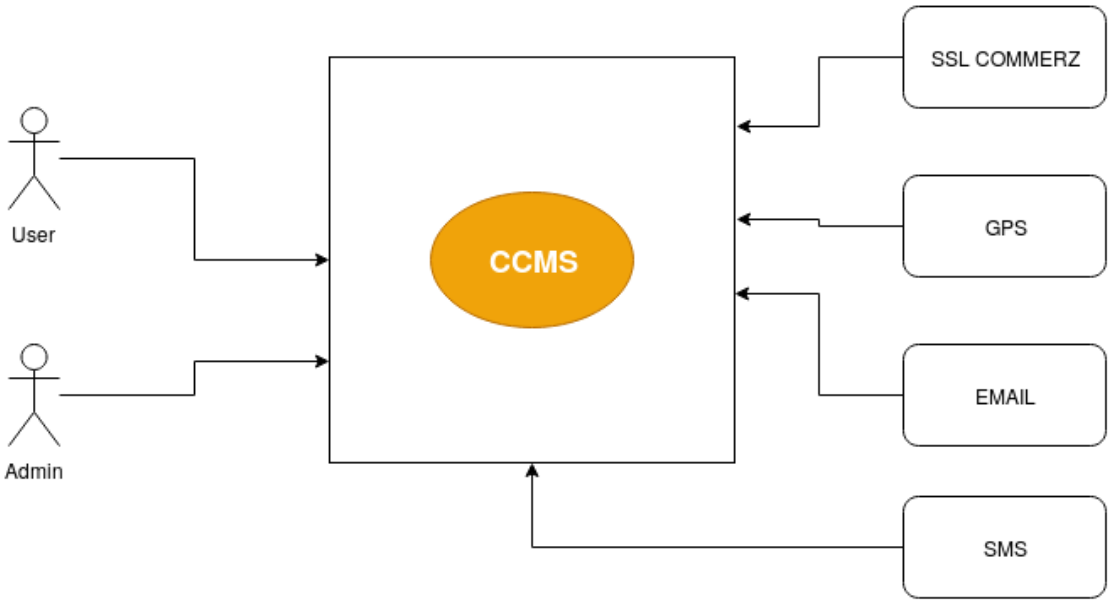
Secondary Actor

Secondary actors support the system so that primary actors can do their work.
They either produce or consume information.

Use Case diagrams give the non-technical view of overall system.

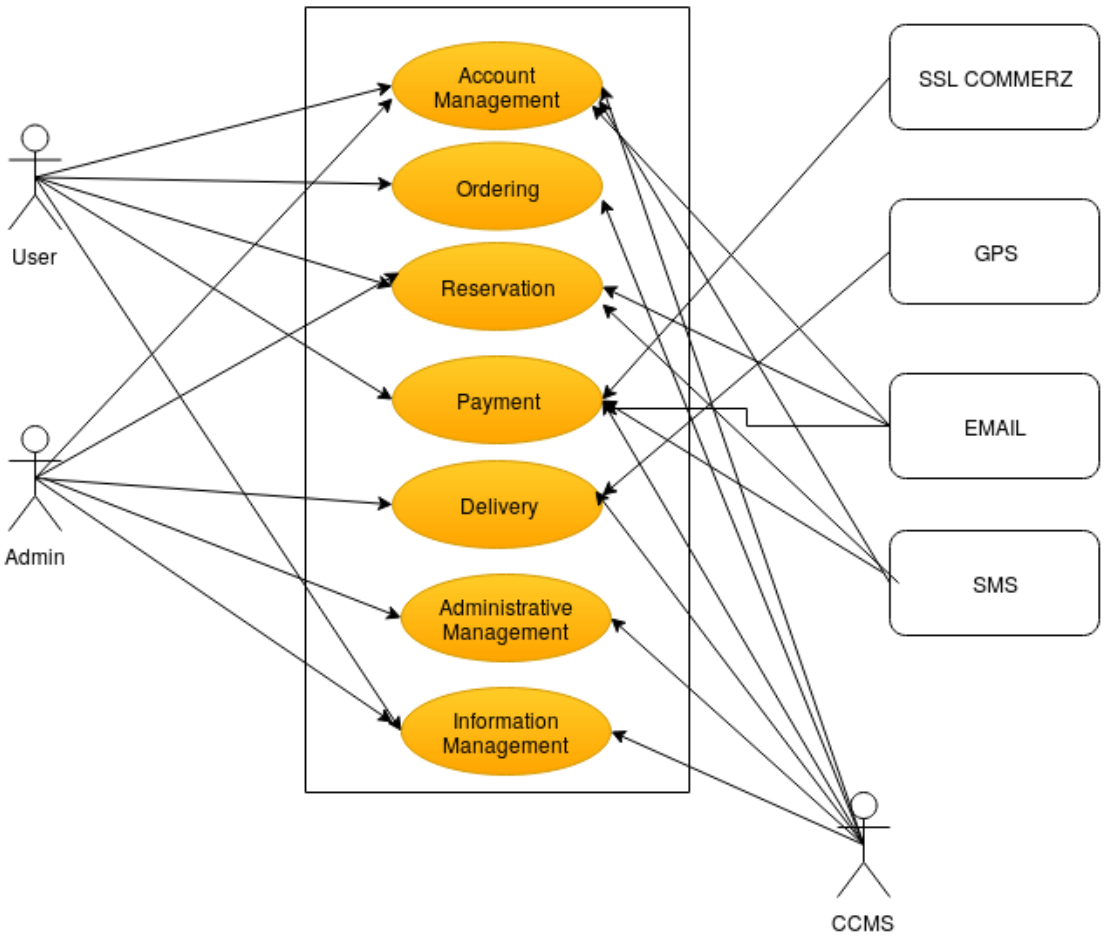
Level : 0

Name: CCMS
Primary Actor: user, admin
Secondary Actor: SSL COMMERZ,EMAIL,SMS,GPS



Level : 1

Name: CCMS
Primary Actor: user, admin,CCMS
Secondary Actor: SSL COMMERZ,EMAIL,SMS,GPS



Description of use case diagram level-1:

1. Account management : Users must create an account and then log into the system. He/she can update his/her profile, can recover password if forgotten. Admin will verify his/her provided credentials.

2. Ordering : Users will log into the system and can order food for lunch before 12:30 p.m.. He can pay through SSL Commerz or cash-on delivery or on-site payment. He/she

can cancel the order.

3. Reservation : After logging into the system, users can book one or more tables for lunch. Users can also book whole cafeteria after lunch hour. He must book before 48 hours before an event. Advance payment of 30% must be done within 2 hours of reservation confirmation.

4. Delivery : If users request for delivery service, admin will deliver through "Pathao" or staff.

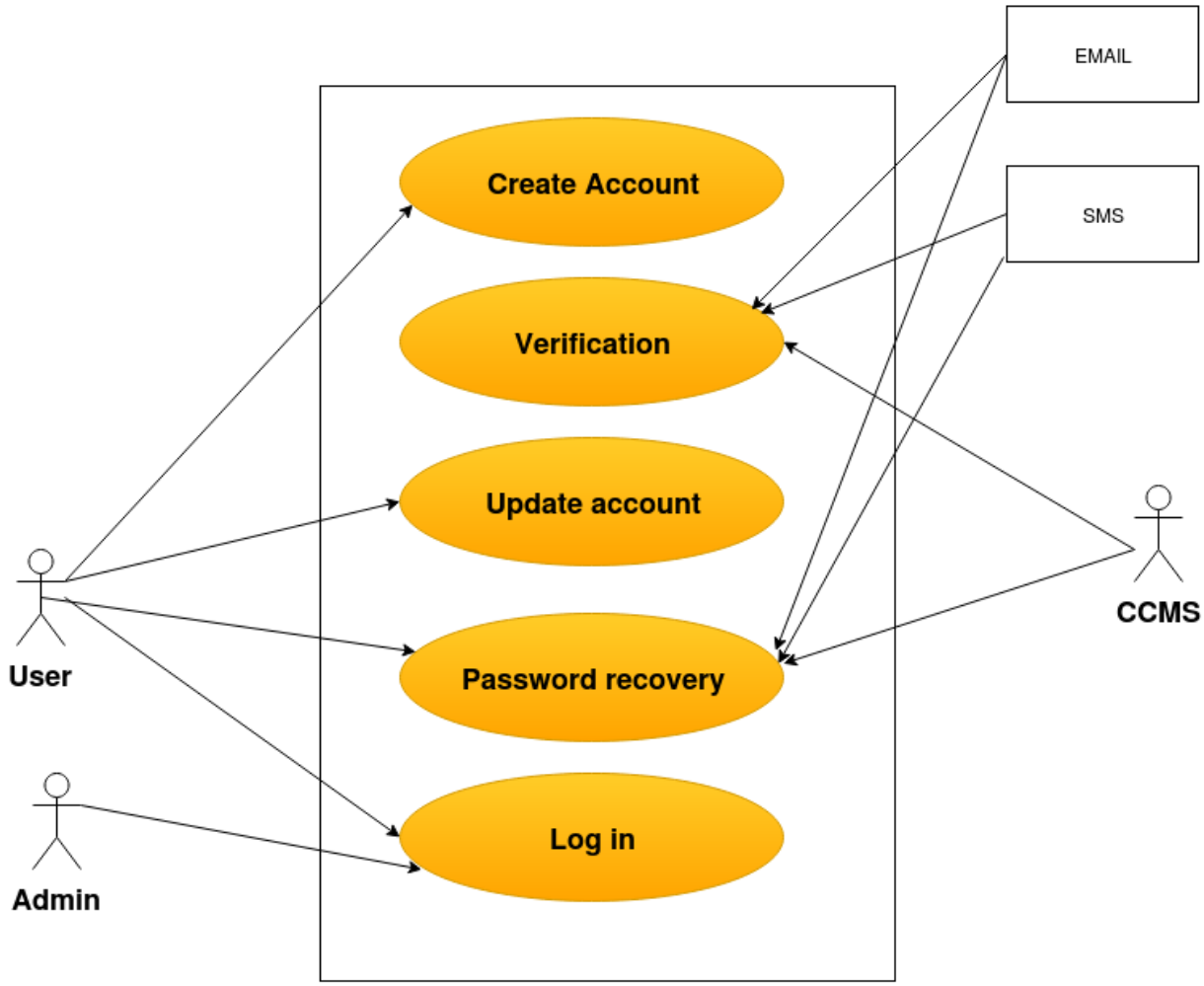
5. Administrative management : Admin will update menu and perform inventory management.

6. Information management : Staff information , user information, transaction information, reservation details, available stuff count, order memo will be stored in the database.

7.Payment: User can pay online using any approved and trusted account through "SSL Commerz"

Level 1.1

Name: Account management
primary actor: User,Admin,CCMS
Secondary actor: Email,Sms



Description of use case diagram level-1.1:

Create account : To create an account, users must provide following credentials: full name, mobile number, email address, teacher/officer id, password, department name, room number and location. Admin will verify and send a confirmation code to the provided mobile number. By inputting this code, account will be created. An account for the admin will be given with a predefined username and password.

Verification : System will verify the user's credentials from teacher's and officer's database

of Dhaka University.

Update account : Users can update his/her email, mobile number, password and room number.

Password recovery : A user can recover his/her password if forgotten, by using his/her email or phone number.

Log in : Users will log into the system by using his/her registered email-id/ phone number and password.

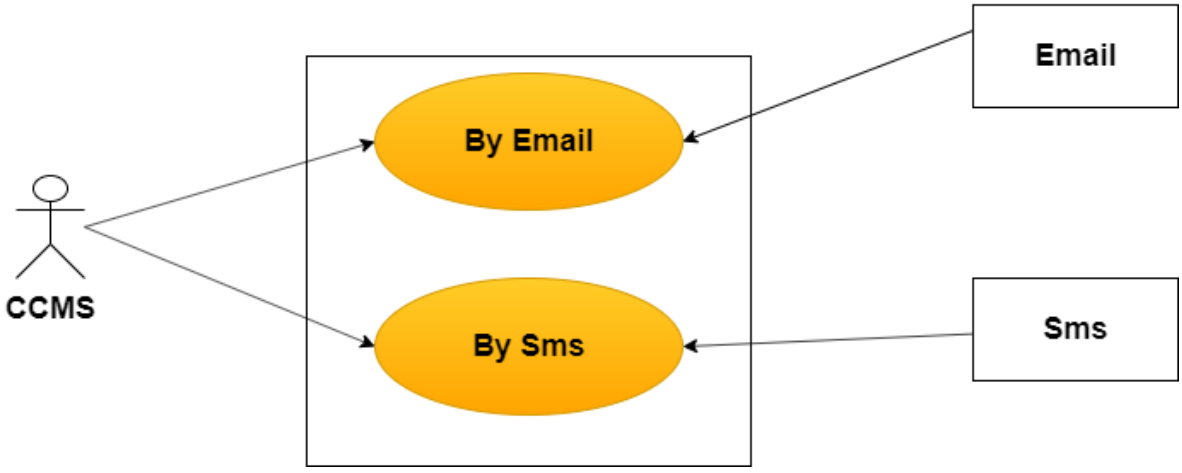
Action Reply :
Action: User provides credentials.
Reply: System will check the validity of the given credentials. For valid information system will allow user(Student or Teacher) to create an account and log into the account.

Action: User provides invalid credentials.
Reply: System will show error message and allows to try again.

Action: User provides credentials for update.
Reply: System will check the validity of the given credentials and after validation updates the given info.

Level 1.1.4

Name: Password Recoery
primaryactor: CCMS
Secondary actor: Email,Sms



Description of use case diagram level-1.1.4:

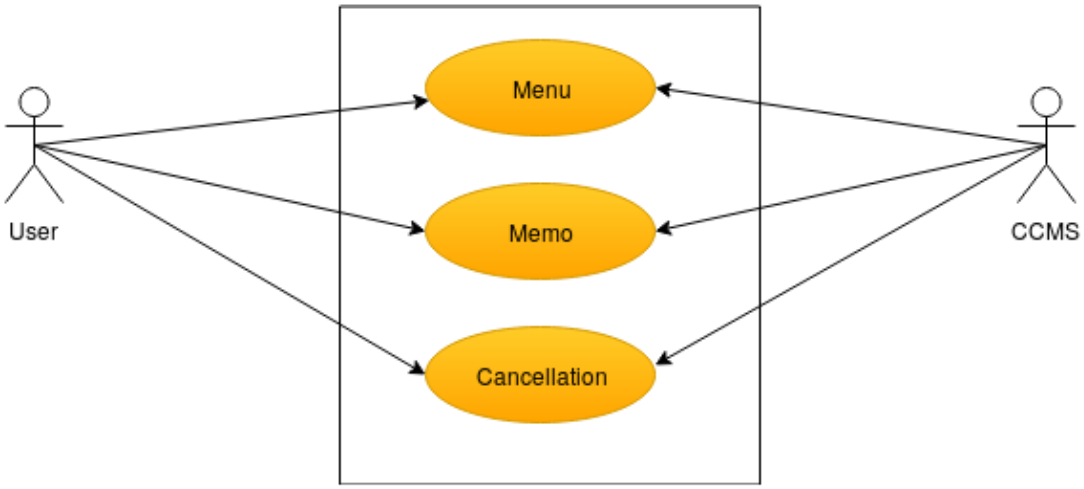
By email : System will send a recovery link to the user's email if he/she clicks on "Forget Password" button. User will then input a new password and his/her password will be updated in database.

By phone : An OTP will be sent to the user's mobile number concurrently, if a user clicks on "Forgot Password" button. User will have to input the OTP within 1 minute and then he/she will have to input a new password. His/her password will be updated in Database.

Action Reply :
Action: User requests for password recovery.
Reply: System will send an OTP or recovery link.

Level : 1.2

Name: Ordering
Primary Actor: user, admin
Secondary Actor: SSL COMMERZ,EMAIL,SMS,GPS



Description of use case diagram level-1.2:

Menu : User can choose to order food from the menu according to his/her taste. He/she can also select quantity.

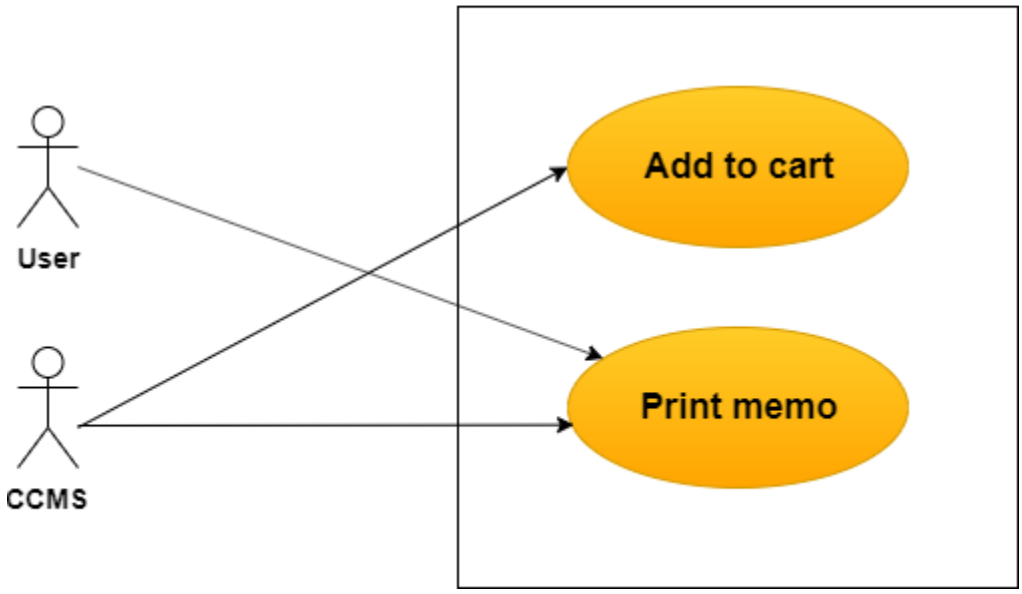
Memo : After confirmation of each order, a memo will be prompted to the user including the following information : Order id, food item with price, quantity, total price.

Cancellation : Users can cancel order after payment, but it must have to be before 12:30 p.m. .He/she will be refunded if cancelled before time.

Action Reply :
Action: User chooses food and confirms the order.
Reply: System prompts memo.
Action: User cancels order in time.
Reply : System refunds money.
Action: User cancels order after a fixed time.
Reply : System doesn't refund money.

Level 1.2.2

Name: Memo
primary actor: User, CCMS



Description of use case diagram level-1.2.2:

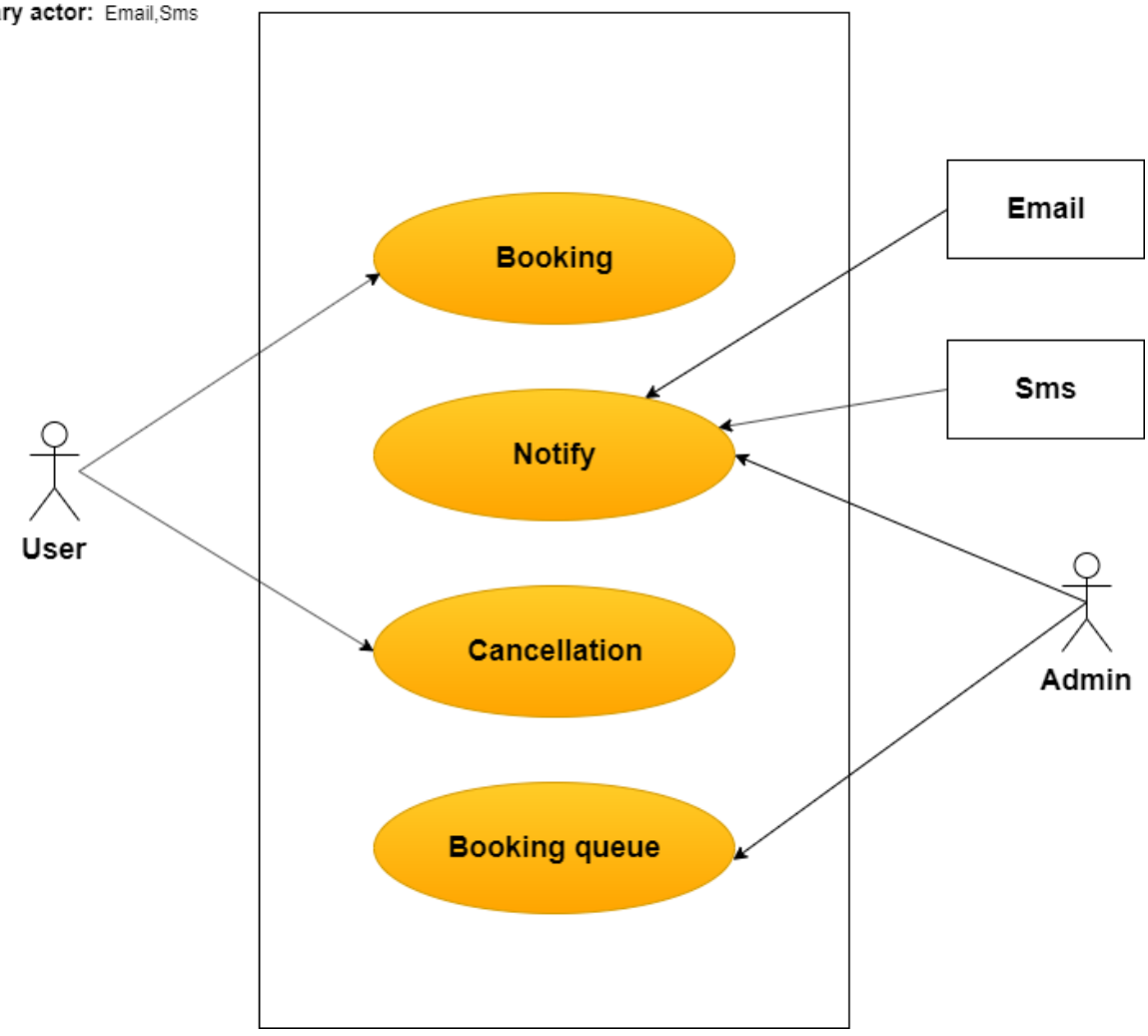
Add to cart : After choosing, the food will be added to virtual cart and the amount of bill to be paid will be shown.

Print memo : In case of office delivery, a printed copy of memo will be provided to the user through the delivery man. In case of on spot payment, a printed copy of memo will be given to the user hand to hand after payment.

Action Reply:
Action: User requests for office delivery.
Reply : System prints a memo.

Level 1.3

Name: Reservation
primaryactor: User,Admin
Secondary actor: Email,Sms



Description of use case diagram level-1.3:

Booking : User can book one or more tables for lunch or can book whole cafeteria after lunch hour.

Notify : Admin will notify user after booking confirmation, cancellation through email/sms.

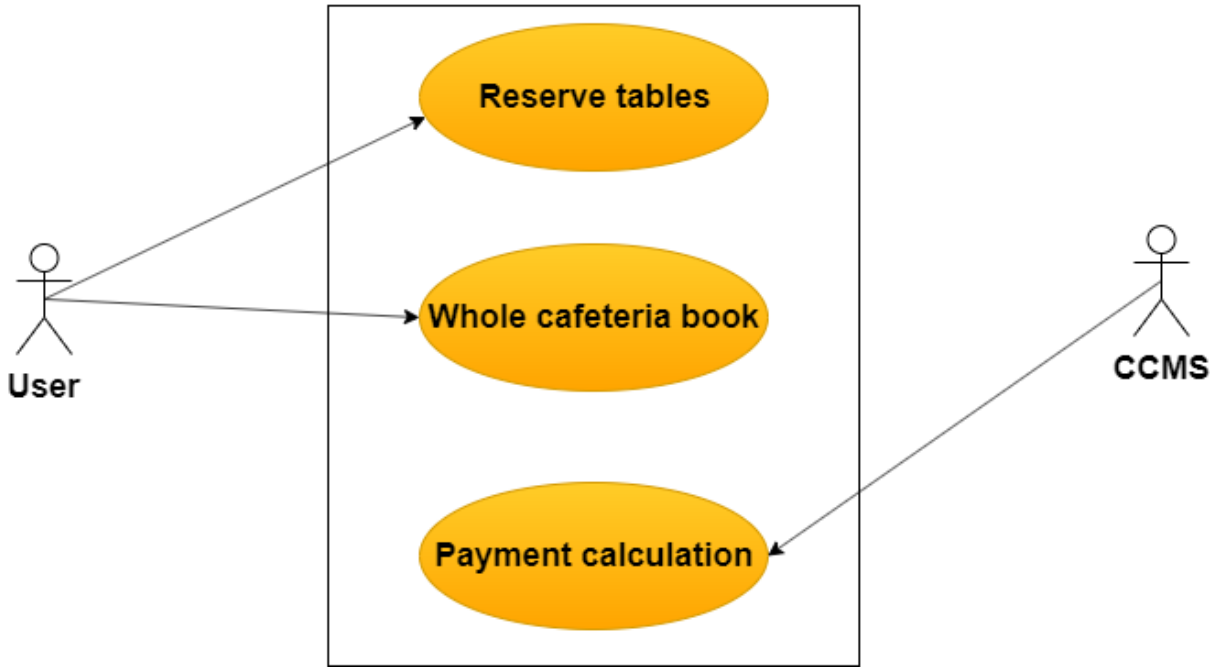
Cancellation : Users can cancel the reservation 36 hours before the event to get refunded.

Booking queue : If any user wants to reserve tables which are already booked, then the user will be assigned into the booking queue. If the user who ordered in the first place cancels the order in time, the queued user next to him/her will be notified and the table(s) will be reopened for booking.

Action Reply:
Action: Users book tables.
Reply: Admin will notify users about confirmation.
Action: Users cancel reservation in time.
Reply: Admin notifies about cancellation , refunds money and manages queue.
Action: Users cancel reservation after a fixed time.
Reply: Admin doesn't refund money .

Level 1.3.1

Name: Booking
primaryactor: User,CCMS



Description of use case diagram level-1.3.1:

Reserve tables : Users can book one or more tables. Then he will select food menu and quantity of people.

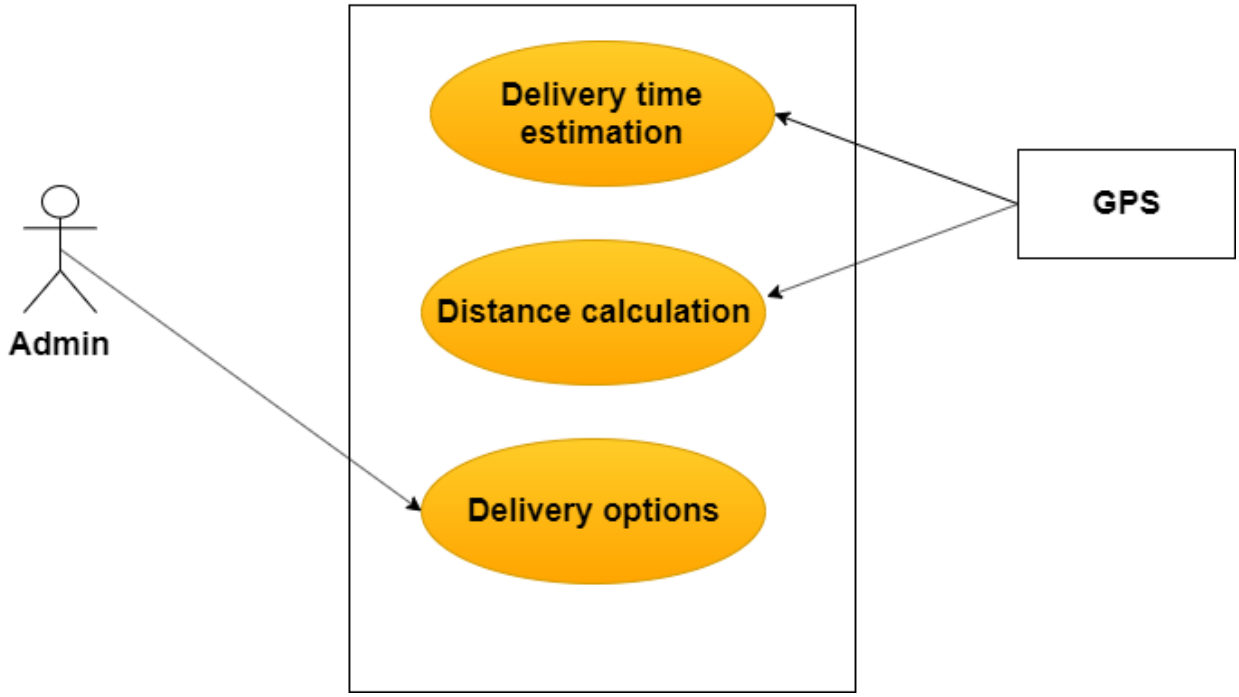
Whole cafeteria book : User can also book the whole Cafeteria after lunch hour for any big occasions or ceremonies. User must book before 48 hours for an event.

Payment calculation : Payment will be calculated according to hall rent, food item and quantity of people.

Action Reply:
Action: Users book tables , select food menu and quantity of people.
Reply: System confirms the booking and calculates the payment.
Action: Users book whole cafeteria, select food menu and quantity of people.
Reply: System confirms the booking and calculates the payment.

Level 1.4

Name: Delivery
primary actor: Admin
Secondary actor : GPS



Description of use case diagram level-1.4:

Delivery time estimation : A delivery time will be estimated by GPS and prompted in user's display.

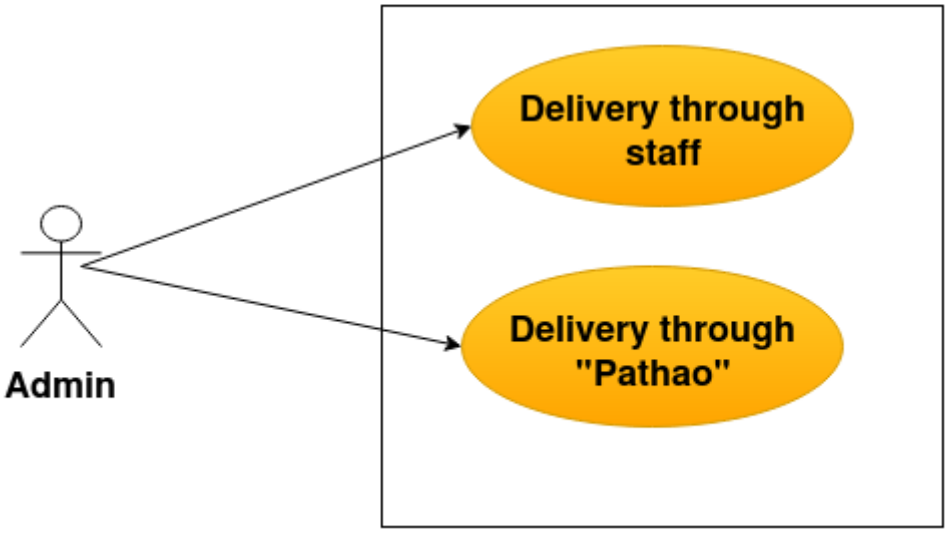
Distance calculation : Distance from CARS will also be calculated through GPS.

Delivery options : Admin will choose one delivery option according to distance.

Action Reply:
Action: Admin chooses a delivery option.
Reply: Delivery will be done according to admin's choice.

Level 1.4.3

Name: Delivery options
primary actor: Admin



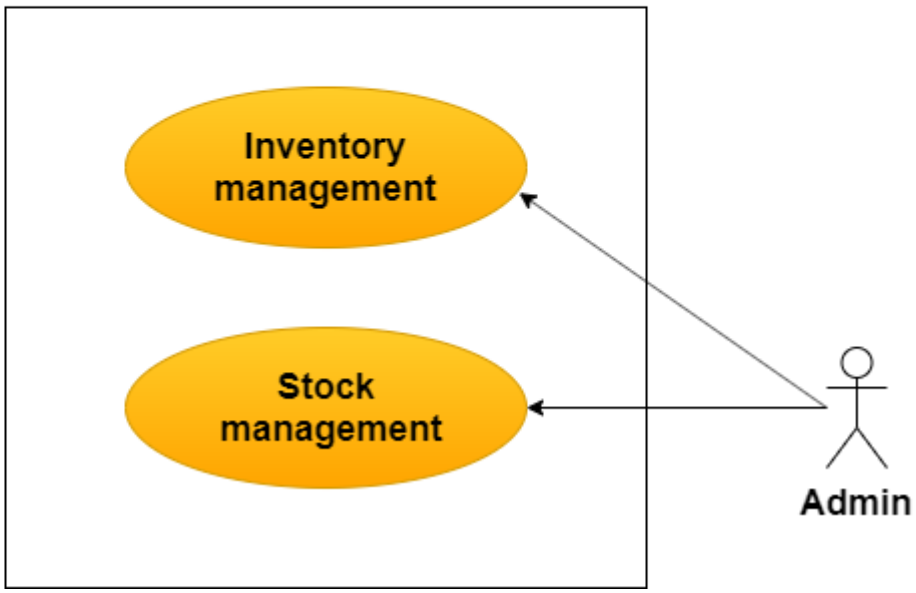
Description of use case diagram level-1.4.3:

Through staff : If the distance is less than 200 meters, admin delivers the food through staff.

Through "Pathao" : If the distance is greater than 200 meters, admin delivers the food through Pathao.

Level 1.5

Name: Administrative management
primary actor: Admin



Description of use case diagram level-1.5:

Inventory management : Administrator will input count of his/her stored grocery items. He/she will update counts everyday. By this, he/she will be able to track which item is going to be stocked out or which item is abundantly stocked.

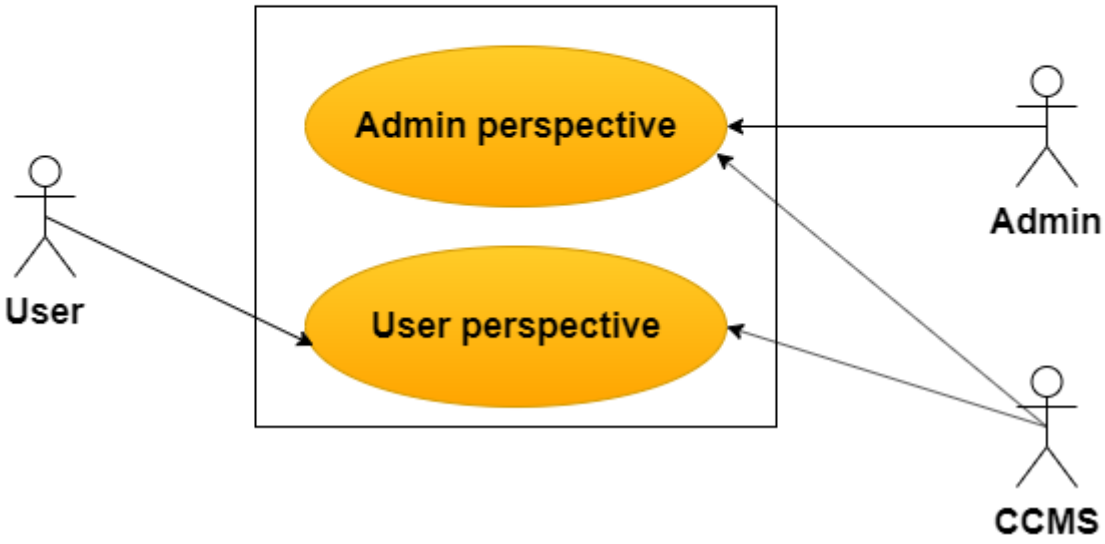
Stock management : The menu will be fixed for a day. If any change is required, then admin will update the menu. The availability of food items will be displayed when the user wants to order.

After every order is confirmed (Online/Offline), menu will be updated and the quantity of items confirmed will be deducted from the availability count. If any order is cancelled, the availability of the food items will also be updated.

Action Reply:
Action: Admin updates the menu.
Reply: Users will see the updated menu .
Action: Admin inputs the count of stored grocery items.
Reply: Admin database will be updated.

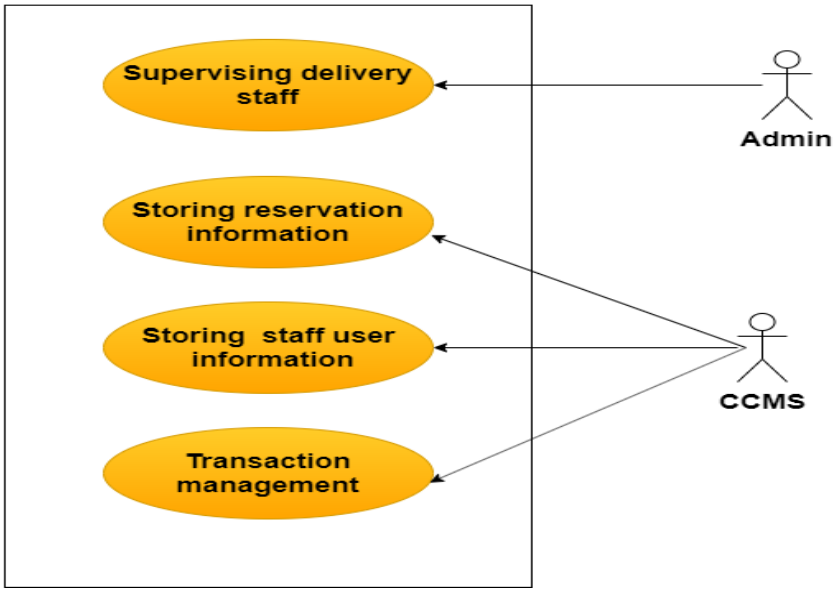
Level 1.6

Name: Information Management
primary actor: User, Admin,CCMS



Level 1.6.1

Name: Admin perspective
primary actor: CCMS, Admin



Description of use case diagram level-1.6.1:

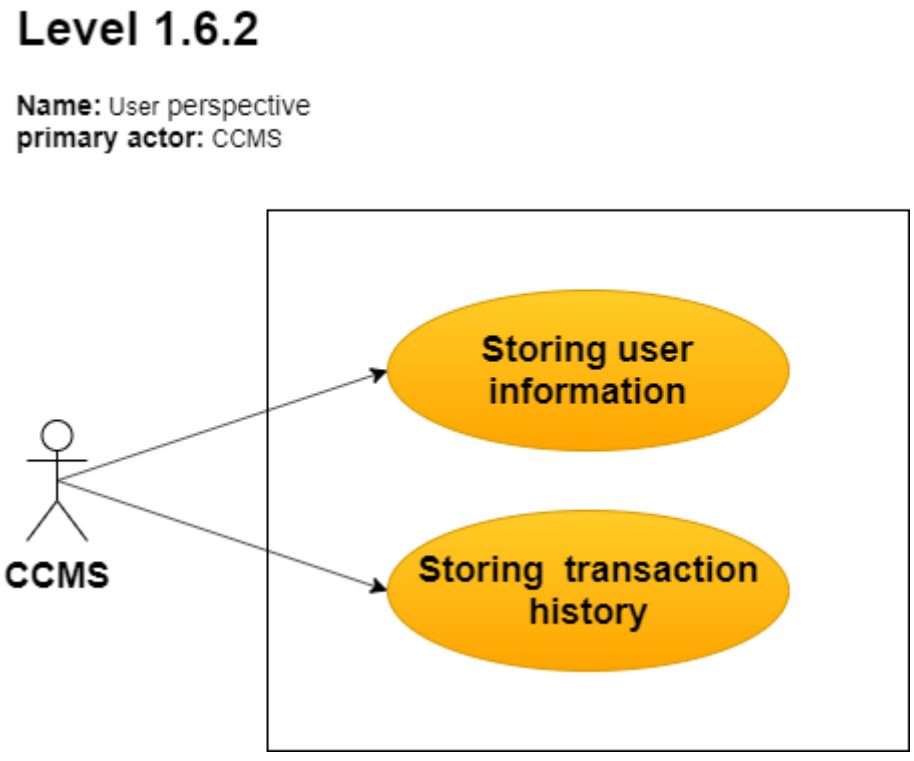
Storing staff and user's information : Staff information and user information will be stored in admin database.

Storing reservation information : Every reservation details will also be stored in admin database.

Supervising delivery staff : Delivery staff numbers will be given input everyday and will be stored in database. If a staff goes for room delivery, the availability count of the staff will be automatically deducted by admin input. After each office delivery, when staff will come back to the cafeteria, admin will update the database again.

Transaction Management:

Action Reply:
Action: Admin inputs delivery staff numbers every day.
Reply: Admin database will be updated.
Action: Admin updates delivery staff numbers.
Reply: Admin database will be updated.



Description of use case diagram level-1.6.2:

Storing user information : A replica of user info from admin database will also be stored in User Database for security consolidation. If a user updates his/her info, it will be updated in user database first and then the same update will be replaced into Admin Database automatically.

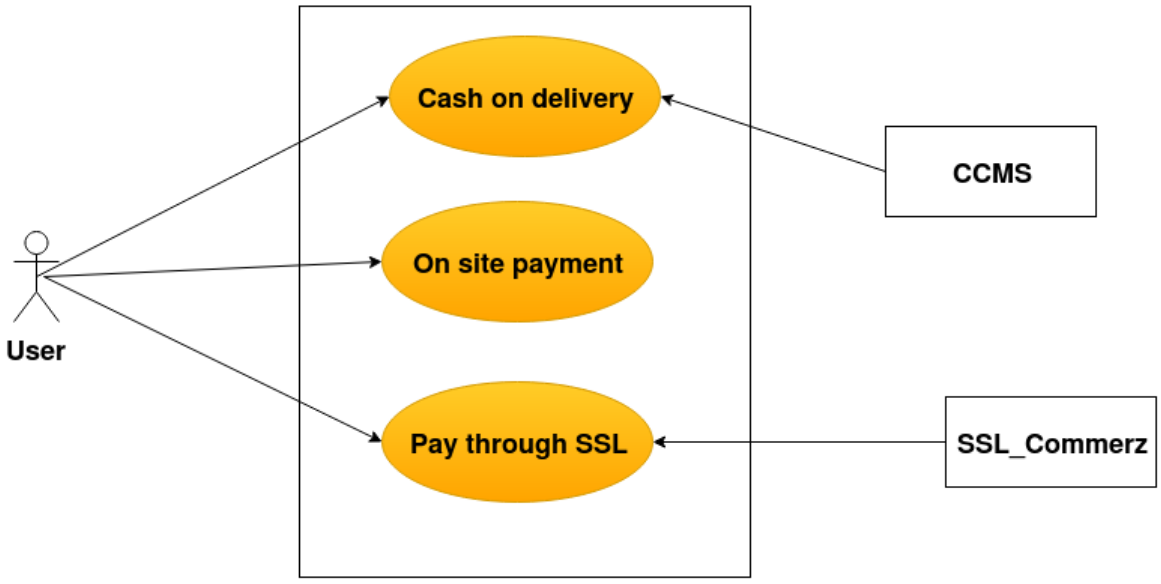
Storing transaction history : Every order memo and transaction history will be stored in User Database.

Action Reply:

Action: User updates his/her info
Reply: User database will be updated as well as admin database.
Action: Order memo and transaction history is created.
Reply: User database will be updated.

Level 1.7

Name: Payment
primary actor: User,CCMS
Secondary actor: SSL_Commerz



Payment : Orders will be confirmed after payment. Users can pay through "SSL Commerz"

or cash-on delivery or on-site payment. A notification will be sent to user's email and phone after order confirmation .

Description of use case diagram level-1.7:

Cash-on delivery : User can pay cash to the delivery man if he chooses Cash On Delivery. If he chooses cash-on delivery service, he has to pay extra delivery charge.

On-site payment : User can also pay on spot if he/she goes to the cafeteria by himself/herself. After payment he can choose any table (not reserved). A memo will be provided including order number.

Pay through SSL : External Sub-System "SSL Commerz" will be integrated for payment method. Cafeteria's Transaction accounts will be added to "SSL Commerz" and users will just have to pay by logging into their account using this system. Notification email and sms will be sent to user after every transaction by "SSL Commerz" .It will also be automatically added to admin database by "SSL Commerz" system

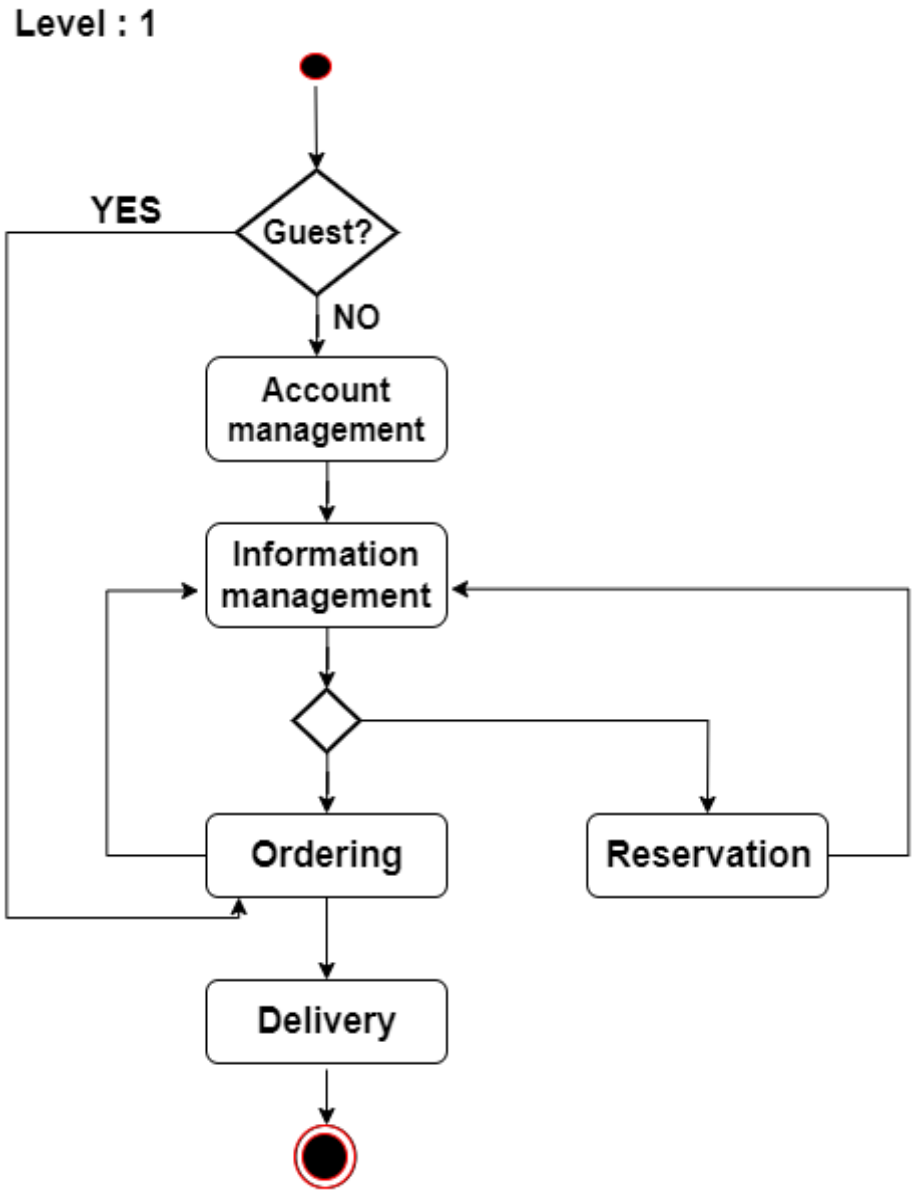
Action Reply:
Action: Users pay through "SSL Commerz" .
Reply: Notification email and sms will be sent.
Action: Users pay through "SSL Commerz" .
Reply: Admin database will be updated automatically.

Activity Diagram

CARS Cafeteria Management System (CCMS)

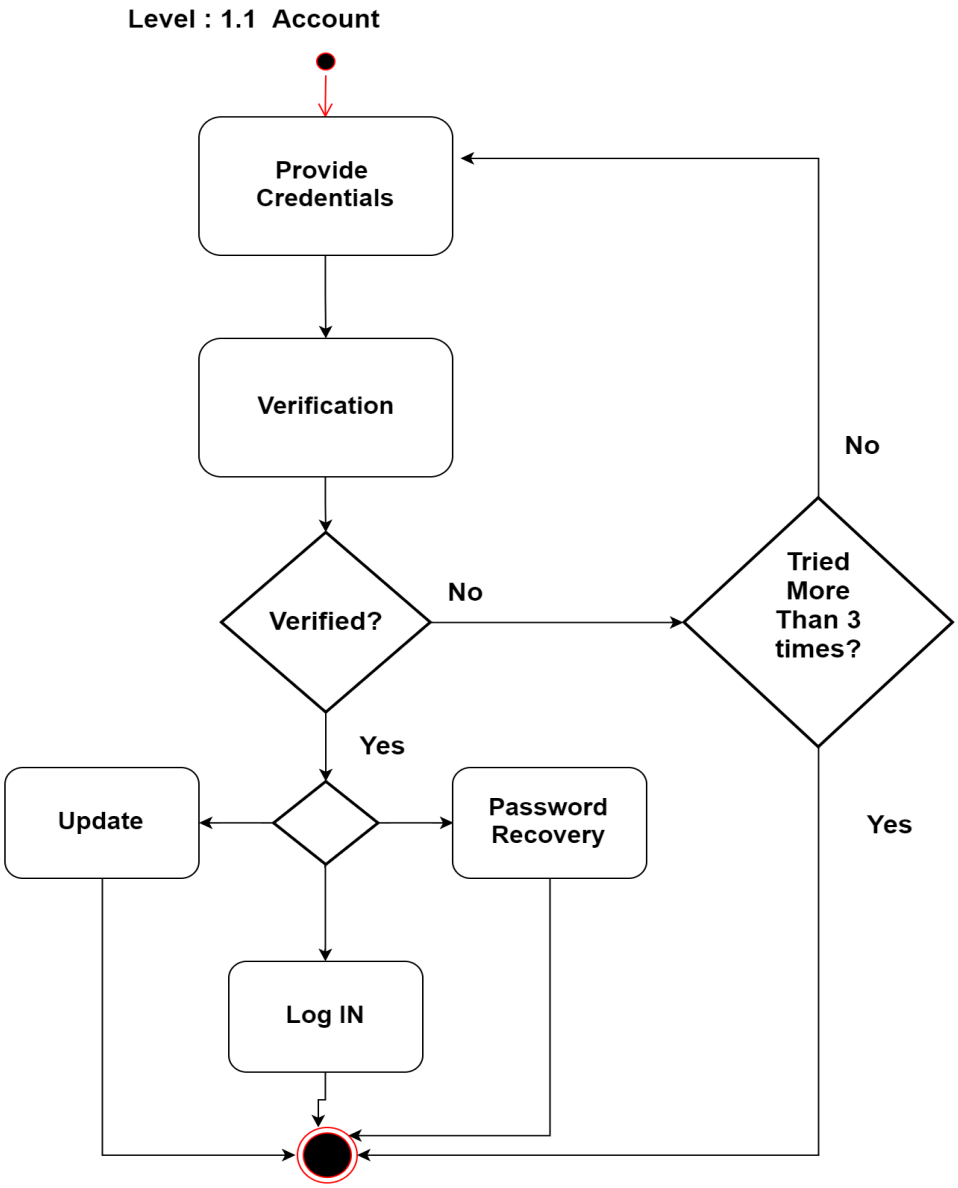
Definition of Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.



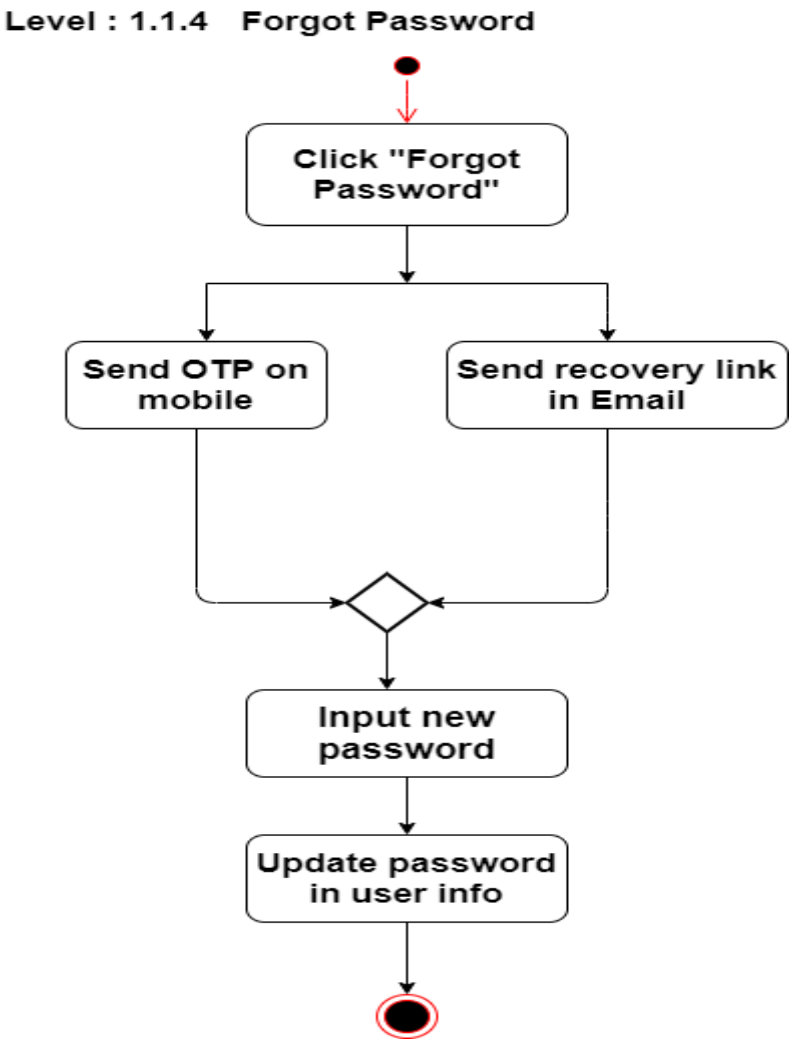
Level : 1
Name: CCMS
Reference: Use Case level 1

Account Management



Level : 1.1
Name: Account Management
Reference: Use Case level 1.1

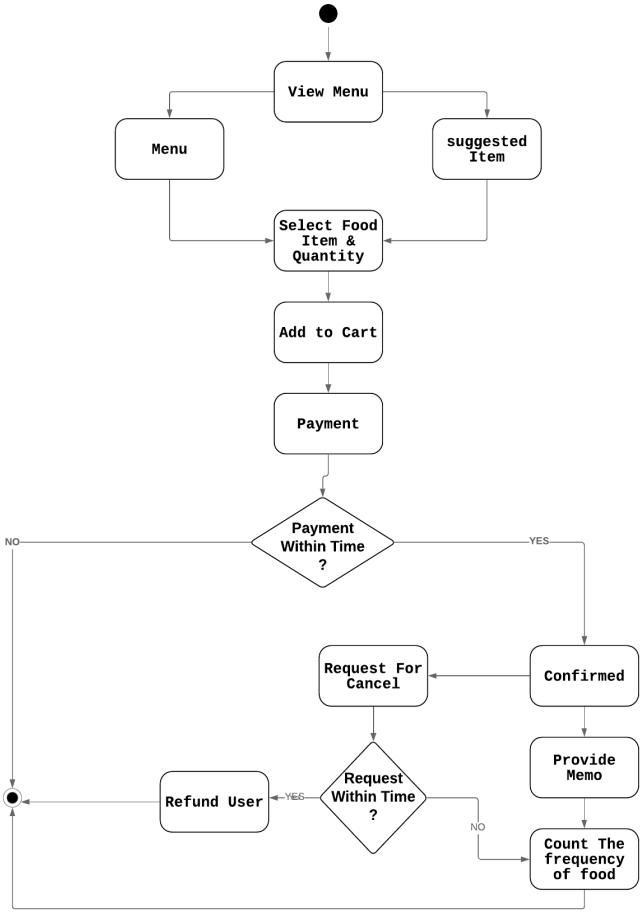
Password recovery



Level: 1.1.4
Name : Password recovery
Reference: Use Case level 1.1.4

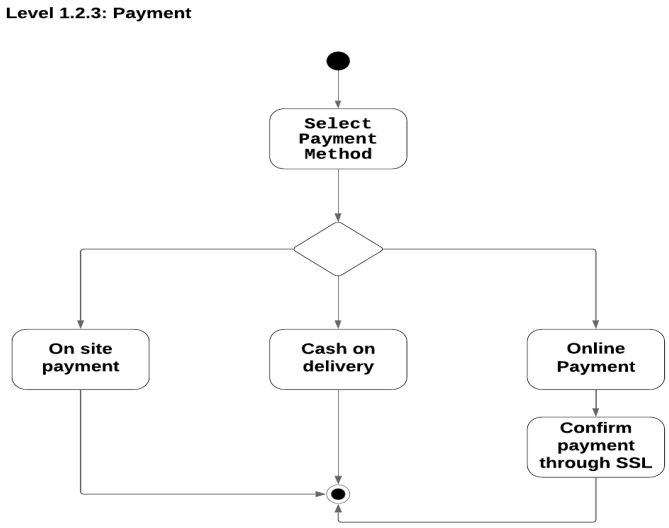
Ordering

level 1.2: Ordering



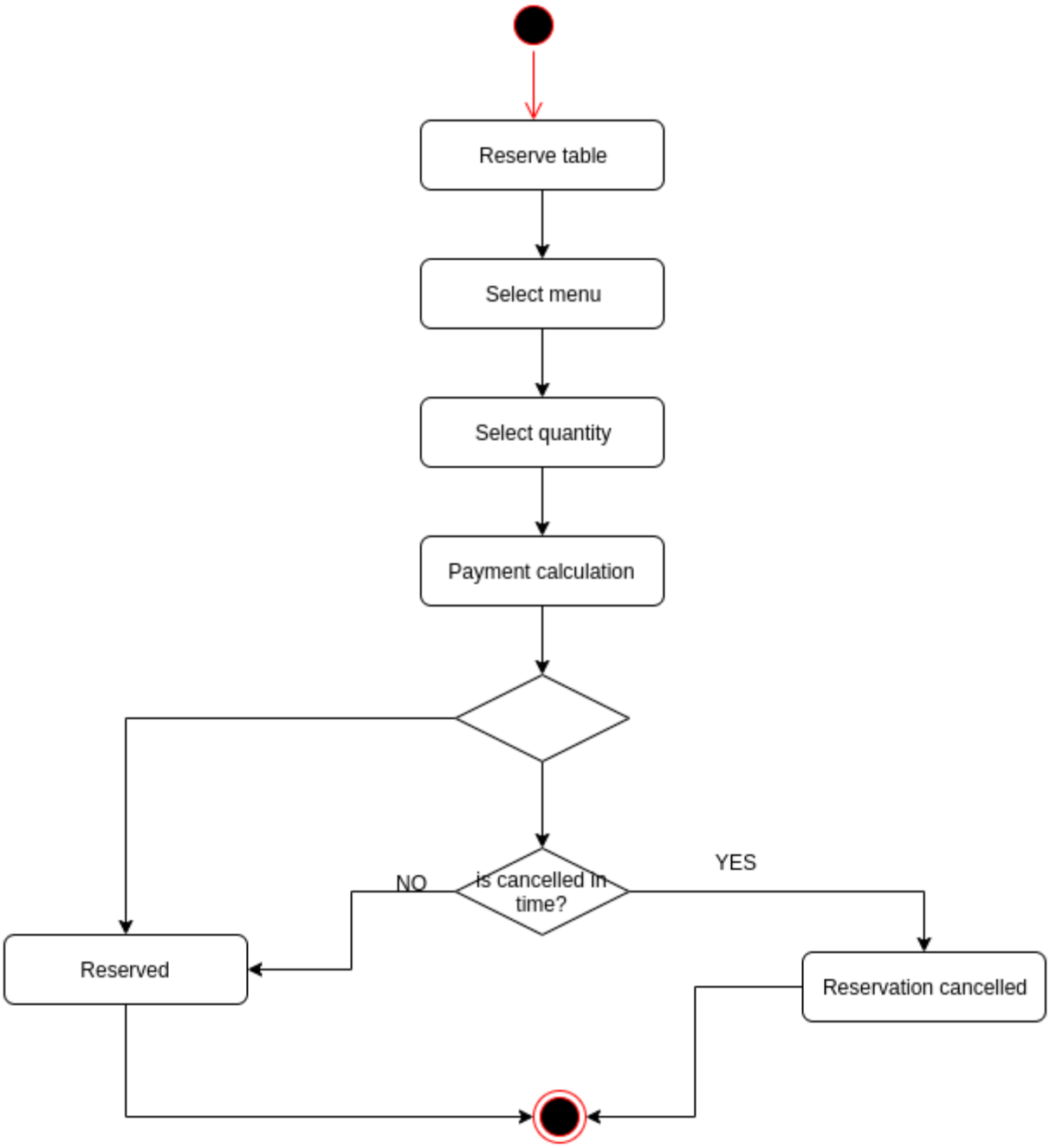
Level: 1.2
Name : Ordering
Reference: Use Case level 1.2

Payment

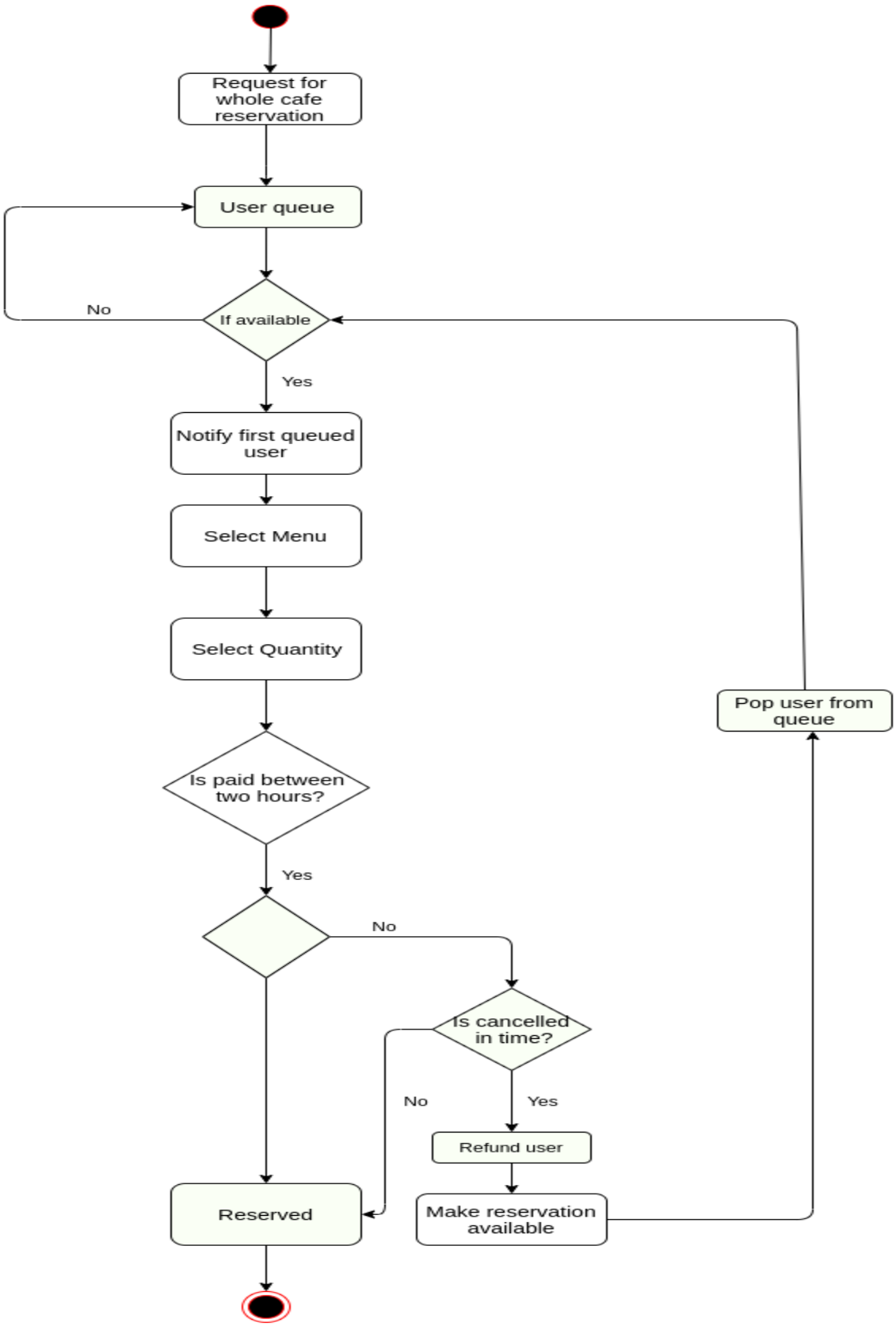


Level : 1.2.3
Name : Payment
Reference: Use Case level 1.7

Reservation

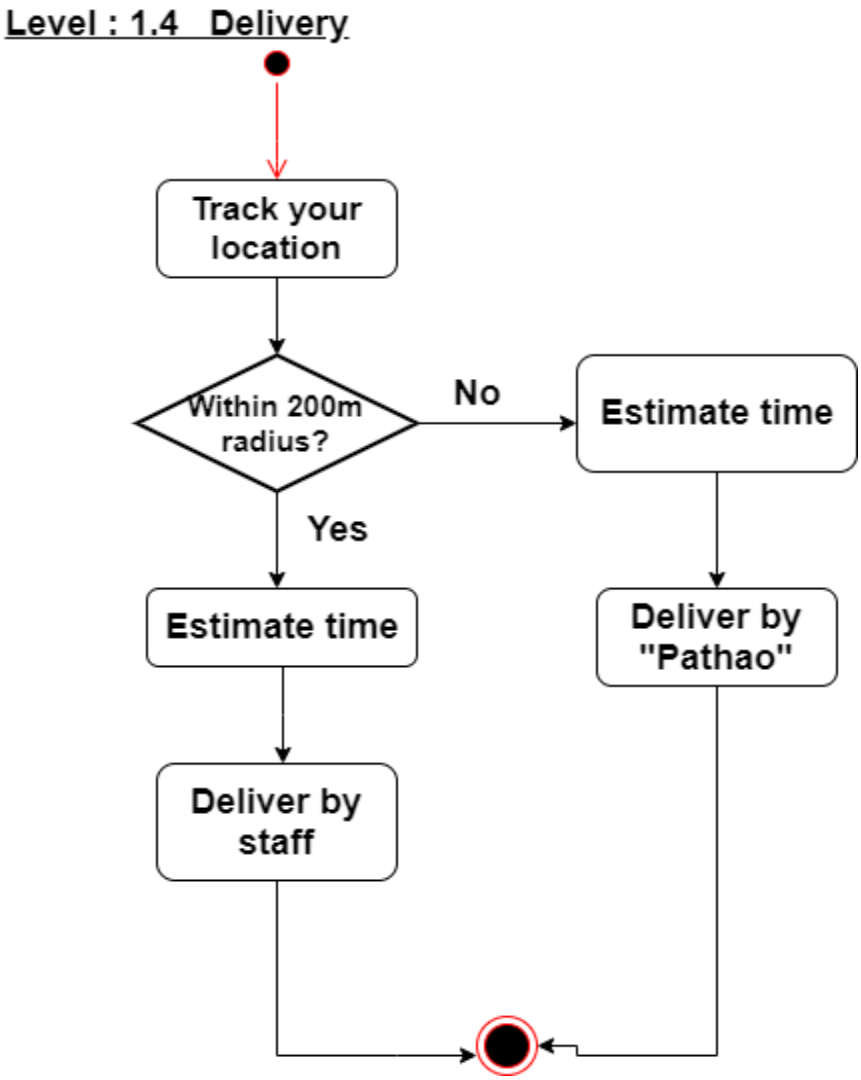


Level: 1.3.1
Name : Reservation_1
Reference: Use Case level 1.3


Request for whole cafe reservation
User queue
If available
No
Yes
Notify first queued user
Select Menu
Select Quantity
Is paid between two hours?
Yes
No
Is cancelled in time?
No
Yes
Refund user
Make reservation available
Reserved
Pop user from queue

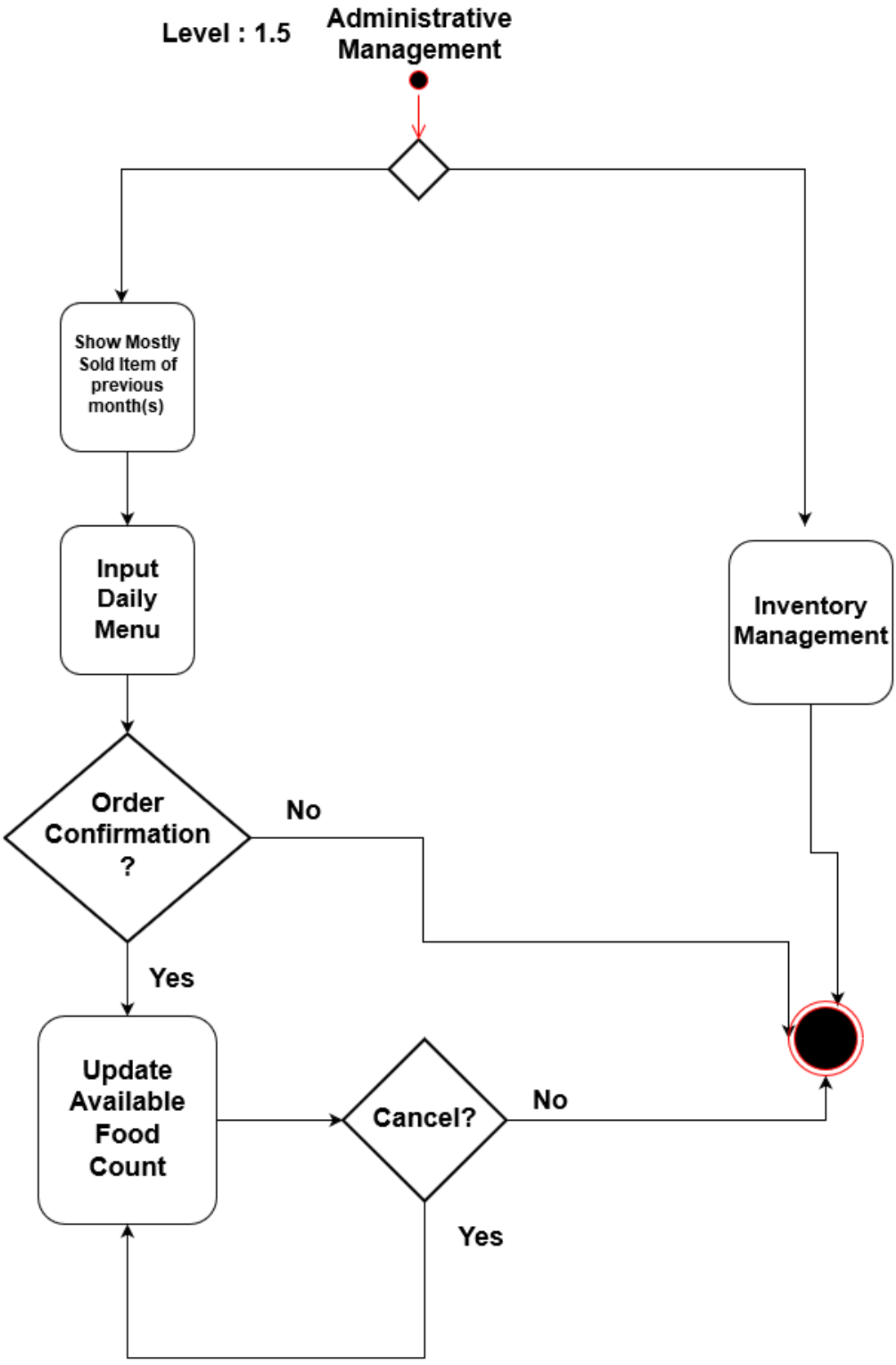
Level: 1.3.2
Name : Reservation_2
Reference: Use Case level 1.3

Delivery



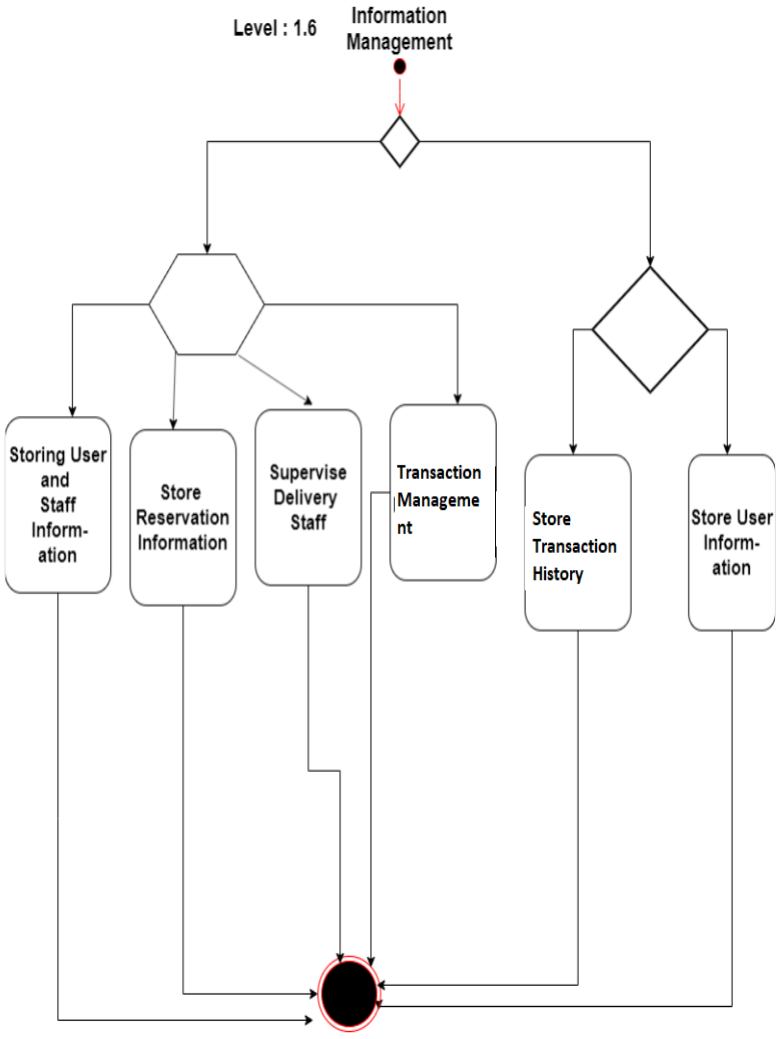
Level: 1.4
Name : Delivery
Reference: Use Case level 1.4

Administrative management



Level : 1.5
Name : Administrative management
Reference: Use Case level 1.5

Information management



Level : 1.6
Name : Information management
Reference: Use Case level 1.6

Swimlane Diagram

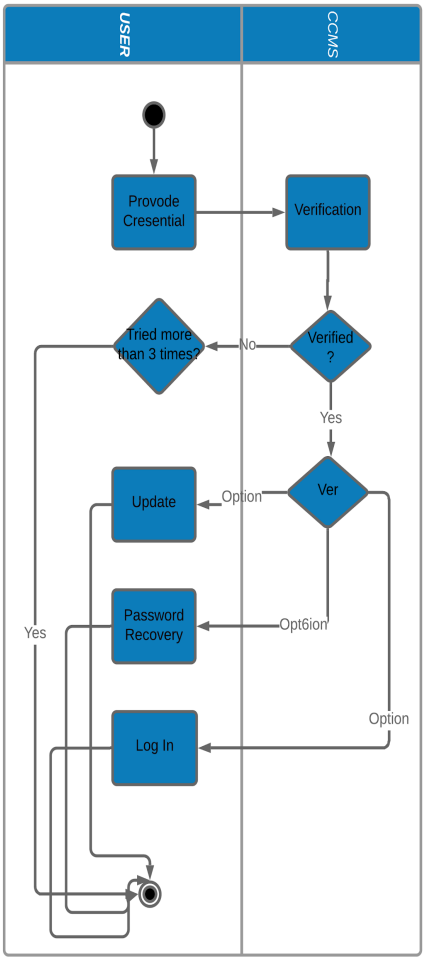
Definition :

A swimlane diagram is a type of flowchart that delineates who does what in a process. Using the metaphor of lanes in a pool, a swimlane diagram provides clarity and accountability by placing process steps within the horizontal or vertical "swimlanes" of a particular employee, work group or department. It shows connections, communication and handoffs between these lanes, and it can serve to highlight waste, redundancy and inefficiency in a process.

SID(Swimlane ID): 1.1

Name : Account

level 1.1:Account

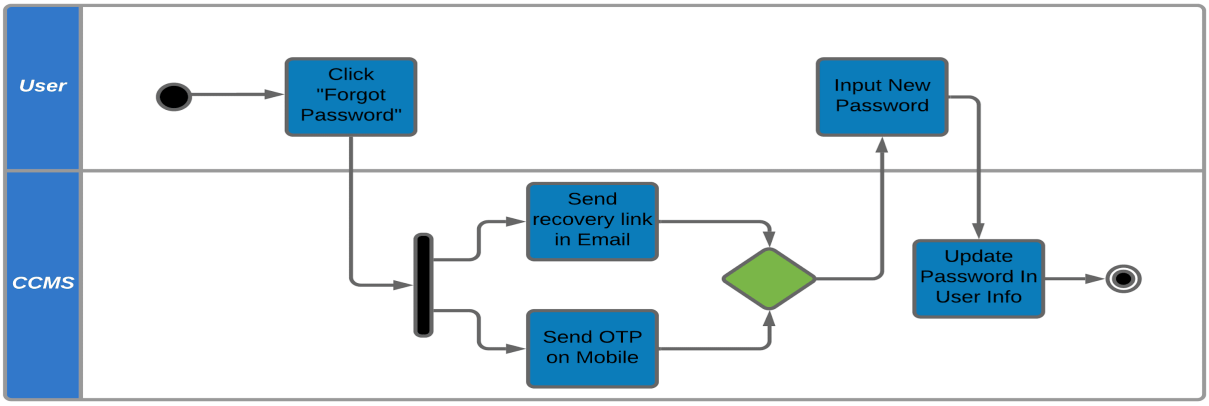


Reference: Use Case & Activity level 1.1

SID: 1.1.4

Name : Password Recovery
Reference: Use Case & Activity level 1.1.4

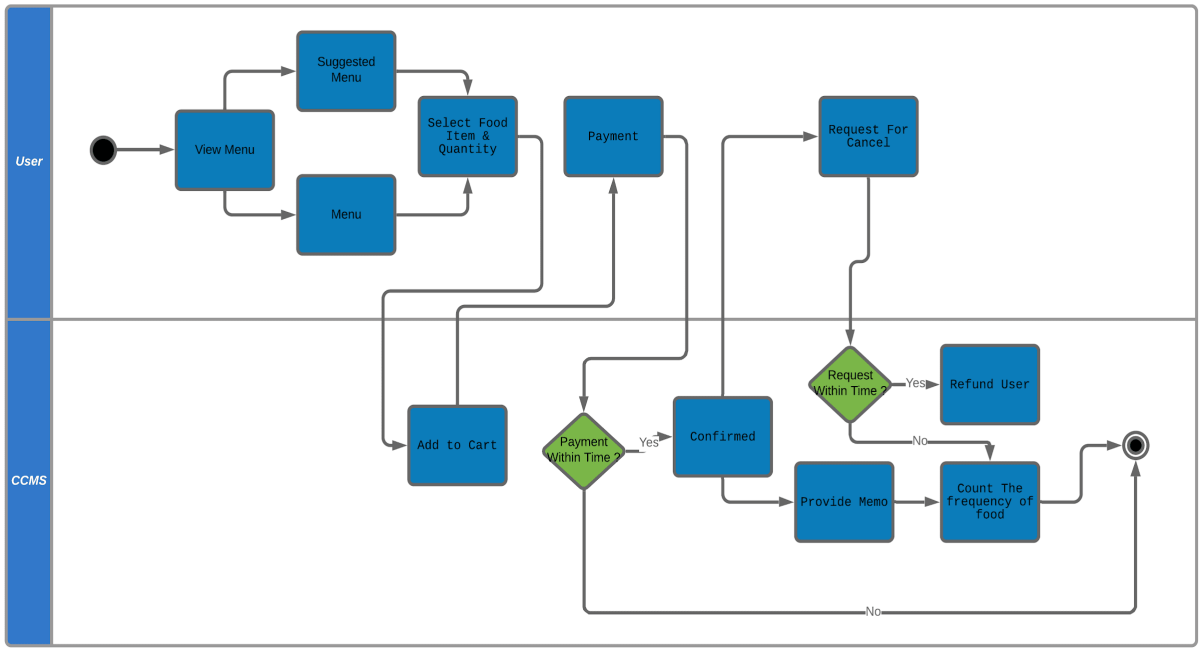
level 1.1.4:
Password Recovery



SID: 1.2

Name : Ordering
Reference: Use Case & Activity level 1.2

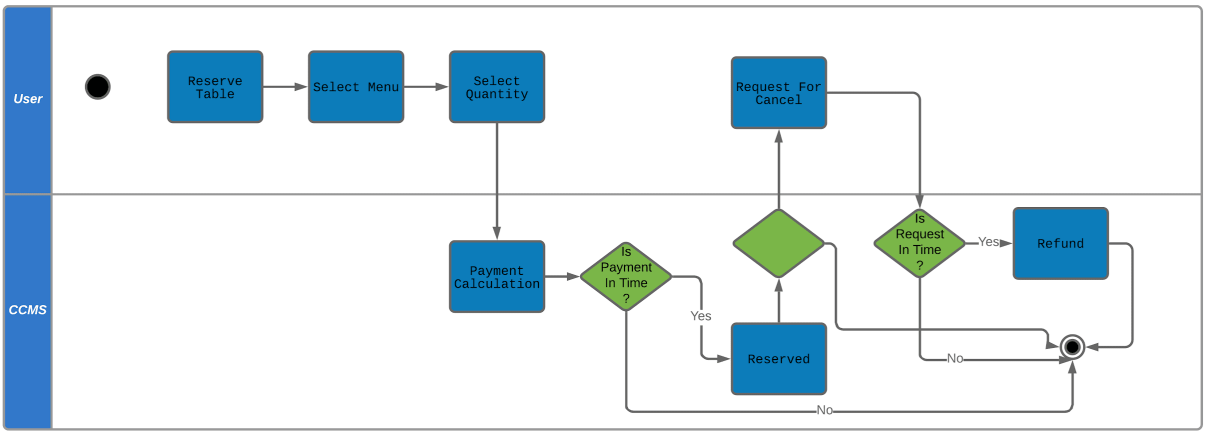
level 1.2: Ordering



SID: 1.3.1

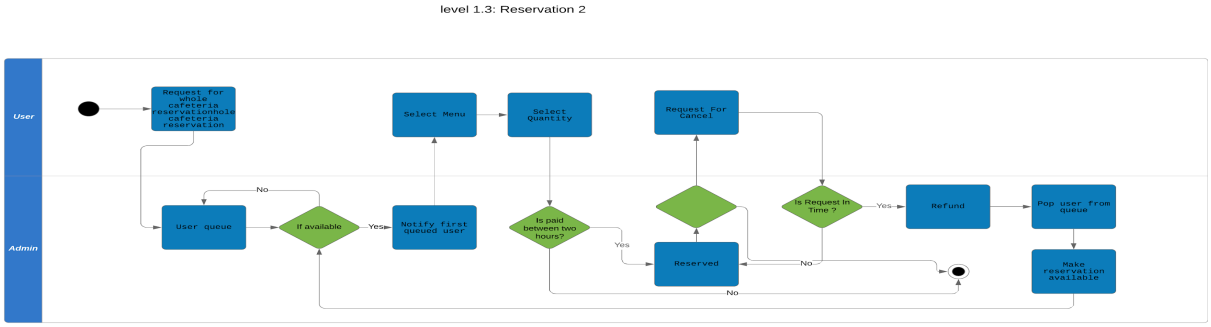
Name : Reservation 1
Reference: Use Case level 1.3 & Activity 1.3.1

level 1.3: Reservation 1



SID: 1.3.2

Name : Reservation 2

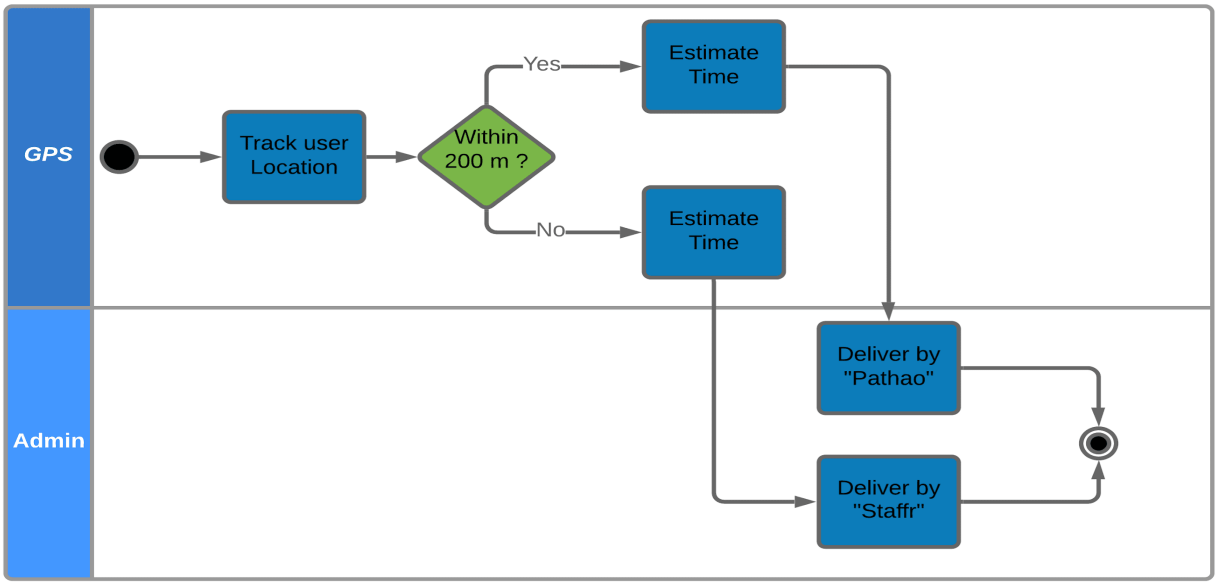


Reference: Use Case level 1.3 & Activity 1.3.2

SID: 1.4

Name : Delivery
Reference: Use Case & Activity level 1.4

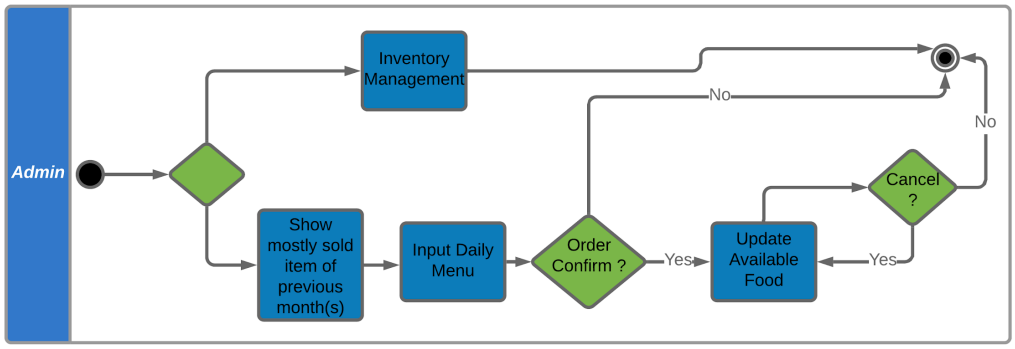
level 1.4: Delivery



SID: 1.5

Name : Administrative Management
Reference: Use Case & Activity level 1.5

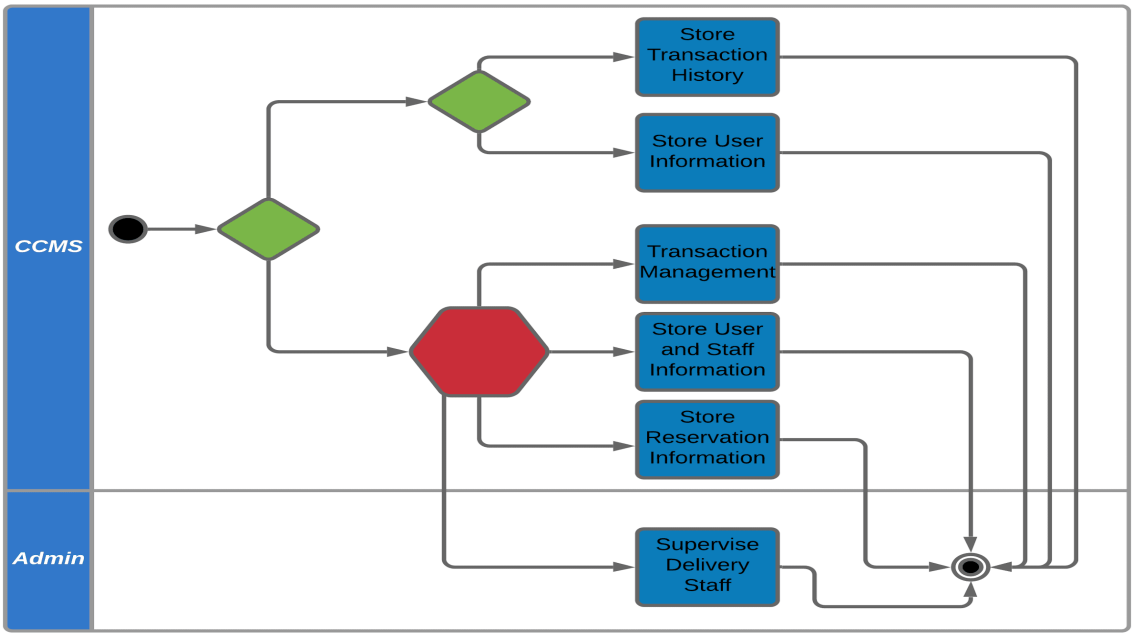
level 1.5:
Administrative
Management



SID: 1.6

Name : Information Management
Reference: Use Case level 1.6

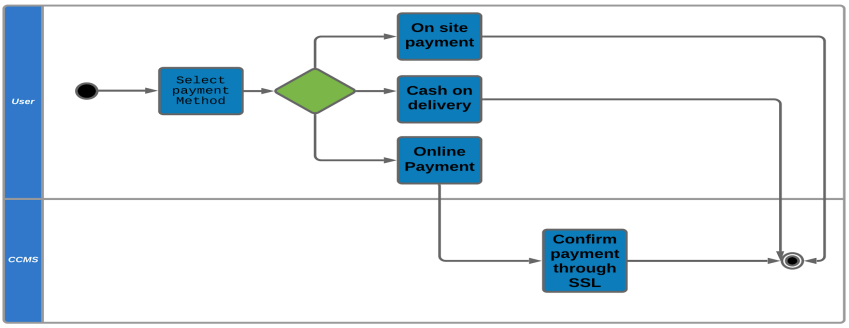
level 1.6:
Information
Management



SID: 1.7

Name : Payment
Reference: Use Case level 1.7 & Activity Level 1.2.3

level 1.7: Payment



Data Based Modelling

DATA MODELING CONCEPT :

If software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated, then the software team chooses to create data models as part of overall requirements modeling. The entity-relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects and the information about how the data objects are entered, stored, transformed and produced within the system.

DATA OBJECTS :

A data object is a representation of composite information that must be understood by the software. Here, composite information means information that has a number of different

properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

Data object identification :

Serial	Noun	Problem(p)/ solution(s) space	Attribute
1	CCMS	p	
2	CARS cafeteria	p	
3	admin	s	9,10,11,14,19
4	teacher	s	9,10,11,12,14,15,16,17, 19
5	order	s	19,34,33,79,82
6	reservation	s	80,19,51,37,38,74,79,81, 39,82
7	account	p	
8	information	p	
9	full name	s	
10	mobile number	s	
11	email address	s	
12	teacher id	s	
13	officer id	s	
14	password	s	
15	department name	s	
16	room number	s	
17	location	s	
18	Verification code	s	

continued on next page

continued from previous page

19	username	s	
20	credentials	p	
21	database	p	
22	Dhaka University	p	
23	employees	p	
24	grocery item	s	
25	SSL Commerz	s	
26	Cafeteria's transaction account	s	
27	Secured and encrypted protocol	p	
28	Notification email	s	
29	Notification sms	s	
30	user	p	
31	payment	p	
32	book table	p	
33	memo	s	34,37,38,49,39
34	Order id	s	
35	Delivery man	p	
36	Cash on delivery	p	
37	Food item	s	
38	quantity	s	
39	Total price	s	
40	office delivery	p	
41	cancellation	p	
42	transaction	s	83,84,34,80
43	profile	p	

continued on next page

continued from previous page

44	Recovery link	s	
45	OTP	s	
46	menu	s	37,38,49
47	stock	s	
48	Virtual cart	s	
49	price	s	
50	delivery	p	
51	Table number	s	
52	occasion	p	
53	ceremony	p	
54	event	p	
55	Booking queue	p	
56	First place	p	
57	Queued user	p	
58	Room delivery	p	
59	University campus	p	
60	service	p	
61	Delivery charge	s	
62	Estimated time	s	
63	User's display	p	
64	staff	s	
65	GPS	s	
66	"Pathao" rider	p	
67	Admin database	s	
68	User database	s	
69	Delivery staff number	s	

continued on next page

continued from previous page

70	Availability count	s	
71	Sum of transactions	s	
72	Transaction history	p	
73	Replica of user info	p	
74	Quantity of people	s	
75	Administrative management	s	
76	Inventory management	s	24,38
77	verification	s	
78	Officer	s	9,10,11,13,14,15,16,17, 19
79	Confirmation Time	s	
80	Reservation id	s	
81	Event time	s	
82	Request time	s	
83	Transaction id	s	
84	Transaction amount	s	

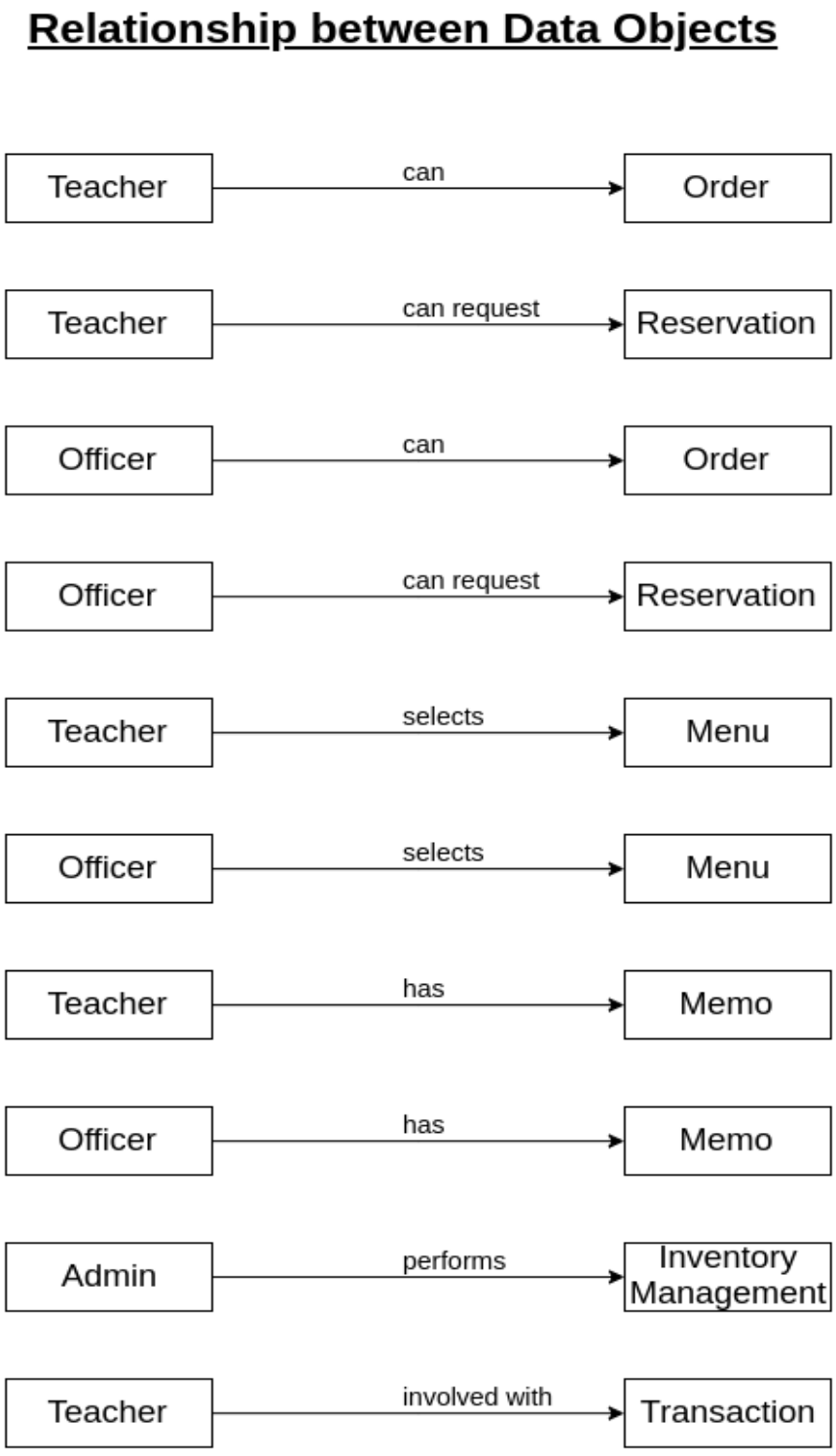
Final data object :

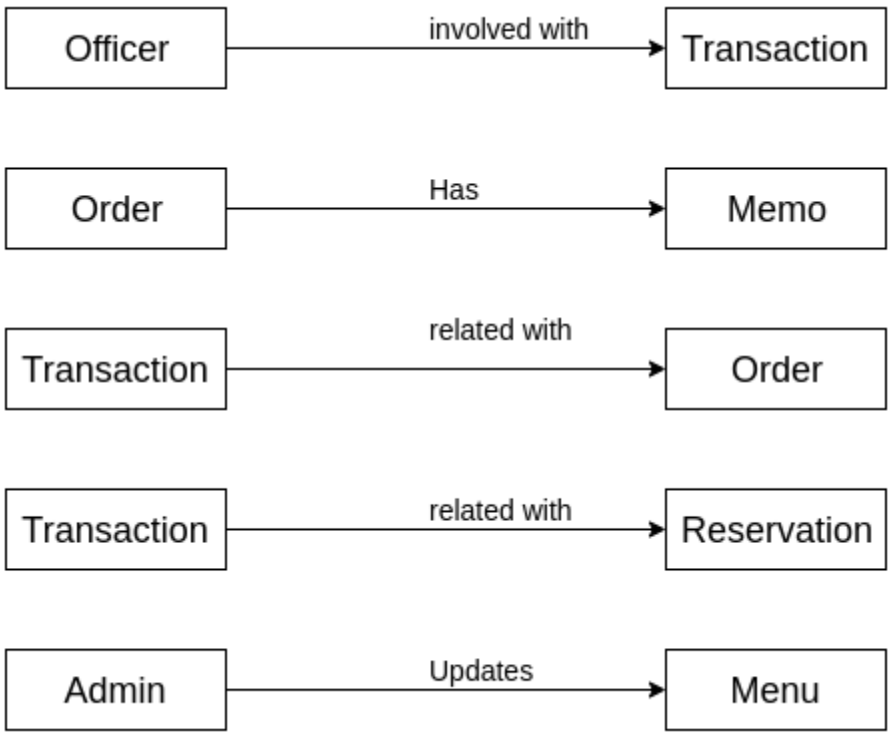
1) Admin
2) Teacher
3) Officer
4) Order
5) Reservation
6) Memo
7) Transaction

8) Menu

9) Inventory management

Data Object Relationship:



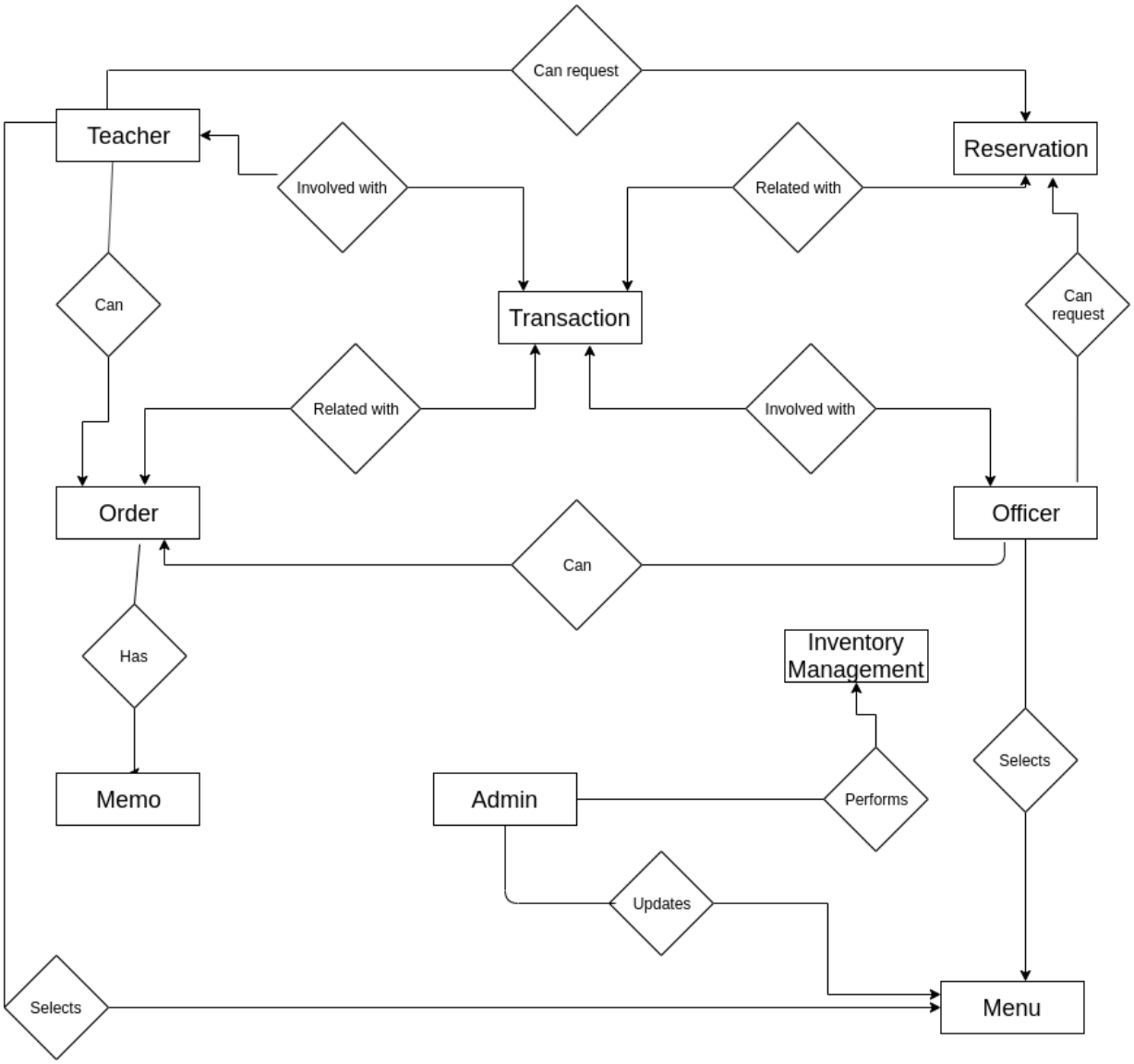


ER Diagram:

Definition of ER Diagram

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system.

ER Diagram


Can request
Teacher
Reservation
Involved with
Related with
Can
Transaction
Can request
Related with
Involved with
Order
Officer
Can
Has
Inventory Management
Selects
Memo
Admin
Performs
Updates
Selects
Menu

Schema Diagram

Data Object	Attribute	Type	Size
Admin	-full name -mobile number -email address -username -password	Varchar Varchar Varchar Varchar Varchar	40 40 40 40 40
Teacher	-full name -mobile number -email address -teacher id -password -department name -room number -location -username	Varchar Varchar Varchar Varchar Varchar Varchar Number Varchar Varchar	40 40 40 40 40 40 40 40
Officer	-full name -mobile number -email address -officer id -password -department name -room number -location -username	Varchar Varchar Varchar Varchar Varchar Varchar Number Varchar Varchar	40 40 40 40 40 40 40 40
Order	-Order id -username -memo -Request time -Confirmation time	Varchar Varchar Memo Date & time Date & time	40 40
Reservation	-Reservation id -username -food item -quantity -total price -table number -quantity of people -confirmation time -event time -request time	Varchar Varchar List Number Number Number Number Time Time Time	40 40
Memo	-order id -food item	Varchar Varchar	40 40

CLASS-BASED MODELING

CLASS BASED MODELING CONCEPT :

Class-based modeling represents the objects that the system will manipulate, the operations that will be applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

Noun list from Cafeteria Management System

No	Noun	No	Noun
1	CCMS	20	credentials
2	CARS cafeteria	21	database
3	admin	22	Dhaka University
4	teacher	23	employees
5	order	24	grocery item
6	reservation	25	SSL Commerz
7	account	26	Cafeteria's transaction account
8	information	27	Secured and encrypted protocol
9	full name	28	Notification email
10	mobile number	29	Notification sms
11	email address	30	user
12	teacher id	31	payment
13	officer id	32	book table
14	password	33	memo
15	department name	34	Order id
16	room number	35	Delivery man
17	location	36	Cash on delivery
18	Verification code	37	Food item

continued on next page

continued from previous page

19	username	38	quantity
39	Total price	59	University campus
40	office delivery	60	service
41	cancellation	61	Delivery charge
42	transaction	62	Estimated time
43	profile	63	User's display
44	Recovery link	64	staff
45	OTP	65	GPS
46	menu	66	"Pathao" rider
47	stock	67	Admin database
48	Virtual cart	68	User database
49	amount	69	Delivery staff number
50	delivery	70	Availability count
51	Table number	71	Sum of transactions
52	occasion	72	Transaction history
53	ceremony	73	Replica of user info
54	event	74	Quantity of food
55	Booking queue	75	Administrative management
56	First place	76	Inventory management
57	Queued user	77	verification
58	Room delivery	78	Officer
		79	reservation id

Verb list

No	Verb	No	Verb
1	Create Account	31	show(stock)
2	Provides info	32	choose item
3	verify info	32	select(quantity)
4	send code	33	add(to cart)
5	input code	34	pay(bill)
6	update info	35	deliver
7	change	36	confirm(order)
8	recover (password)	37	give(memo)
9	send(recovery link)	38	cancel(order)
10	click(link)	39	refund
11	input(password)	40	reserve
12	update(database)	41	book(table,cafeteria)
13	send(otp)	42	choose(table)
14	input(new otp)	43	notify(cancellation)
15	log in	44	assign(to booking queue)
16	order	45	reopen(booking queue)
17	provide(menu)	46	communicate
18	integrate	47	store (staff and user info)
19	send(transaction email)	48	store (reservation details)
20	add (admin database)	49	count (staff)
21	go (to cafe)	50	deduct

continued on next page

continued from previous page

22	choose (payment type)	51	come
23	cancel (automatically)	52	see
24	prompt memo	53	add (transaction detail)
25	input (grocery item)	54	store (user database)
26	update (count)	55	update (user database)
27	request (delivery)	56	replace(admin database)
28	prompt (estimated time)	57	fix
29	track	58	update menu
30	display	59	deduct (food item)

General classification

Candidate classes were then characterized in seven general classification. The seven general characteristics are as follows:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

Potential nouns to become a class after general classification criteria :

Noun	General Classification
CCMS	4
admin	4,7
teacher	4,7
officer	4,7
order	3
reservation	3
account	2,7
delivery	3
cancellation	3
transaction	3
profile	2,7
menu	1,2
stock	2
virtual cart	2
booking queue	2
database	2,7
admin database	2,7
user database	2,7
employee	4,5,7
grocery item	1
SSL Commerz	1,3
Cafeteria's transaction account	1
email	1,3
sms	1,3
user	4,5,7

continued on next page

continued from previous page

payment	3
memo	2
cash on delivery	3
staff	4,5,7
GPS	1
Replica of user info	2
Administrative management	3
Inventory management	3
verification	3

Selection Criteria

The candidate classes are then selected as classes by six Selection Criteria. A candidate class generally becomes a class when it fulfills around three characteristics.

1. Retain information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

Potential general classified nouns to become a class after selection criteria :

continued on next page

continued from previous page

Noun	Selection criteria
CCMS	1,2,6 (selected)
admin	1-6 (selected)
teacher	1-6 (selected)
officer	1-6 (selected)
order	1,2,3,6 (selected)
reservation	1,2,3,6 (selected)
account	1-6 (selected)
delivery	2,3,6 (selected)
cancellation	
transaction	1,3
profile	1,3,4
menu	1
stock	1
virtual cart	1,3,2 (selected)
booking queue	2
admin database	1-6 (selected)
user database	1-6 (selected)
employee	
grocery item	
SSL Commerz	2, 6 (selected)
Cafeteria's transaction account	1,6 (selected)
email	2,5,6 (selected)
sms	2,5,6 (selected)
user	1-6
payment	3,4,5,6 (selected)

continued on next page

continued from previous page

memo	1
cash on delivery	
staff	
GPS	6 (selected)
Replica of user info	
Administrative management	6
Inventory management	
verification	2,6

Attribute and Method Identification

Class name	Attribute	Method

continued on next page

continued from previous page

CCMS		+verify_info() +notify_user() +send_otp() +send_link() +update_user_database() +update_admin_database() +provide_menu() +display_stock() +add_to_cart() +provide_memo() +notify_after_transaction() +prompt_memo() +update_item_count() +display_most_consumed_food() +display_relevant_food() +analyze_order()
Admin	-full_name - mobile_number - email_address - user_name -password	+login() +notify_after_cancellation() +notify_after_confirmation() +input_grocery_item() +update_grocery_item() +fix_menu() +update_menu() +input_staff_count() +update_staff_count() +call_delivery()

continued on next page

continued from previous page

Teacher	-full_name -mobile_number -email_address -teacher_id -department_name -room_number -location -password	+create_account() +recover_password() +update_info() +login() +order() +choose_payment_type() +request_delivery() +pay() +cancel() +reserve()
Officer	-full_name -mobile_number -email_address -officer_id -department_name -room_number -location -password	+create_account() +recover_password() +update_info() +login() +order() +choose_payment_type() +request_delivery() +pay() +cancel() +reserve()
Order	-order_id -food_item_with_price -quantity -total_price -isConfirmed -isCancelled -payment_type -order_time -isRefunded	+display_menu() +choose_item() +select_quantity() +store_order_detail() +auto_cancel() +refund_user()

continued on next page

continued from previous page

Reservation	-reservation id -total_people -total_table -reservation_type -menu -isInQueue -isConfirmed -isCancelled -payment_type -reservation_time	+reserve_table() +reserve_whole_cafe() +customize_menu() +manage_queue() +refund() +store_reservation_detail() +auto_cancel() +calculate_amount()
Account	-full_name -mobile_number -email_address -teacher_id/officer_id -department_name -room_number -location -password	+getFull_name() +setFull_name() +getMobile_number() +setMobile_number() +getEmail_address() +setEmail_address() +getTeacher/officer_id() +setTeacher/officer_id() +getDepartment_name() +setDepartment_name() +getRoom_number() +setRoom_number() +getLocation() +setLocation() +getPassword() +setPassword()
Delivery	-delivery_time -delivery_number -delivery_location	+staff_delivery() +pathao_delivery()

continued on next page

continued from previous page

Admin_ Database	-staff_number -memo_number	+transaction_history() +store_staffInfo() +update_staff() +calc_transaction() +store_groceryInfo()
User_ Database	-memo_number	+transaction_history() +update_info()
Virtual_Cart	-food_quantity -total_amount -estimated_deliveryTime -delivery_location -contact_info	+calculate_amount() +prompt_virtual_cart()
SSL_ COMMERZ	-isTransaction_ completed -isNotified	+make_transaction() +notify_admin() +add_payment_to_account()
Cafeteria's_ Transaction_ Account	-transaction_ dailyCount -transaction_ weeklyCount -transaction_ monthlyCount	+update_ dailyCount() +update_ weeklyCount() +update_ monthlyCount()
Email		+send_confirmation()
SMS		+send_confirmation()
Payment	-payment_type -isPayment_completed -payment_time -transaction_id	+notify_user() +online_payment() +cash_on_delivery()
GPS	-distance -estimated_time	+call_GPS() +display_distance_time()

Analysis

All classes included in class based diagram are selected as class for our system.

CRC card

Class name	Responsibility	Collaborator

continued on next page

continued from previous page

CCMS	• verifying information • notifying user • sending otp and recovery link • updating user and admin database • providing menu • displaying stock • adding to cart • providing memo • notifying after transaction • prompting memo • deducting item • Displaying mostly consumed food • Displaying all relevant food • Analyzing order of previous month	Teacher, Officer, SMS, Email, Admin_database, User_database, Admin, Virtual_cart, Order, Payment

continued on next page

continued from previous page

Admin	• Inventory management • notifying after confirmation and cancellation • managing daily menu • staff management • invoking delivery method	Teacher, Officer, Admin_database, SMS, Email, Cafeteria's_ Transaction_account
Teacher	• Ordering food • Payment • Reservation	Account, Payment, Order, Reservation, Delivery
Officer	• Ordering food • Payment • Reservation	Account, Payment, Order, Reservation, Delivery

continued on next page

continued from previous page

Order	• Display menu • Storing order details • Cancelling order • Refunding order	Teacher, Officer, User_database, Admin_database, Payment
Reservation	• Managing reservation • Managing booking queue • Refunding user • Storing reservation details • Cancelling reservation • Calculating reservation payment	Teacher, Officer, User_database, Admin_database, Payment
Account	• Creating account • Updating profile • Viewing profile	

continued on next page

continued from previous page

Delivery	• Managing delivery methods	GPS, CCMS
Admin_Database	• Storing transaction history • Storing staff Info • Storing reservation details • Updating staff count • Calculating period wise transaction • Storing Grocery Info	Payment, Order, Reservation, Account, Admin , CCMS
User_Database	• Storing user's transaction history • Storing updated info	Payment, Order, Reservation, Account, CCMS

continued on next page

continued from previous page

Virtual_Cart	• Calculating amount • Prompting virtual cart	Order, Teacher, Officer
SSL_COMMERZ	• Making transaction • Notifying admin about payment completion • Adding paid amount to cafeteria's transaction account	Cafeteria's_ Transaction_Account, Payment, Admin
Cafeteria's_Transaction_Account	• Storing daily transaction count • Prompting period wise count	Payment
Email	• Sending confirmation email	

continued on next page

continued from previous page

SMS	• Sending confirmation sms	
Payment	• Managing user's payment method	SSL_COMMERZ, Teacher, Officer, SMS, Email
GPS	• Calling external subsystem "GPS"	

CLASS CARDS

After identifying our final classes we have generated the following class cards.

Table: Class Card for CCMS Class

CCMS	
Attribute	Method
	+verify_info() +notify_user() +send_otp() +send_link() +update_user_database() +update_admin_database() +provide_menu() +display_stock() +add_to_cart() +provide_memo() +notify_after_transaction() +prompt_memo() +update_item_count() +deduct_staff_count() +display_most_consumed_food() +display_relevant_food() +analyze_order()
Responsibilities	Collaborator
• verifying information • notifying user • sending otp and recovery link • updating user and admin database • providing menu • displaying stock • adding to cart • providing memo • notifying after transaction • prompting memo	Teacher Officer SMS Email Admin_Database User_Database Admin Virtual_cart Order Payment

Table: Class Card for Admin Class

Admin	
Attribute	Method
-full_name -mobile_number -email_address -user_name -password	+login() +notify_after_cancellation() +notify_after_confirmation() +input_grocery_item() +update_grocery_item() +fix_menu() +update_menu() +input_staff_count() +store_order_ reservation_detail() +update_staff_count() +call_delivery()
Responsibilities	Collaborator
• Inventory management • notifying after confirmation and cancellation • managing daily menu • staff management • invoking delivery method	Teacher Officer Admin_Database SMS Email Cafeteria's_ Transaction_Account

Table: Class Card for Teacher Class

Teacher	
Attribute	Method
-full_name -mobile_number -email_address -teacher_id -department_name -room_number -location -password	+create_account() +recover_password() +update_info() +login() +order() +choose_payment_type() +request_delivery() +pay() +cancel() +reserve()
Responsibilities	Collaborator
• Ordering food • Payment • Reservation	Account Payment Order Reservation Delivery

Table: Class Card for Officer Class

Officer	
Attribute	Method
-full_name -mobile_number -email_address -officer_id -department_name -room_number -location -password	+create_account() +recover_password() +update_info() +login() +order() +choose_payment_type() +request_delivery() +pay() +cancel() +reserve()
Responsibilities	Collaborator
• Ordering food • Payment • Reservation	Account Payment Order Reservation Delivery

Table: Class Card for Order Class

Order	
Attribute	Method
-order_id - food_item_with_price - quantity -total_price -isConfirmed -isCancelled -payment_type -order_time -isRefunded	+display_menu() +choose_item() +select_quantity() +store_order_detail() +auto_cancel() +refund_user()
Responsibilities	Collaborator
• Display menu • Storing order details • Cancelling order • Refunding order	Teacher Officer User_Database Admin_Database Payment

Table: Class Card for Reservation Class

Reservation	
Attribute	Method
-total_people -total_table -reservation_type -menu -isInQueue -isConfirmed -isCancelled -payment_type -reservation_time	+reserve_table() +reserver_whole_cafe() +customize_menu() +manage_queue() +refund() +store_reservation_detail() +auto_cancel() +calculate_amount()
Responsibilities	Collaborator
• Managing reservation • Managing booking queue • Refunding user • Storing reservation details • Cancelling reservation • Calculating reservation payment	Teacher Officer User_Database Admin_Database Payment

Table: Class Card for Account Class

Account	
Attribute	Method
-full_name -mobile_number - email_address - teacher_id/officer_id - department_name - room_number -location -password	+getFull_name() +setFull_name() +getMobile_number() +setMobile_number() +getEmail_address() +setEmail_address() +getTeacher/officer_id() +setTeacher/officer_id() +get-Department_name() +setDepartment_name() +getRoom_number() +setRoom_number() +getLocation() +setLocation() +getPassword() +setPassword()
Responsibilities	Collaborator
• Creating account • Updating profile • Viewing profile	

Table: Class Card for Delivery Class

Delivery	
Attribute	Method
-delivery_time -delivery_number - delivery_location	+staff_delivery() +pathao_delivery()
Responsibilities	Collaborator
• Managing delivery methods	GPS CCMS

Table: Class Card for Admin_Database Class

Admin_Database	
Attribute	Method
-staff_number -memo_number	+transaction_history() +store_staffInfo() +reservation_details() +update_staff() +calc_transaction() +store_groceryInfo()
Responsibilities	Collaborator
• Storing transaction history • Storing staff Info • Storing reservation details • Updating staff count • Calculating period wise transaction • Storing grocery info	Payment Order Reservation Account Admin CCMS

Table: Class Card for User_Database class

User_Database	
Attribute	Method
-memo_number	+transaction_history() +update_info()
Responsibilities	Collaborator
• Storing user's transaction history • Storing updated info	Payment Order Reservation Account CCMS

Table: Class Card for VIrtual_Cart Class

Virtual_Cart	
Attribute	Method
-food_quantity -total_amount -estimated_deliveryTime -delivery_location -contact_info	+calculate_amount() +prompt_virtual_cart()
Responsibilities	Collaborator
• Calculating amount • Prompting virtual cart	Order Teacher Officer

Table: Class Card for SSL_COMMERZ Class

SSL_COMMERZ	
Attribute	Method
-isTransaction_completed -isNotified	+make_transaction() +notify_admin() +add_payment_to_account()
Responsibilities	Collaborator
• Making transaction • Notifying admin about payment completion • Adding paid amount to cafeteria's transaction account	Cafeteria's_ Transaction_Account , Payment, Admin

Table: Class Card for Cafeteria's_Transaction_Account Class

Cafeteria's_Transaction_Account	
Attribute	Method
-transaction_dailyCount -transaction_weeklyCount -transaction_monthlyCount	+update_dailyCount() +update_weeklyCount() +update_monthlyCount()
Responsibilities	Collaborator
• Storing daily transaction count • Prompting period wise count	Payment

Table: Class Card for Email Class

Email	
Attribute	Method
	+send_confirmation()
Responsibilities	Collaborator
• Sending confirmation email	

Table: CLass Card for SMS Class

SMS	
Attribute	Method
	+send_confirmation()
Responsibilities	Collaborator
• Sending confirmation SMS	

Table: Class Card for Payment Class

Payment	
Attribute	Method
-payment_type -isPayment_completed -payment_time -transaction_id	+notify_user() +online_payment() +cash_on_delivery()
Responsibilities	Collaborator
• Managing user's payment method	SSL_COMMERZ Teacher Officer SMS Email

Table: Class Card for GPS Class

GPS	
Attribute	Method
-distance -estimated_time	+call_GPS() +display_distance_time()
Responsibilities	Collaborator
• Calling external subsystem "GPS"	

CRC Diagrams

Diagram ID: 1

Name: CCMS

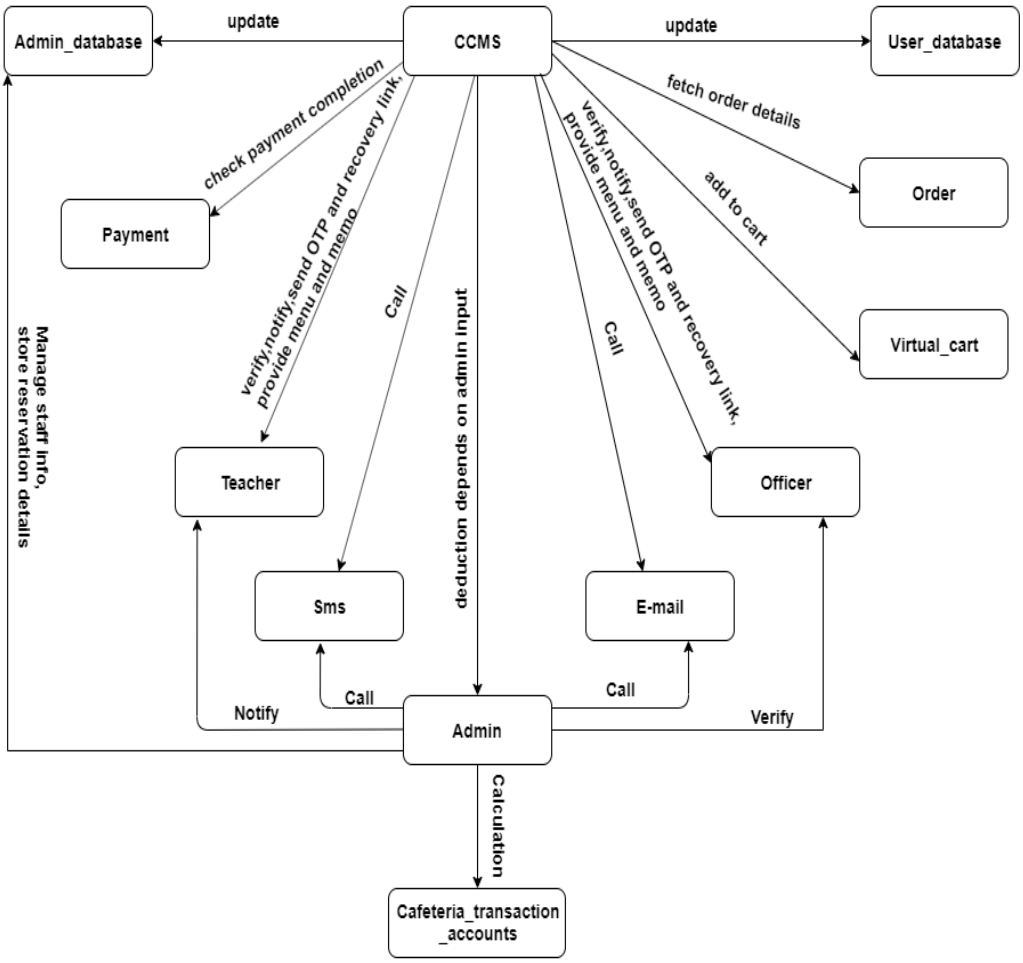


Diagram ID: 2

Name: Account

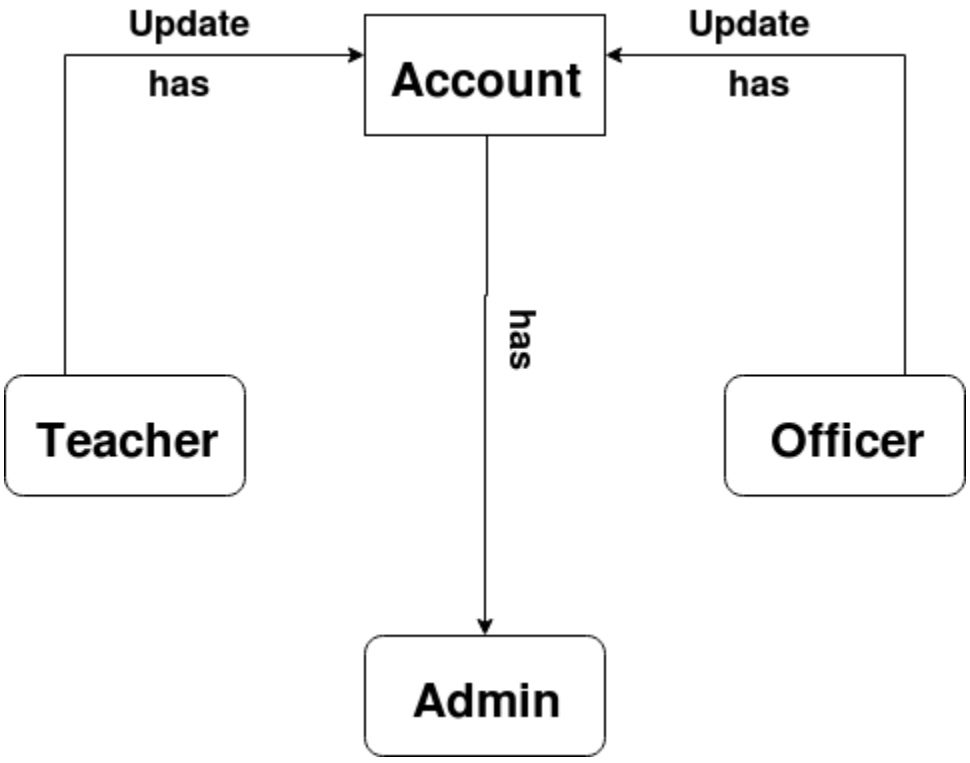


Diagram ID: 3

Name: Order

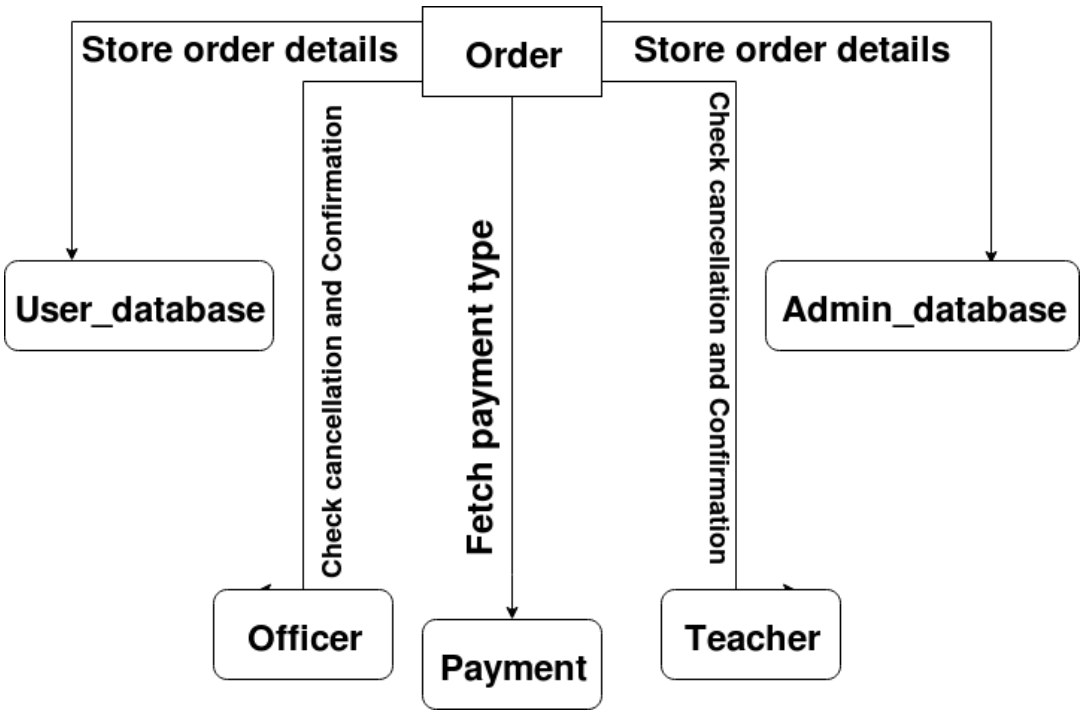


Diagram ID: 4

Name: Payment

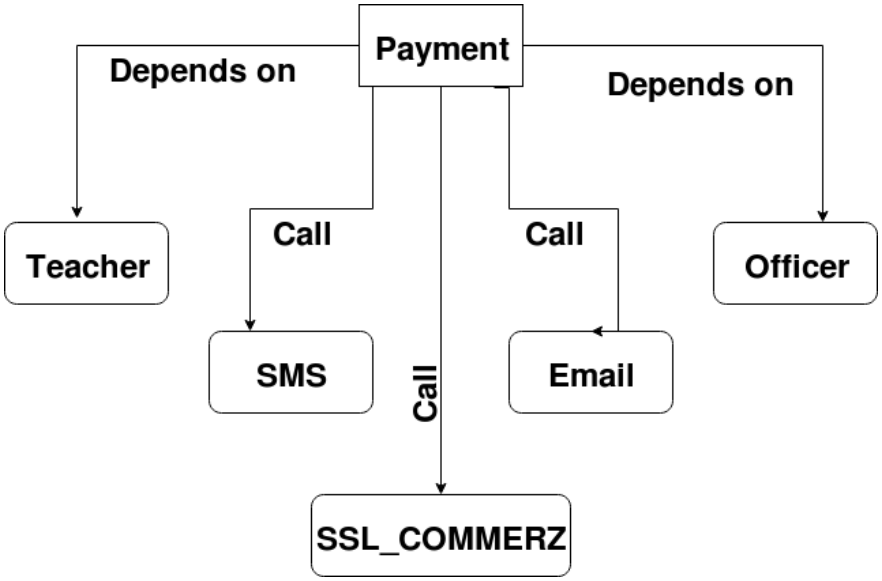


Diagram ID: 5

Name: Reservation

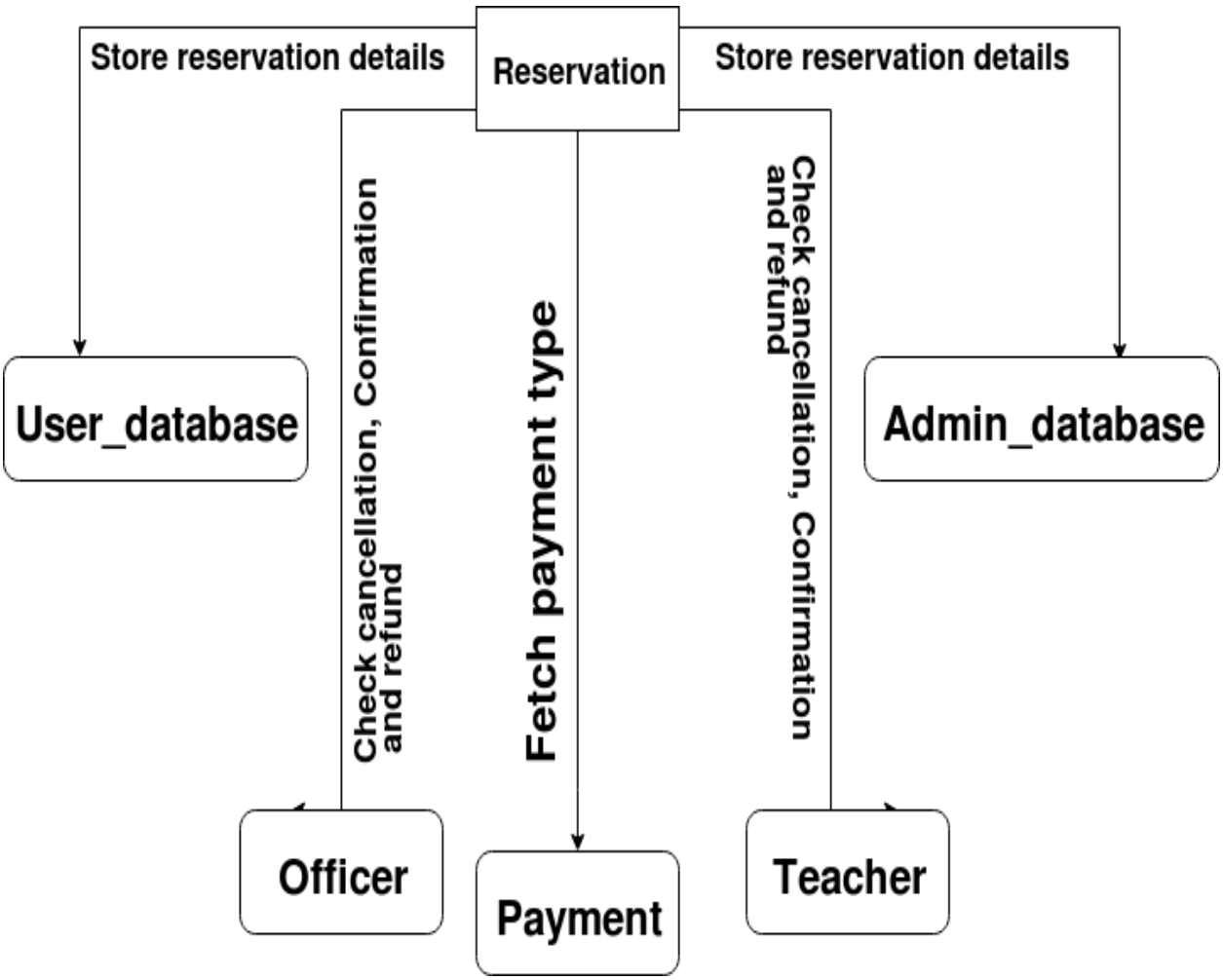


Diagram ID: 6

Name: Virtual Cart

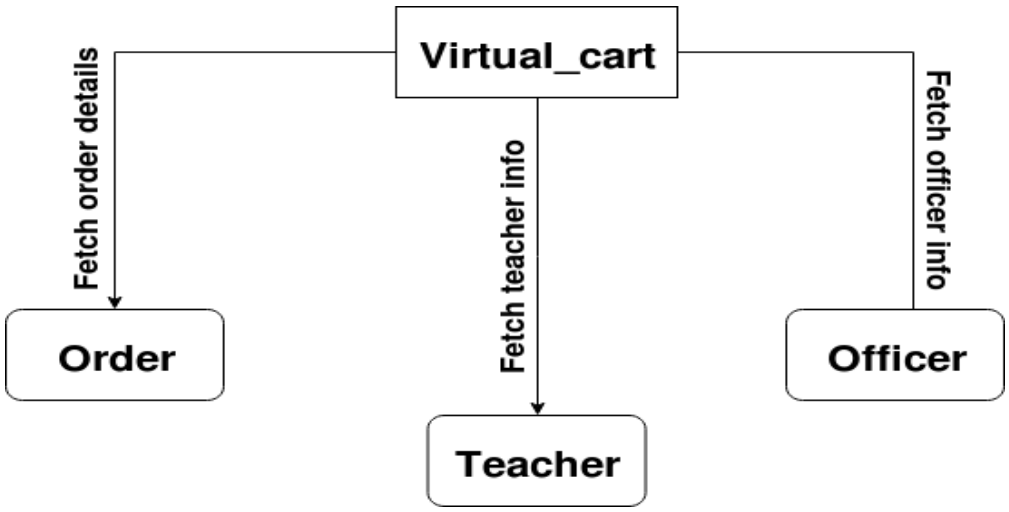


Diagram ID: 7

Name: Cafeteria's Transaction Account

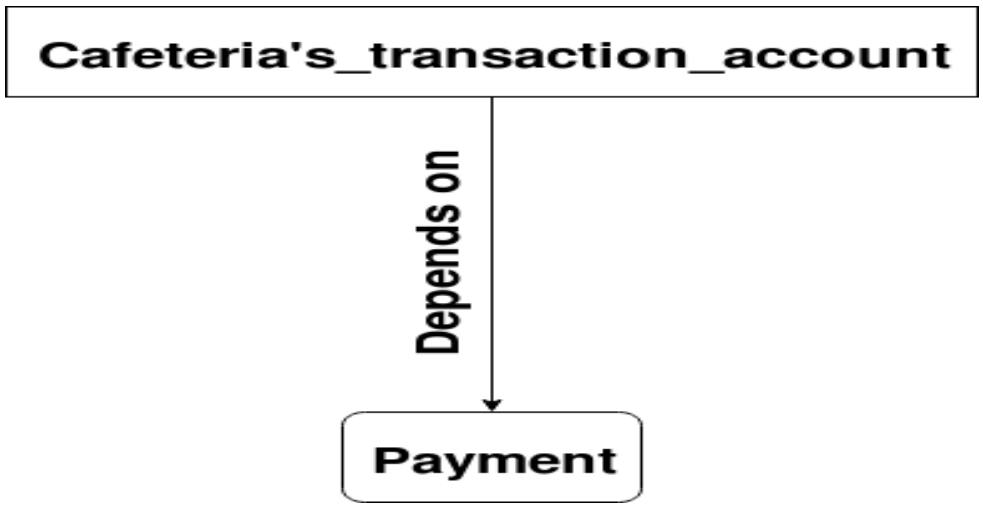


Diagram ID: 8

Name: Delivery

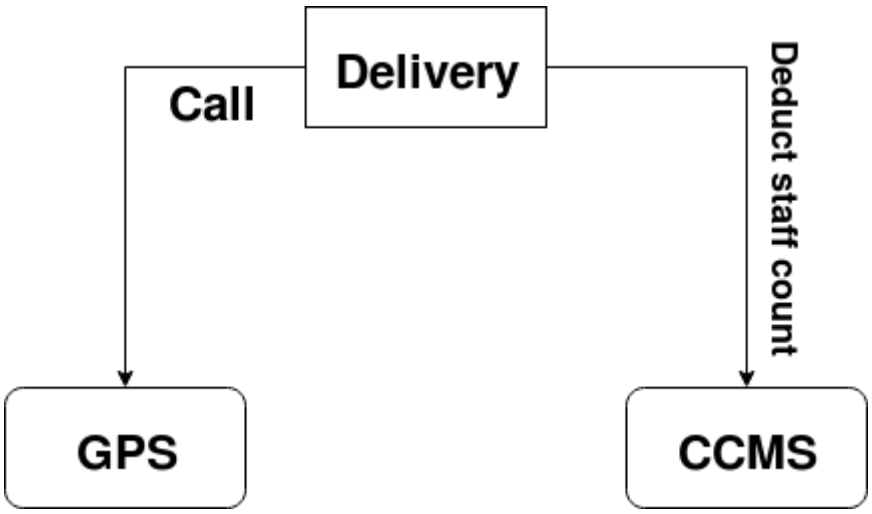


Diagram ID: 9

Name: User database

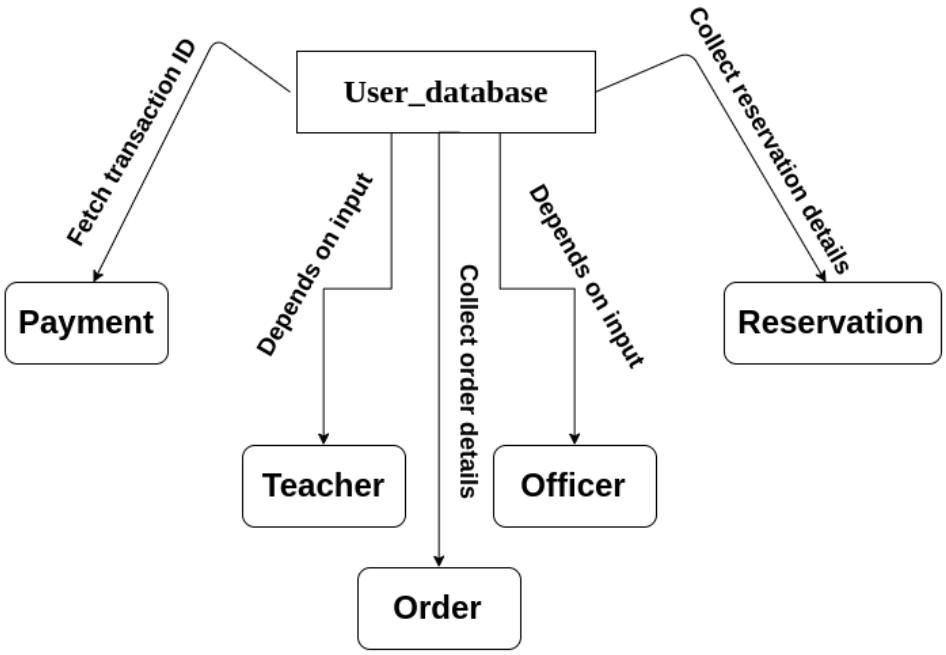
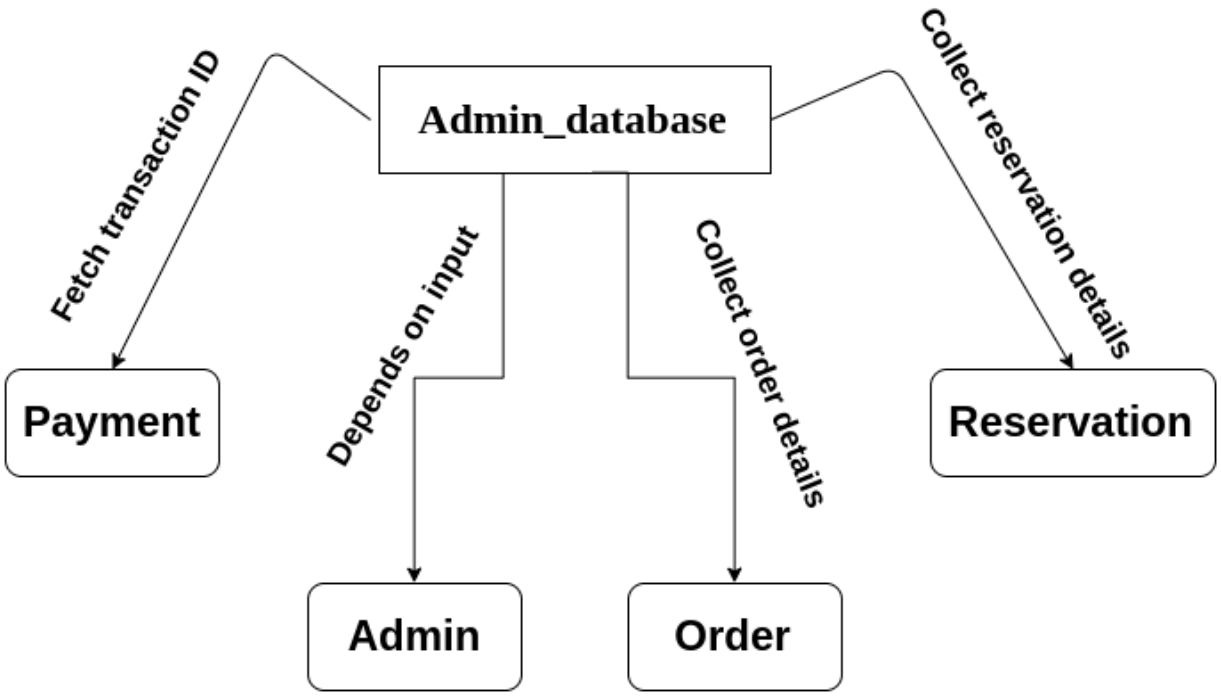


Diagram ID: 10

Name: Admin database



BEHAVIORAL MODELING OF CCMS

STATE TRANSITION DIAGRAM :

State diagram represents active states for each class the events (triggers). For this we identified all the events, their initiators and collaborators.

Event Table:

SL no.	Event	State Name	Initiator	Collaborator	Associated method

continued on next page

continued from previous page

1	Will create an account	Create_ account	Teacher, Officer	CCMS, Account, Email, SMS	+create_ account() +verify_ info() +notify_ user() +send_ confirmation()
2	Will provide information	Provide_ info	Teacher, Officer		+create_ account()
3	Users' credentials Will be verified	Verify_ info	CCMS		+verify_ info()
4	Will be able to update information	Update_ info	Teacher, Officer	Account, CCMS	+update_ info() +setFull_ name() +setMobile_ number() +setEmail_ address() +setTeacher/officer_ id() +setDepartment_ name() +setRoom_ number() +setLocation()
5	Will be able to recover password	Recover_ password	Teacher, Officer	SMS, Email, CCMS	+recover_ password() +send_ otp() +send_ link()
6	Will login to system	logIn_ to_system	Teacher, Officer, Admin		+login()

continued on next page

continued from previous page

7	Will be able to order food	Order_ food	Teacher, Officer	Order	+order() +display_ menu() +choose_ item() +select_ quantity()
8	Will provide daily menu	Provide_ menu	Admin		+fix_ menu() +update_ menu()
9	Will be able to choose food item and quantity	Choose_ item	Teacher, Officer	Admin_ database	+display_ menu() +choose_ item() +select_ quantity()
10	Will add to virtual cart	Add_to _cart	CCMS	Order	+choose_ item() +select_ quantity() +add_to _cart()
11	Bill will be shown	Show_ bill	Virtual _cart		+calculate_ amount() +prompt_ virtual_cart()
12	Can pay the bill	Pay_ bill	Teacher, Officer	Payment, SSL_ COMMERZ	+pay() +online_payment() +cash_on_ delivery() +make_ transaction() +notify_admin()
13	Order will be confirmed	Confirm_ order	CCMS	SMS, Email	+notify_ after_transaction() +send_ confirmation()

continued on next page

continued from previous page

14	Memo will be given	Give_memo	CCMS	Admin_Database, User_Database	+provide_memo() +transaction_history()
15	Will be able to cancel order	Cancel_order	Teacher, Officer		+cancel()
16	Will be refunded	Refund_money_	Reservation, Order	Cafeteria's_Transaction_Account, SSL_COMMERZ	+refund_user() +make_transaction() +notify_admin() +add_payment_to_account()
17	Can reserve multiple tables	Reserve_table	Teacher, Officer	Reservation	+reserve() +reserve_table() +customize_menu()
18	Can book the whole cafeteria	Book_cafeteria	Teacher, Officer	Reservation	+reserve() +reserve_whole_cafe()
19	Will provide customized menu	Choose_item	Teacher, Officer	Reservation	+customize_menu()
20	Payment will be calculated	Calculate_bill	Virtual_cart, Reservation		+calculate_amount()
21	Will be notified	Notify_user	Admin, CCMS	SMS, Email	+notify_after_transaction() +send_confirmation()
22	Will be able to cancel reservation	Cancel_reservation	Teacher, Officer	Reservation	+cancel() +refund()
23	Will be assigned into the booking queue	Assign_in_queue	Reservation	Teacher, Officer	+manage_queue()

continued on next page

continued from previous page

24	Will pay through "SSL COMMERZ"	Pay_ Through_ SSL	Payme nt	Teacher, Officer, SSL_ COMMERZ	+pay() +online_ payment() +make_ transaction() +notify_ admin() +add_payment_ to_account()
25	Pay onsite	Pay_ onsite	Admin	Admin_ database	+transaction_ history()
26	Can avail Cash-on delivery	Cash_ On_ delivery	Teacher, Officer	Admin	+cash_on _delivery()
27	Memo will be prompted	Prompt_ memo	CCMS		+prompt_ memo()
28	Memo will be printed	Print_ memo	Admin	CCMS	+provide_ memo()
29	Can request for room delivery	Room_ delivery	Teacher, Officer	Delivery	+request_ delivery()
30	Estimated time will be prompted	Estimate_ time	GPS		+call_GPS() +display_ distance_time()
31	Can choose delivery method	Delivery_ method	Delivery	GPS, Admin	+staff_ delivery() +pathao_ delivery()
32	Location will be tracked	Track_ location	GPS	Delivery, Virtual_ cart	+call_GPS()
33	Delivery through staff	Delivery_ Through_ staff	Delivery	Admin	+staff_delivery()
34	Delivery through "Pathao"	Delivery_ Through_ pathao	Delivery	Admin	+pathao_delivery()
35	All info will be stored	Store_ all_info	CCMS	Admin_ database, User_ database	+update_user _database() +update_ admin_database()

continued on next page

continued from previous page

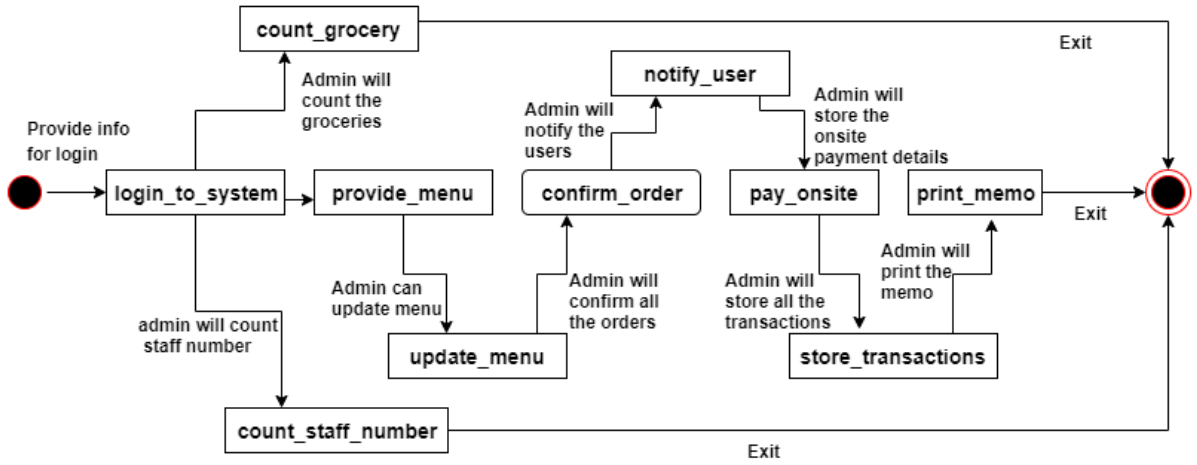
36	Staff number will be counted	Count_ Staff_ number	Admin	Admin_ database	+input_staff _count() +update_staff _count() +store_ staffInfo() +update_ staff()
37	Sum of transactions will be calculated	Calculate_ transactions	Cafeteria's Transaction Account	Admin_ database	+update_ dailyCount() +update_ weeklyCount() +update_ monthlyCount()
38	Store order memo and transaction history	Store_ transactions	Order, Reservation	Admin_ Database, User_ database	+store_reservation _detail() +store_order _detail()
39	replica of user database will be made	Make_ replica	CCMS	Admin_ database, User_ database	+update_ user_database() +update_ admin_database()
40	menu will be updated	Update_ menu	Admin		+update_menu
41	availability of food item will be displayed	Display_ Available _food	CCMS		+display_stock()
42	availability of grocery items will be counted	Count_ grocery	Admin	Admin_ database	+input_grocery _item() +update_grocery _item()
43	Prompt mostly consumed item	Mostly_ Consumed _item	CCMS	Order	+display_most _consumed_food()
44	Display all relevant food according to user's taste	Display_ Relevant_ food	CCMS	Order	+display_ relevant_food()
45	System will analyze previous months order	Analyze_ order	CCMS	Order	+analyze_ order()

State Transition

ID : 1

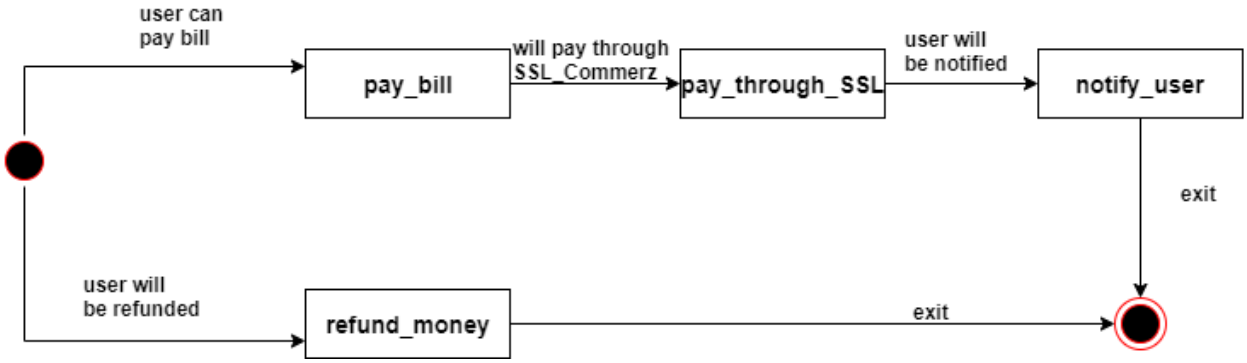
Name : Admin

ADMIN:



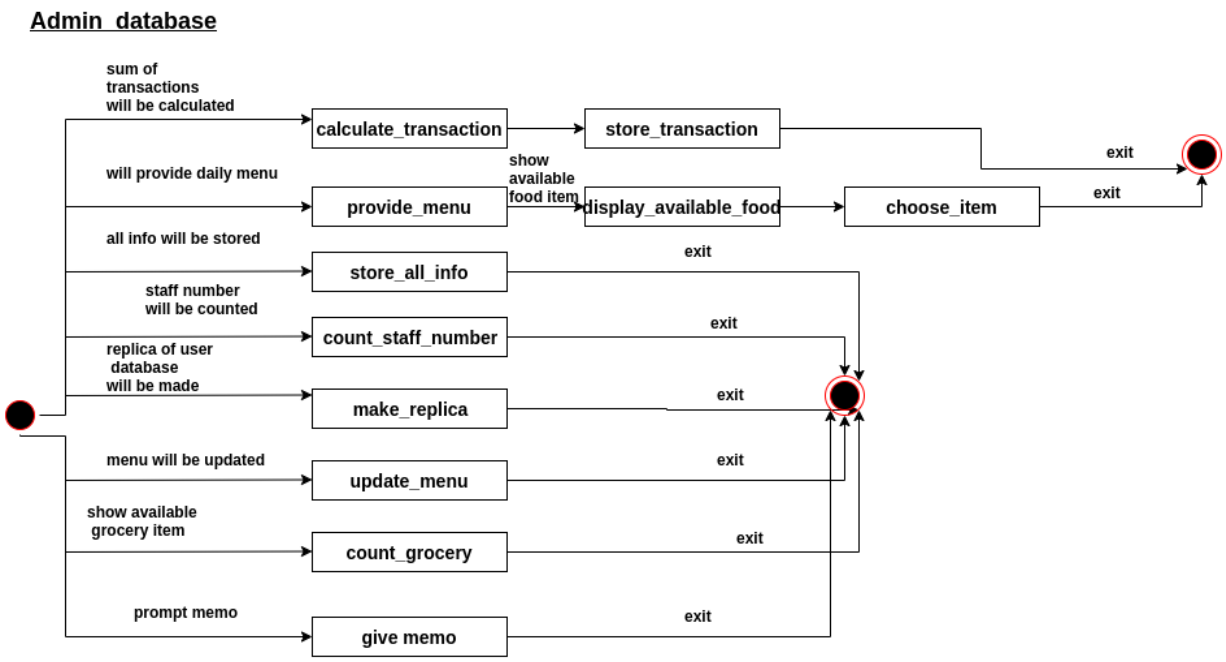
ID : 2

Name : SSL COMMERZ



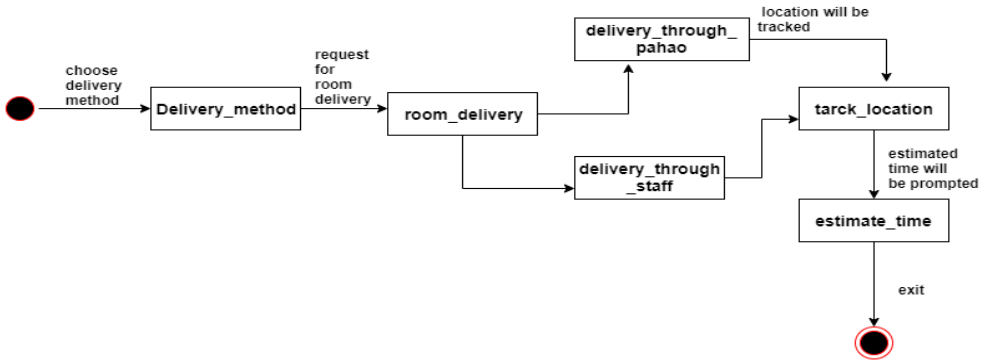
ID : 3

Name : Admin_database



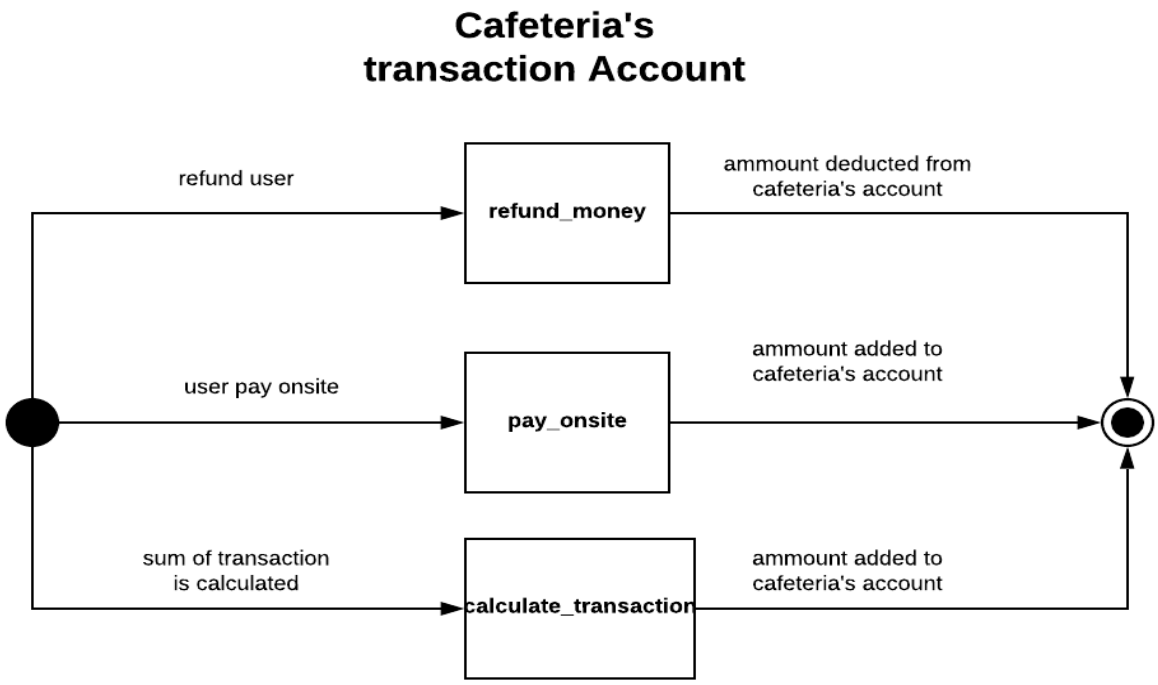
ID : 4

Name : Delivery



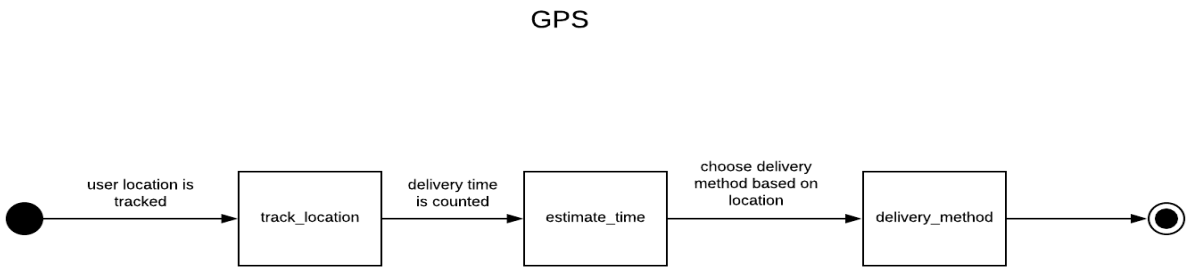
ID : 5

Name : Cafeteria's transaction account



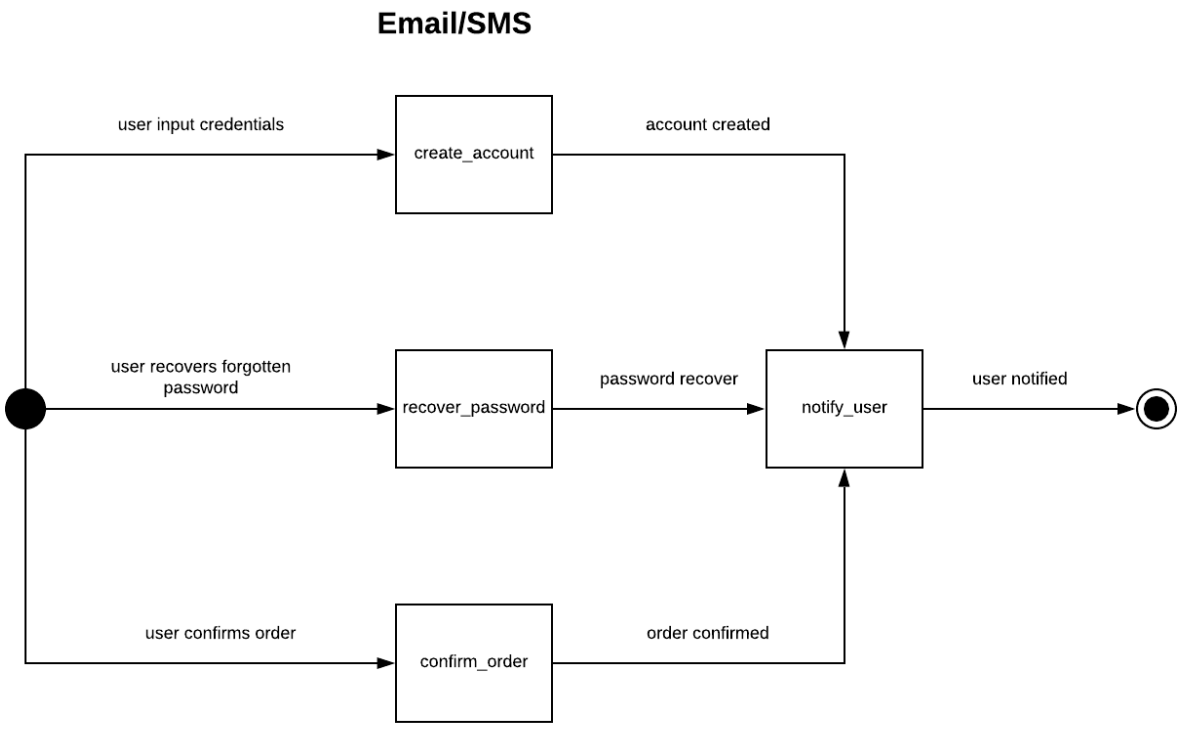
ID : 6

Name : GPS



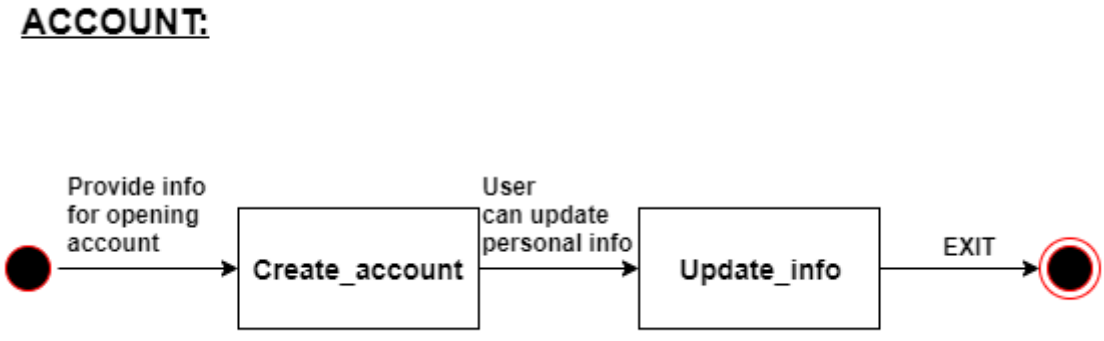
ID : 7

Name : Email/SMS



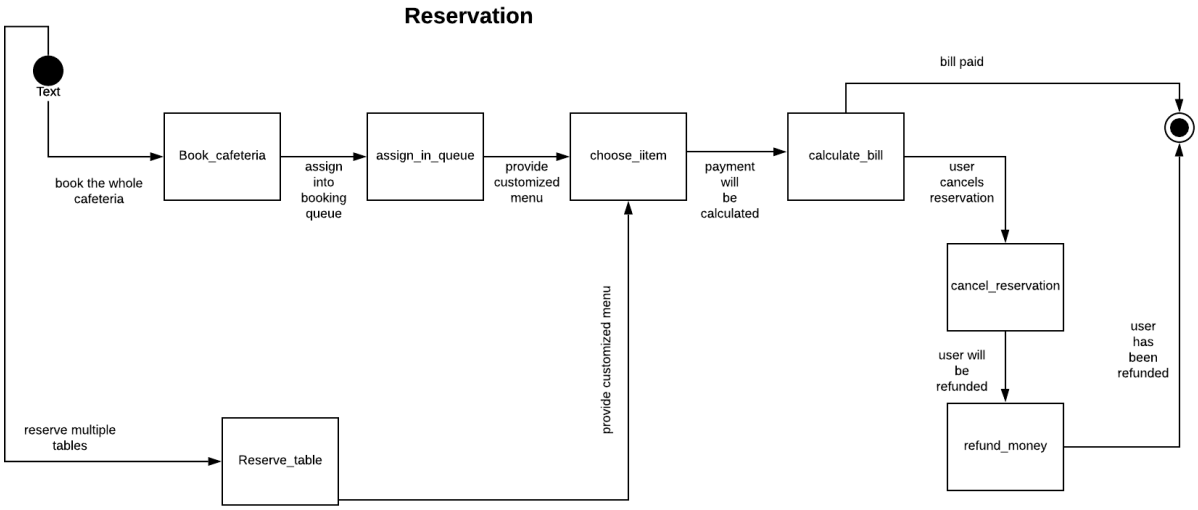
ID : 8

Name : Account



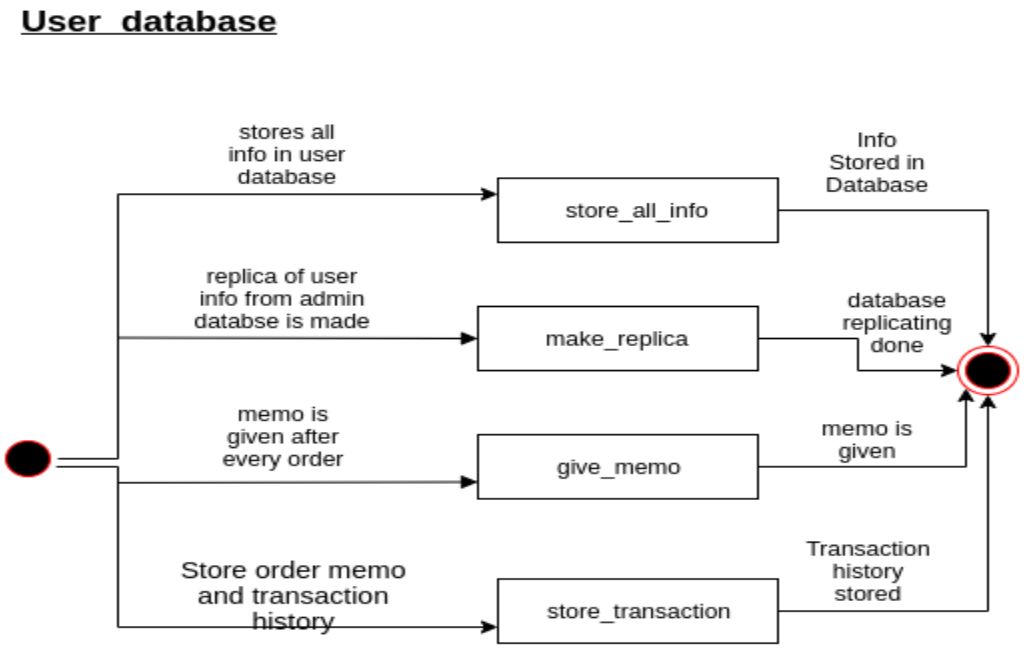
ID : 9

Name : Reservation



ID : 10

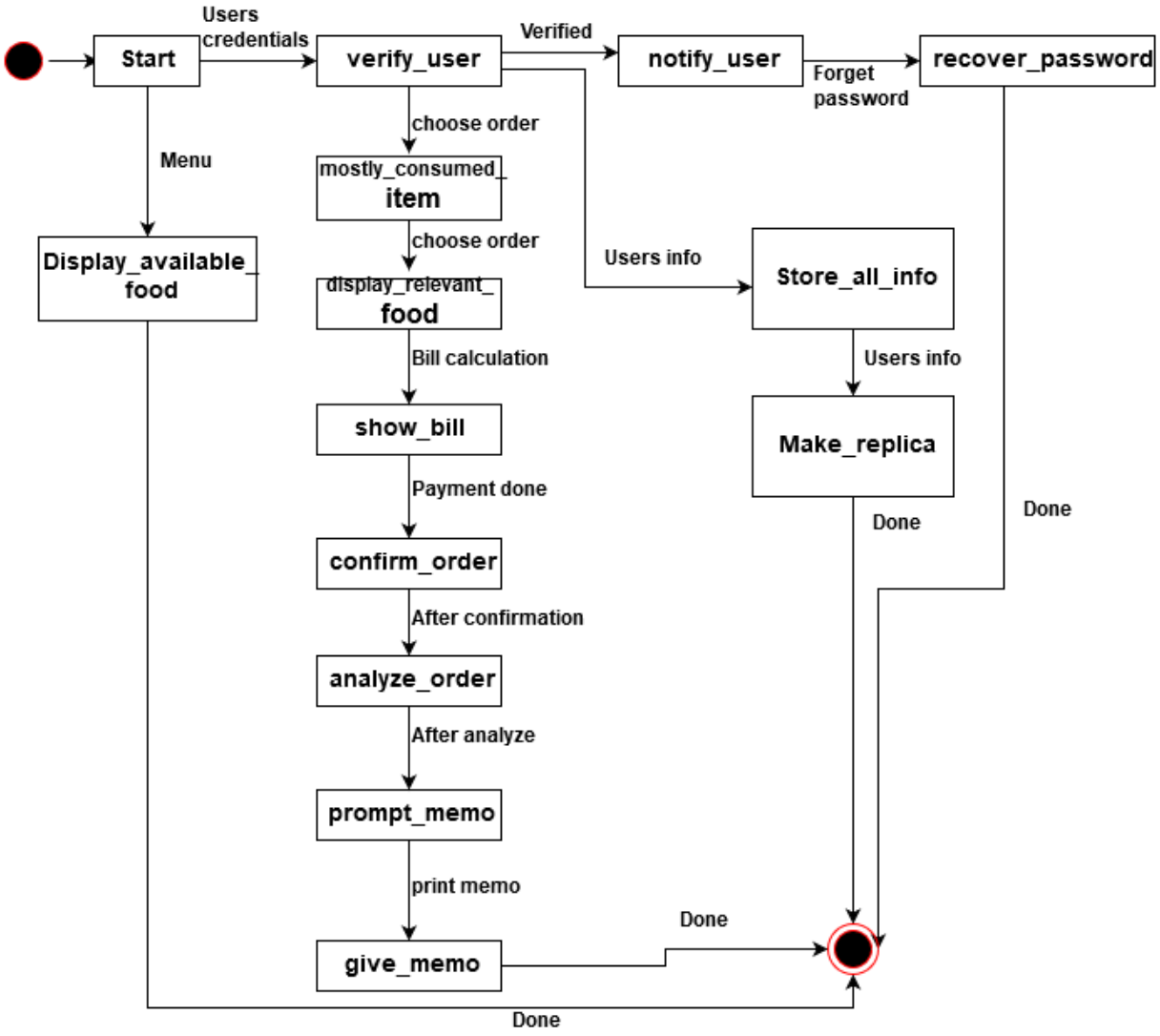
Name : User_database



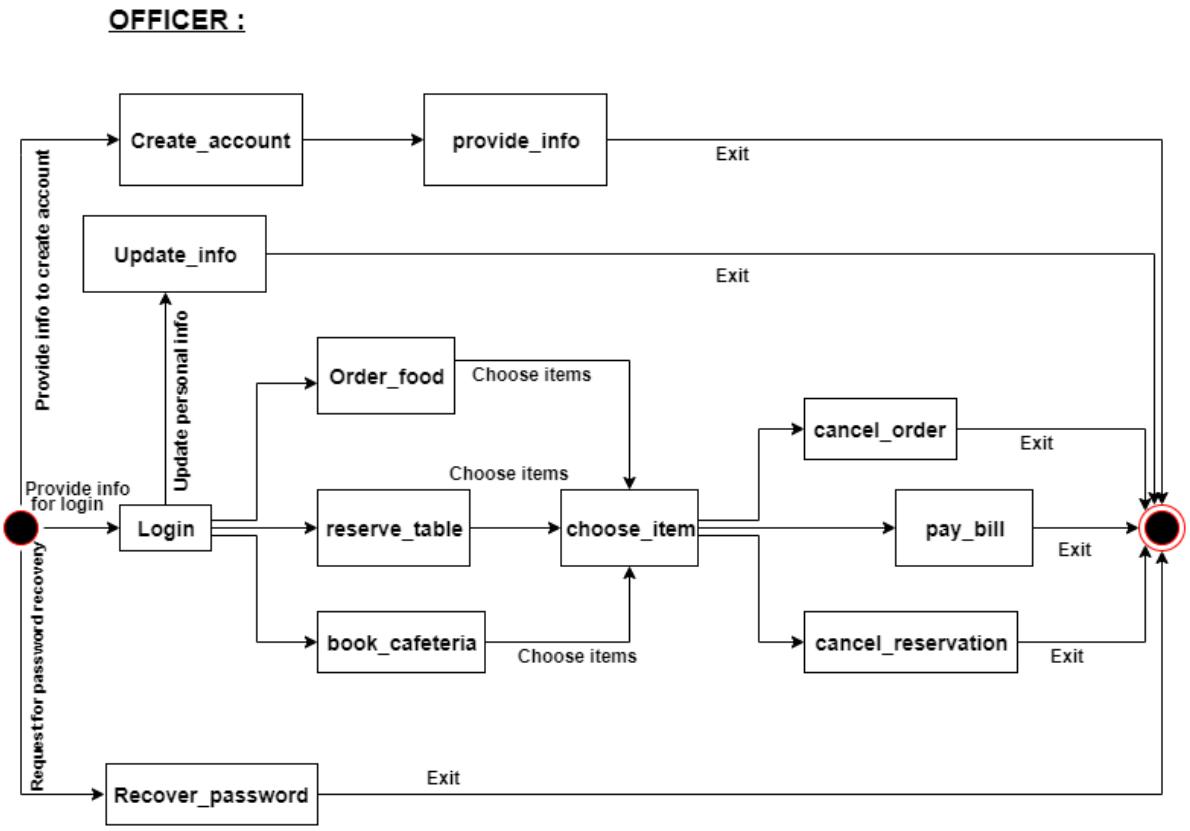
ID : 11

Name : CCMS

CCMS:

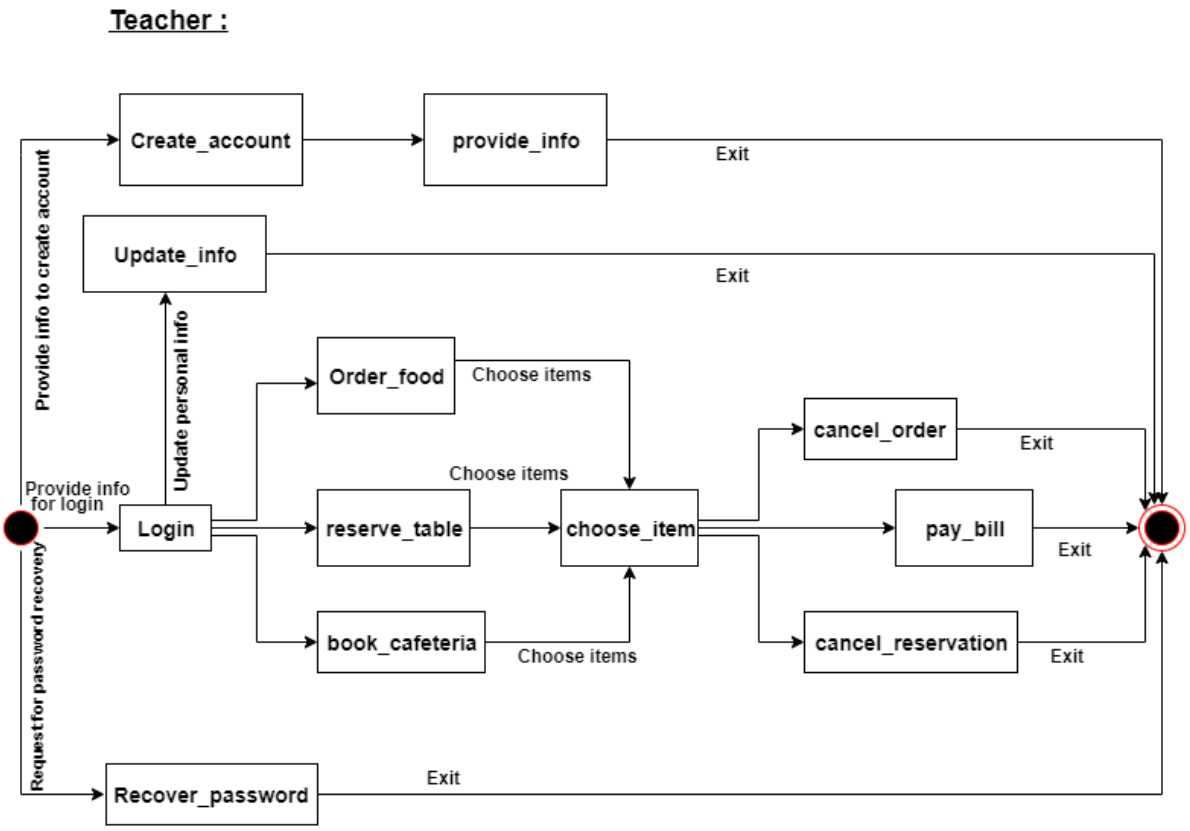

Start
Users credentials
verify_user
Verified
notify_user
Forget password
recover_password
Menu
choose order
mostly_consumed_ item
choose order
display_relevant_ food
Bill calculation
show_bill
Payment done
confirm_order
After confirmation
analyze_order
After analyze
prompt_memo
print memo
give_memo
Done
Display_available_ food
Done
Users info
Store_all_info
Users info
Make_replica
Done
Done

ID : 12



Name : Officer

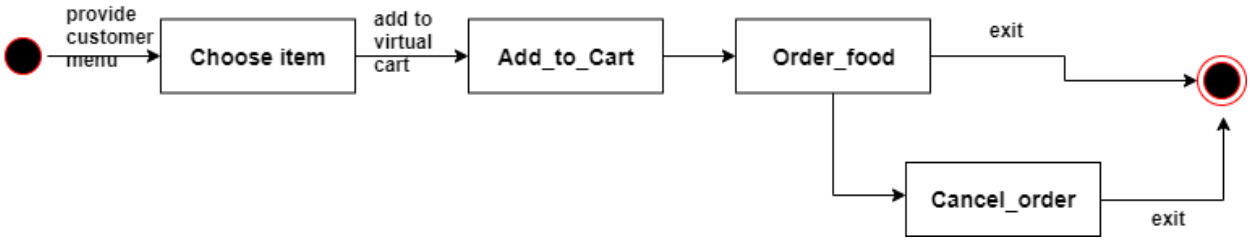
ID : 13



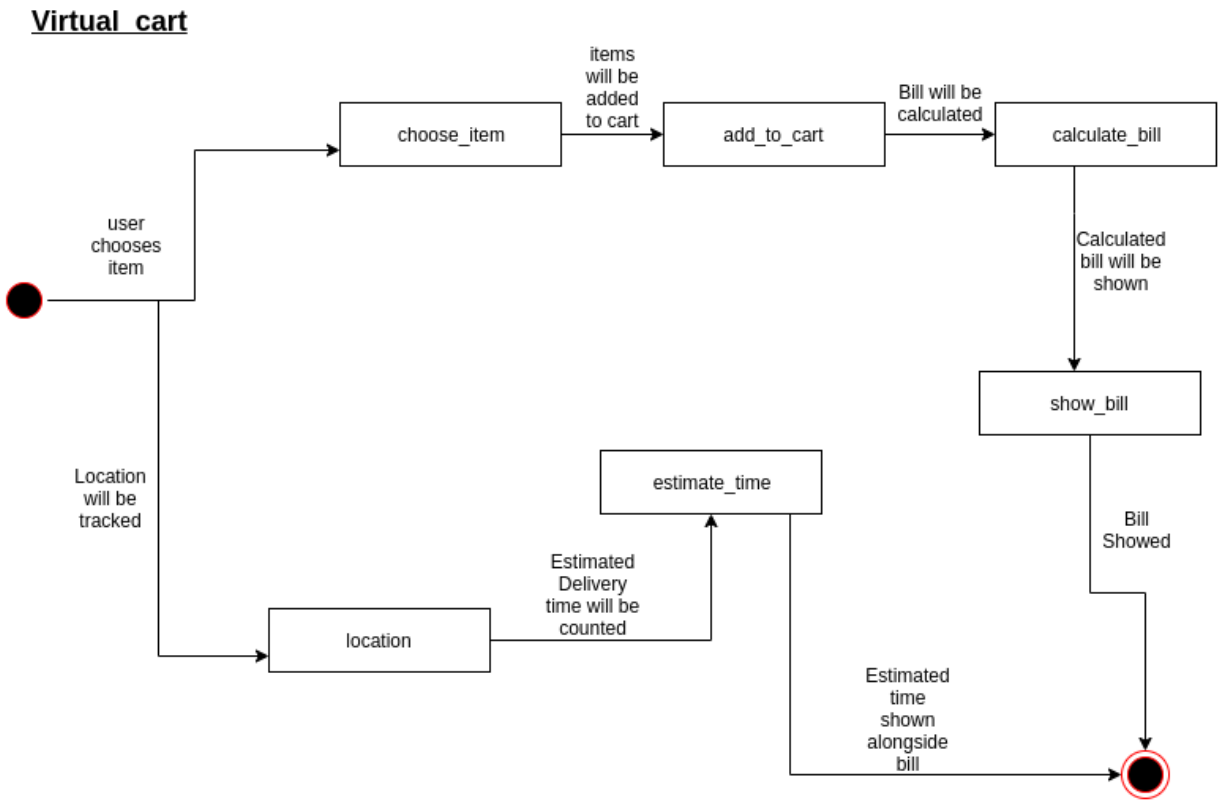
Name : Teacher

ID : 14

Name : Order



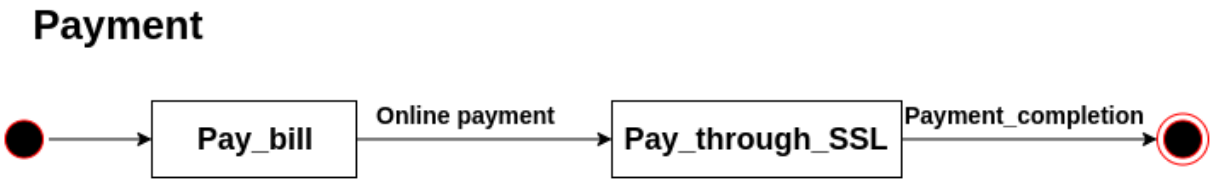
ID : 15



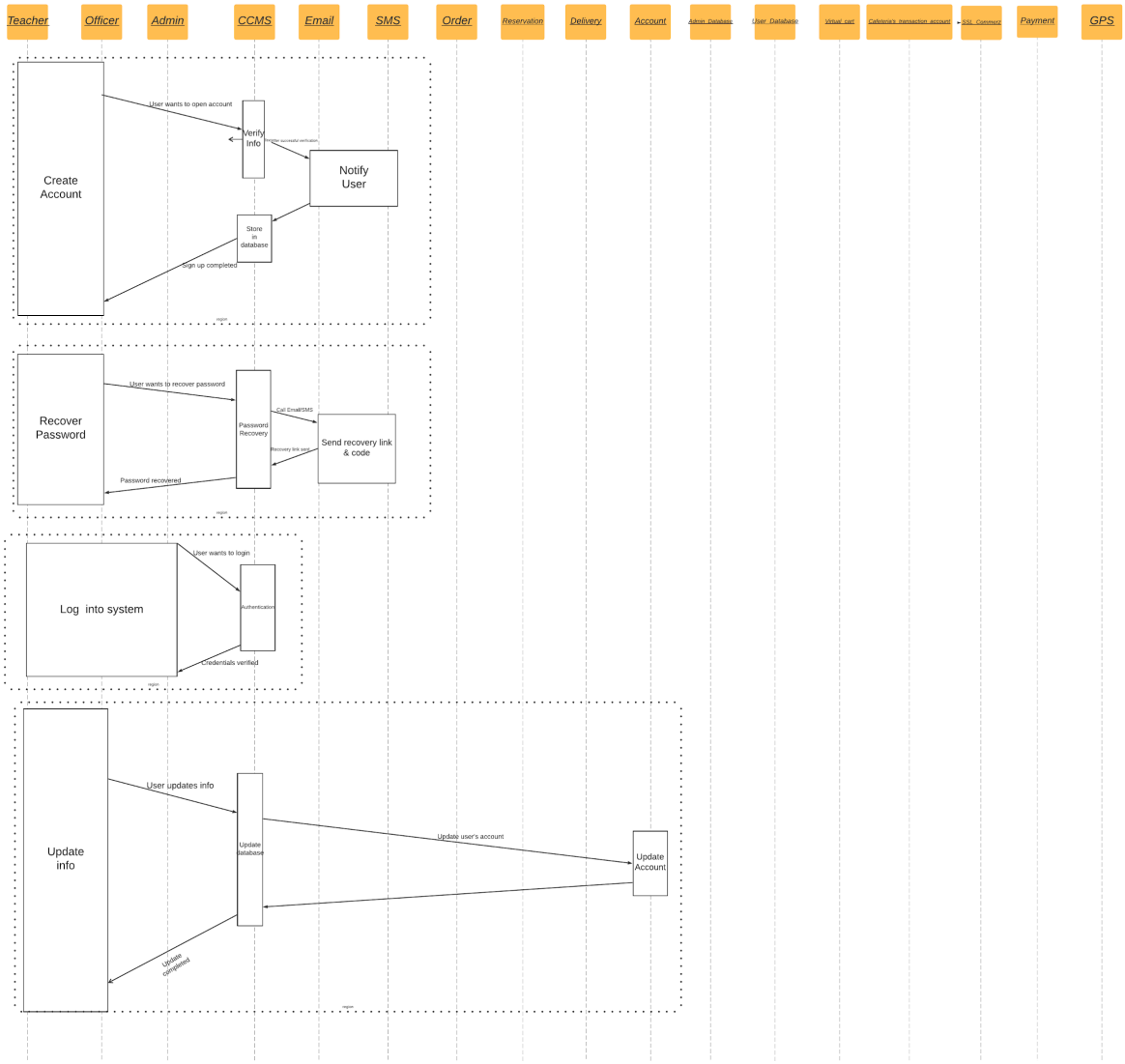
Name : Virtual cart

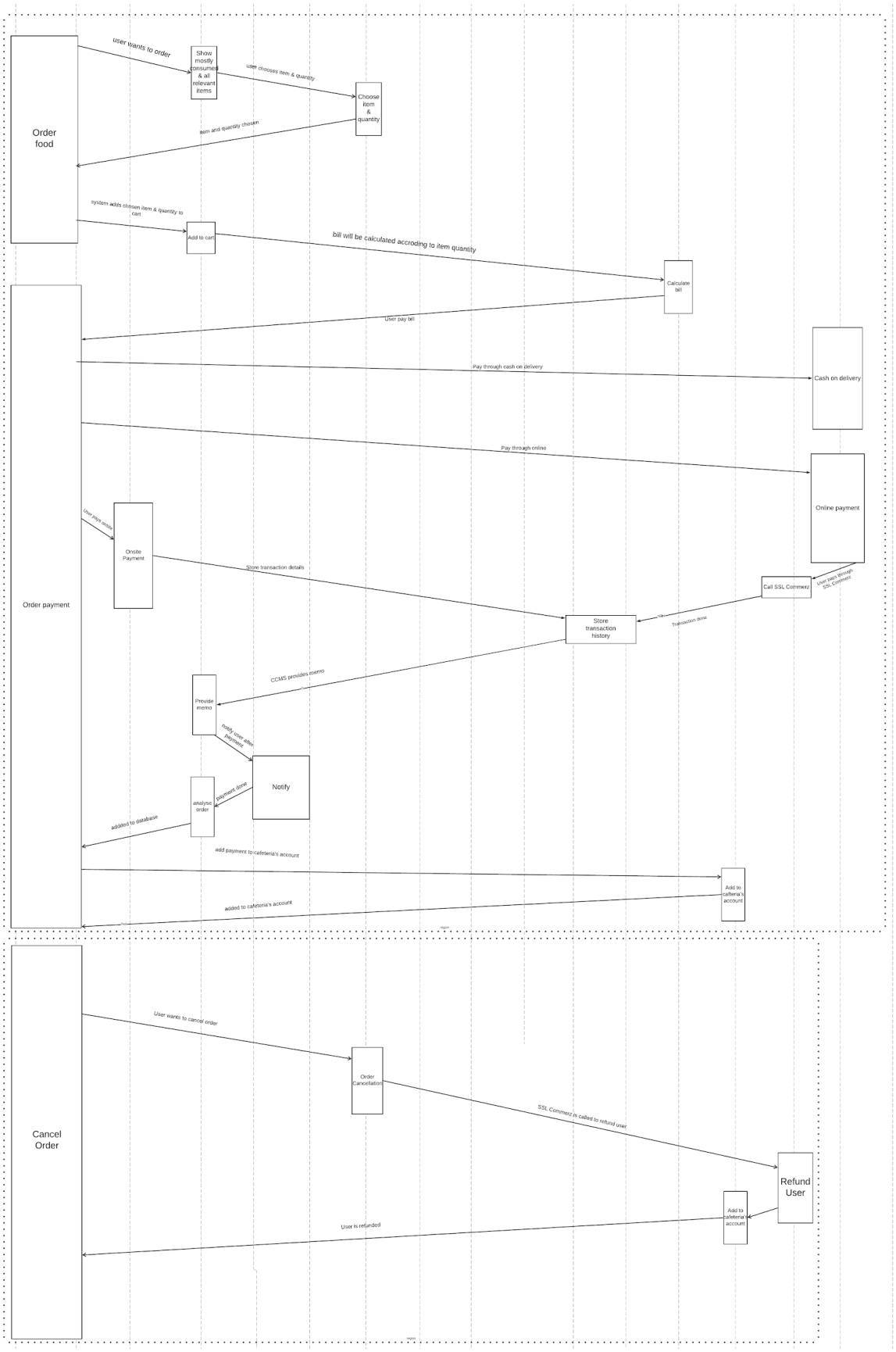
ID : 16

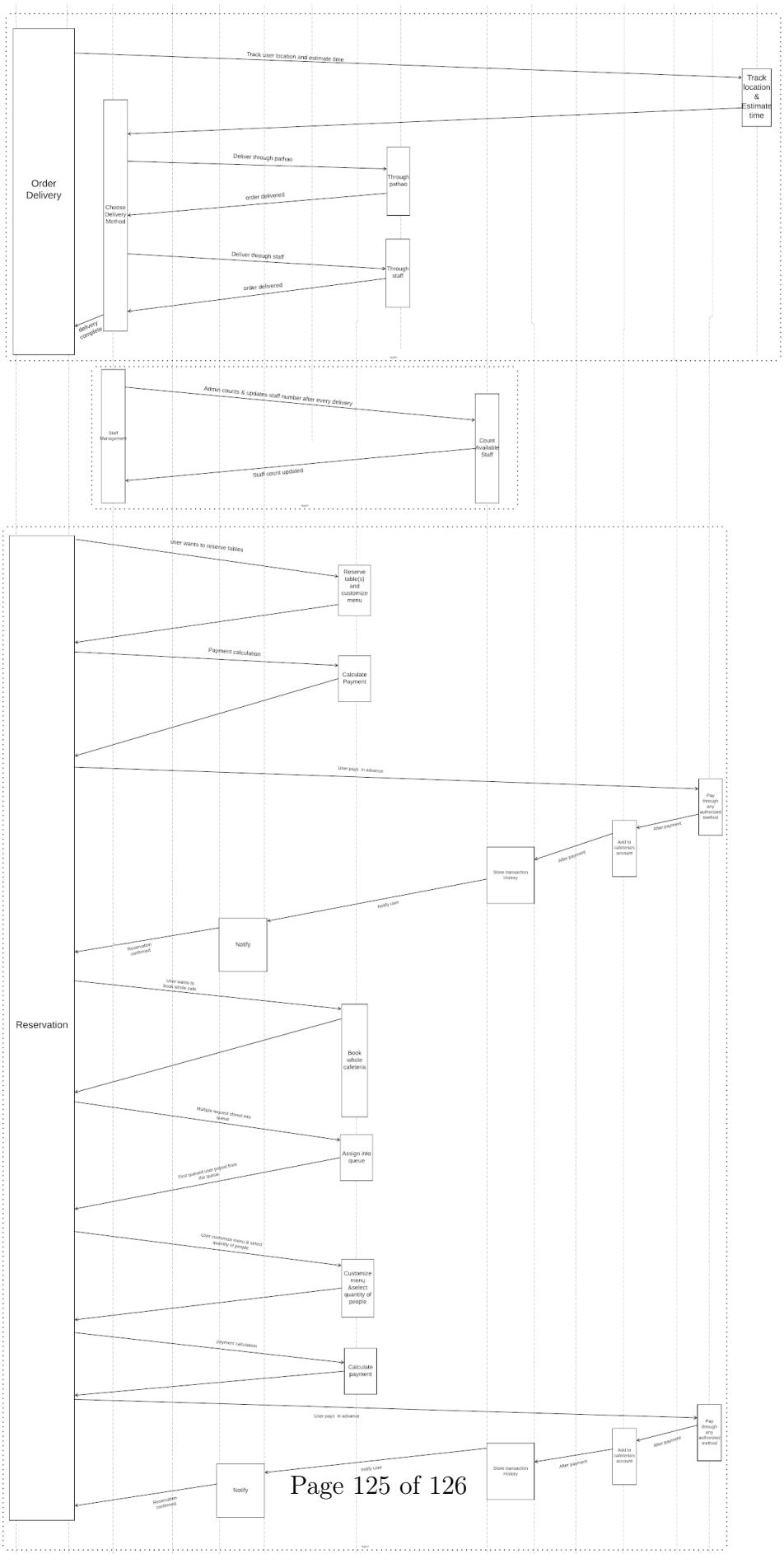
Name : Payment

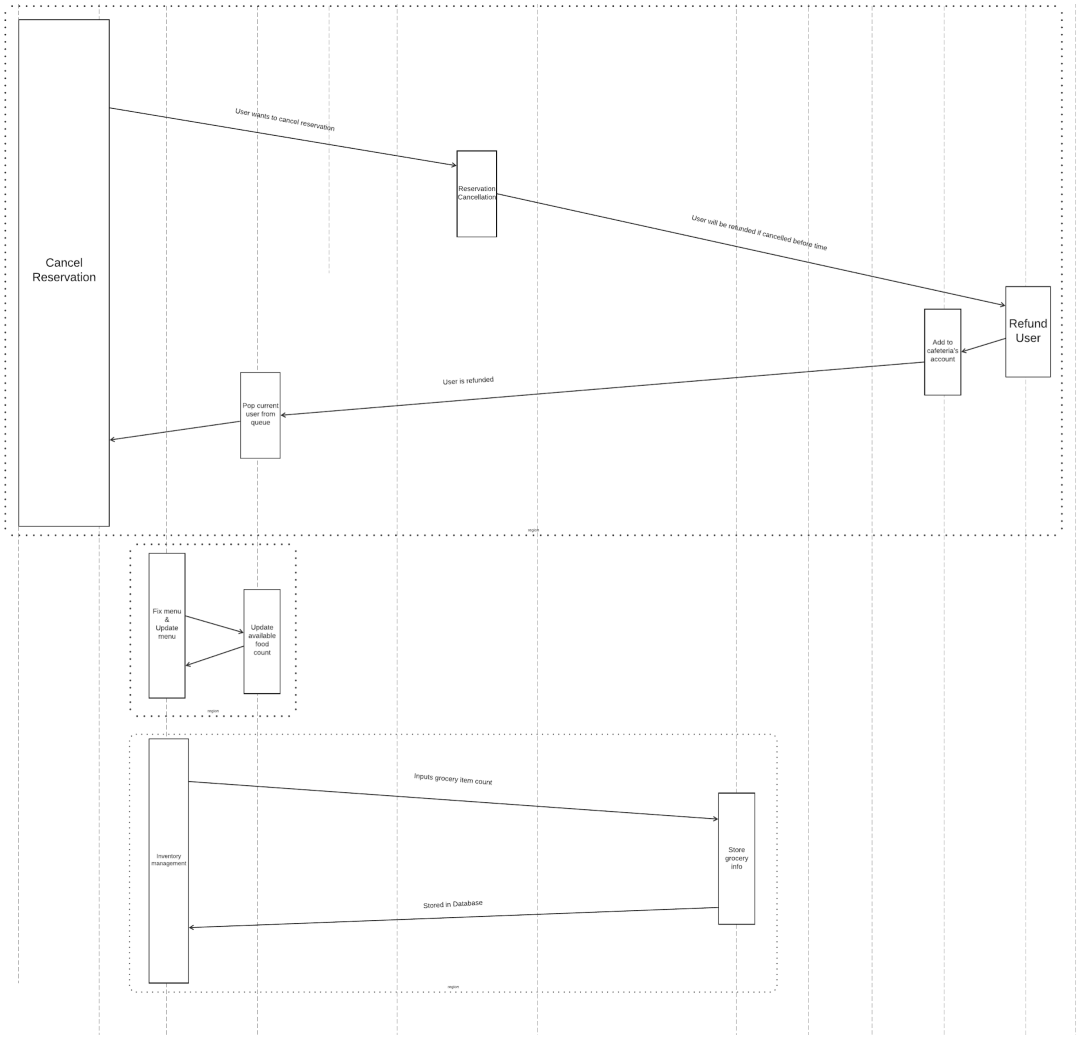


Sequence Diagram:




Order food
user wants to order
Show mostly consumed & all relevant items
user chooses item & quantity
Choose item & quantity
item and quantity chosen
system adds chosen item & quantity to cart
Add to cart
bill will be calculated accroding to item quantity
Calculate bill
User pay bill
Order payment
Pay through cash on delivery
Cash on delivery
Pay through online
Online payment
User pays online
Onsite Payment
Store transaction details
Call SSL Commerz
User pays through SSL Commerz
Store transaction history
Transaction done
CCMS provides memo
Provide memo
notify user about payment
Notify
payment done
analyse order
added to database
add payment to cafeteria's account
Add to cafteria's account
added to cafeteria's account
Cancel Order
User wants to cancel order
Order Cancellation
SSL Commerz is called to refund user
Refund User
Add to cafeteria account
User is refunded


Order Delivery
Track user location and estimate time
Track location & Estimate time
Choose Delivery Method
Deliver through pathao
Through pathao
order delivered
Deliver through staff
Through staff
order delivered
Delivery complete
Staff Management
Admin counts & updates staff number after every delivery
Count Available Staff
Staff count updated
Reservation
User wants to reserve tables
Reserve table(s) and customize menu
Payment calculation
Calculate Payment
User pays in advance
Pay through any authorized method
After payment
Add to cafeteria's account
After payment
Store transaction History
Notify user
Notify
Reservation confirmed
User wants to book whole cafe
Book whole cafeteria
Multiple request stored into queue
Assign into queue
First optimized user pulled from this queue.
User customize menu & select quantity of people
Customize menu &select quantity of people
Payment calculation
Calculate payment
User pays in advance
Pay through any authorized method
After payment
Add to cafeteria's account
After payment
Store transaction History
Notify user
Notify
Reservation confirmed


Cancel Reservation
User wants to cancel reservation
Reservation Cancellation
User will be refunded if cancelled before time
Refund User
Add to cafeteria's account
User is refunded
Pop current user from queue
Fix menu & Update menu
Update available food count
Inventory management
Inputs grocery item count
Store grocery info
Stored in Database