CAFETERIA MANAGEMENT SYSTEM

Software Requirement Specification

Group 5

BSSE 10th batch

Institute of Information Technology.

University of Dhaka

Cafeteria Management System

Submitted to

Dr. Kazi Muheymin – Us – Sakib

Professor

Institute of Information Technology

University of Dhaka

Submitted by

Group 5

Sadman Sakib	BSSE 1005
Rabiul Islam Sujon	BSSE 1012
Lamisa Quaiyum Shamma	BSSE 1018
Mehedi Hasan Sun	BSSE 1025
Nazmul Hossain	BSSE 1032
Sudip Kumar Sah	BSSE 1040

4th Semester, BSSE 10th Batch

Institute of Information Technology

University of Dhaka

Submission Date: 11 November 2019

LETTER OF TRANSMITTAL

11th November, 2019

Dr. Kazi Muheymin-Us-Sakib

Professor

Institute of Information Technology

University of Dhaka

Subject: Submission of term report on “Cafeteria Management System”.

Sir,

With due respect, we are submitting the report on the above topic you assigned to us. In this report, we have given our best effort albeit some shortcomings. We earnestly hope that you would excuse our errors and oblige thereby.

Yours sincerely

Sadman Sakib	(BSSE 1005)
Rabiul Islam Sujon	(BSSE 1012)
Lamisa Quaiyum Shamma	(BSSE 1018)
Mehedi Hasan Sun	(BSSE 1025)
Nazmul Hossain	(BSSE 1032)
Sudip Kumar Sah	(BSSE 1040)

2nd Year, 4th Semester, 10th Batch

Institute of Information Technology

University of Dhaka

Session: 2017-18

ACKNOWLEDGEMENT

We are highly indebted for getting such a tremendous opportunity to prepare the report on Cafeteria Management System. We would like to thank whole-heartedly our course teacher, Dr. Kazi Muheymin-Us-Sakib, Professor, Institute of Information Technology, University of Dhaka, for giving us guidelines about how we can prepare this report. In completing this paper, we have collected various important data and information from CARS Cafeteria. We are thankful to all of the works cited.

ABSTRACT

The study is made for Cafeteria Management System. The scope of the study is to analyze the Cafeteria Management System and to know its functions and drawbacks, and design the SRS of this system. The object of this study is to develop an SRS (software requirements and specifications) of Cafeteria Management System. This study also describes the current system of the CARS Cafeteria.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This chapter is a part of our software requirement specification for the project “Cafeteria Management System”. In this chapter we focus on the intended audience for this project.

1.1 Purpose

This document briefly describes the Software Requirement Analysis of Cafeteria Management System. It contains functional, non-functional and supporting requirements and establishes a requirements baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered and organized by topic. 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.

1.2 Intended audience

This SRS is intended for several audiences including the customers as well as the project managers, designers, developers, and testers.

The customer 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.

1.3 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.

CHAPTER 2

INCEPTION OF CMS

2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how a software project gets started and what the scope and nature of the problem to be solved is. The goal of the inception phase is to identify concurrent needs and conflicting requirements among the stakeholders of a software project. At project inception, we established a basic understanding of the effectiveness of preliminary communication and collaboration between the other stakeholders and the software team.

To establish the groundwork, the following factors have been worked on to the inception phase:

- Icebreaking
- List of stakeholders.
- Recognizing multiple viewpoints.
- Working towards collaboration.
- Requirements questionnaire.

2.2.1 Icebreaking

Icebreaking refers to the fact that to diminish the communication barrier between you and the other person. It is a crucial part since it denotes the acceptance of our proposal. We started this face by talking with them with context free languages. Their behavior, respond to our question or willing to take a change in their shops solely depends on this phase.

2.1.2 List of Stakeholders

Stakeholders refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that might be affected by its installation. At inception, a list of people who will contribute input as requirements are elicited. The initial list will grow as stakeholders

are contacted because every stakeholder will be asked: “Whom else do you think I should talk to?”

To identify the stakeholders, we consulted with Cafeteria Manager and asked him the following questions:

- Who is paying for the project?
- Who will be using the project outcomes?
- Who gets to make decisions about the project (if this is different from the money source)?
- Who has the resources I need to get the project done?
- Whose work will my project affect? (During the project and also once the project is completed).

We identified the following stakeholders for our automated cafeteria management system.

- Teacher.
- Officer.
- Manager.
- Staff.

2.1.3 Recognizing Multiple Viewpoints

Different stakeholders demand different features from the software. To satisfy the stakeholders, most of these features should be included in the software.

Teacher and officer’s viewpoint:

- Easy to use
- Different payment method
- Online payment.

Manager’s viewpoint:

- Storing staff information
- Storing stock information
- Calculate profit information

2.1.4 Working towards Collaboration

Every stakeholder has their own requirements. There are some common and conflicting requirements of our stakeholders. That's why we followed the following steps to merge these requirements-

- Find the common and conflicting requirements.
- Categorize them.
- List the requirements based on stakeholder's priority points.
- Make final decision about requirements.

Common viewpoints:

- Error free effective system
- User friendly
- Easy to maintain the software
- Strong authentication system.

Conflicting viewpoints:

- Keeping information regarding day to day transaction.
- Developing the project in minimum budget

Final requirements:

We finalize the following requirements based on stakeholder's priority point:

- User friendly system.
- Strong authentication.
- Maximum error free system. (5%-10% error is considerable).
- Restrict access to functionality of the system based upon user roles.
- Easy to maintain the software.

Final requirements:

We finalize the following requirements based on stakeholder's priority point:

- User friendly system.
- Strong authentication.

- Maximum error free system. (5%-10% error is considerable).
- Restrict access to functionality of the system based upon user roles.
- Easy to maintain the software.

2.1.5 Requirements Questionnaire

At first some context free questions were asked for identifying the stakeholders. Context free questions are helpful to identifying some stakeholders who cannot be identified by structural questions. Then questions regarding the software were regarding to know their demands. The questionnaires are included in the appendix section.

2.2 Conclusion

The Inception phase helped us to establish a basic understanding about the Cafeteria management system of University of Dhaka, identify the stakeholders who will be benefited if this system becomes automated, define the nature of the system and the tasks done by the system, and establish a preliminary communication with our stakeholders.

In our project, we have established a basic understanding of the problem, the nature of the solution that is desired and the effectiveness of preliminary communication and collaboration between the stakeholders and the software team. More studies and communication will help both sides (developer and client) to understand the future prospect of the project. Our team believes that the full functioning document will help us to define that future prospect.

CHAPTER 3

ELICITATION OF CMS

This chapter specifies the Elicitation phase.

3.1 Introduction

Requirements Elicitation is a part of requirements engineering that is the practice of gathering requirements from the users, customers and other stakeholders. Many difficulties were faced, like understanding the problems, making questions for the stakeholders, limited communication with stakeholders due to a short amount of time and volatility. Though it is not easy to gather requirements within a very short time, these problems have been surpassed in an organized and systematic manner.

3.2 Eliciting Requirements

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. The following tasks were done for eliciting requirements-

- Collaborative Requirements Gathering
- Quality Function Deployment
- Usage Scenarios
- Elicitation work products

3.2.1 Collaborative Requirements Gathering

We have met with many stakeholders in the Inception phase such as the teacher, officer, manager, staff. 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 rule in the whole process) again to elicit the requirements. A slightly different scenario from these approaches has been found.

Following activities have been completed to accomplish this task.

The meetings were conducted with the teacher, officer, manager, staff; they were questioned about their requirements and expectations from the Cafeteria Management System.

They were asked about the problems they were facing with the current manual system.

Lastly, we selected our final requirement list from the meetings.

3.2.2 Quality Function Deployment

Quality Function Deployment (QFD) is a quality management technique that translate the need of the clients into technical requirements for the software. The prime concern of the QFD is customer satisfaction maximization. In order to ensure this, QFD enforces in understanding of what customers describe as ‘valuable’ and then deploy these values throughout the engineering process.

QFD defines three types of requirements:

- Normal Requirements
- Expected Requirements
- Exciting Requirements

3.2.2.1 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.

- Allow new members to register.
- Login system.
- View menu.
- Order food.
- Table reservation.
- Food delivery system.

- Multiple payment methods.
- Event booking.
- Store day to day transaction.
- Notify members via SMS and email.
- Ease of calculation.
- Feedback option.

3.2.2.2 EXPECTED REQUIREMENTS

The requirements that are implicit to the system might not be brought up during the meeting because of their fundamental nature. Despite not being explicitly mentioned, their presence must be ensured. Otherwise, the product will leave customers dissatisfied. These requirements are called expected requirements and these are stated below:

- Database.
- Login type (User/Manager/Staff)
- Error free software
- User friendly
- Data backup.
- Interactive and attractive graphical user interface.
- Strong authentication process.

3.2.2.3 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.

- User can add friends to friend list who use the same system.
- User can suggest food items from the menu to any of his friends using the system.
- System generates food sale and profit loss graph.
- System provides automatic signal when any product or item is short.

3.2.3 USAGE SCENARIO

Cafeteria Management System is an automated system that will allow cafeteria authority to create online assist to teachers and officers of Dhaka University. Cafeteria authority can manage food delivery, order processing and event management through it. We expect our system to have the following features: -

Account management:

Users of this system are teachers and officers of Dhaka University. In order to enter into this system, a user has to create an account first. There are three types of accounts. After a successful account creation, the user can log in to the system by providing their username and password.

Account type:

1. User Account:

This type of account is for teachers and officers of the University of Dhaka.

2. Manager Account:

This type of account is for cafeteria manager.

3. Staff Account:

This type of account is for the staff of the cafeteria. Among them, only cashiers of the cafeteria are allowed to interact with the system.

Account creation:

Users have to provide their full name, username, password, email address, mobile number, address of their respective department and room number to create their account.

Manager account will be created only by super admin before the launch of this system. Later on, more manager account can only be edited and created by the manager. To create the manager's account username, password, email address is needed.

Only manager can create staffs' account. To create a staff's account- his name, username, post, mobile number and email address is needed. All posts (cook, waiter, delivery-man, cashier) of the staff will be shown to the manager. If the manager wants to create a Cashier's account, the cashier has to provide his email address. A link will be sent to that email in which cashier can set a password for his account.

The password will have at least 8 characters which is a combination of letters and numbers. While creating an account if the username, email address or mobile number of a user already exists, the system will give a warning. After providing the information, a verification code will be sent to the email address and phone number. The system will ask for that code. After successfully creating an account, a notification will be sent to the user's email and mobile number.

Authentication:

While creating an account, the teacher has to provide the email address that was stored in Dhaka University database. Then the system will forward that email address to Dhaka university authority and the authority will check the email address in the database.

If the provided email doesn't exist in Dhaka University database, the user has to be referenced by any of the existing teacher or officer. Teachers and officers can search the existing users and send an access request to them. When an existing user gives confirmation, the authentication process will be finished.

Login:

To login into the system, one has to provide his username and password. If the correct username-password pair is not provided, an error dialogue box will be shown. After 5 unsuccessful trials, an email will be sent to the user and he cannot get access to the system for 2 days. He can reset his password or username if he forgets those. For resetting the password, a link will be sent to the user's email where he will be asked to enter his new password.

User can search any existing user and send him/her friend request. If he/she is familiar with the user ,he/she can respond to the request.

Stock management:

User Perspective:

After logging in, user can see the menu. User can check the item lists and the price of those items. Only available foods will be shown to the user.

There will be two types of food items.

1. General food items:

   It refers to foods those are available every day on cafeteria. There are six food items.

   - Chicken with rice, daal and vegetables
   - Rohit fish with rice, daal and vegetables
   - Beef with rice, daal and vegetables
   - Omlet with rice, daal and vegetables
   - Ruti with beef
   - Ruti with daal and vegetables

2. Special food items:

   The authority provides these items on special occasions or events.

   - Sunday: Khicuri with chicken
   - Monday: Khicuri with beef l i
   - Tuesday: Khicuri with chicken
   - Wednesday: Chinese food
   - Thursday: Polao with beef and salad.

User can choose which food item he wants and also can set the quantity of the food in the cart.He can also remove any item and reduce the quantity of the item from his cart . After that he can order those food items. There will be a 'suggest food' option in case any user decides to suggest any food item to his friends. User can choose friends whom he wants to suggest food item.

Cafeteria Perspective:

Authority will keep the record of daily amount of food and cost of raw materials needed for preparing food. The food items with their respective amount will be updated in the database everyday and for every order, the amount of that food item will be reduced from database. The cashiers will do this job. Some customers may not use the system and order manually. On this case, cashiers will reduce the amount of food from the database.If the amount of product for certain food item is less than 5, a notification will be sent to the manager through email by the system.

There will be "Add More Items" option for the manager. He can add, delete and edit any items from the food lists.

Order processing:

There will be three types of food services.

Food Delivery: Once a user orders his preferred food for delivery, a cashier will print a memo and give it to the available delivery-man. The delivery-man will take the food to the given address. Both the delivery-man and the cashier's name will be stored in the database. Thirty taka will be charged for each delivery.

Table reservation: A user can reserve a table for Dine-in. For booking a table, a user has to order at least 40 minutes earlier. A user will be given an order-code through email and SMS. When the user comes to the cafeteria, he has to show the code to the cashier to get the food. If the user does not come within 30 minutes from the booking time, his booking will be cancelled.

Pre-Order: A Users can just order the food with the system without table reservation and come at his preferred time. He will be given an order-code through email and SMS. When he comes to the cafeteria, he needs to show his order-code before having the meal.

Add food preferences: User can add notes about the food. This is not mandatory for the user.

Payment:

Users can pay through three different ways.

1.**Online Payment:** A User can pay through Bkash or rocket or through card. For online payment, he will be asked to log in to his account. After login, the user will be shown the bill and a confirmation message in a dialogue box. After the confirmation of the order, the payment will be completed and a notification will be sent to the user's email and SMS.
For pay through cards users have to go to the cafeteria and swipe the card in the EDC machine.
2.**Cash Payment:** A User can choose "Cash payment" where he will pay the bill in the counter. In this case, the customer's name as well as the cashier's name will be stored in database. After receiving the money, the cashier will confirm the payment completion of the user.
3.**Pay from Salary:** Apart from the procedures mentioned above, a users can choose "Pay from salary". Every day after ordering the food, a notification of the bill will be sent to the user through system and he will be asked for confirmation. At the end of the month, cafeteria management will send his total bill to the University. University will cut the bill from the user's salary and give the cash to the cafeteria.

Event Management:
A user can book cafeteria for an event .There will be an event calendar for the manager to see and edit. Manager can see which dates are booked for an event. The user has to let cafeteria management know about the event at least 3 days ago. Cafeteria has 6 party menus to offer. User can choose any of this menu for the event. Menu and their prices (taka Per head)are listed below -
Menu 1(420 taka):

Plain Polao, Chicken Roast/Bhuna, Mutton Rezala, Mixed Vegetable, kabab-1, Firni/ Yoghurt/ Jarda,Salada, Water

Menu 2(450 taka):
Kacchi Biriani, Chicken Roast ¼, Kabab -1, Firni/ Yoghurt/ Jarda,Borhani/ Soft Drinks, Salad,Water

Menu 3(410 taka):
Plain Rice, Rui Fish Fry, Mixed Vegetable, Chicken Curry ¼, Mash – Prown , MugDal, Salad, Soft Drinks, Water

Menu 4 (465 taka):
Corn Soup, Wonthon -2 Pc, Mixed fried rice, Chinese Vegetable, Fried Chicken-2pc, Chicken Masala, soft drinks, Water

Menu 5(520 taka):
Plain Rice,Rui Fish Fry, Beef Rezala, Chicken Curry ¼,Mixed Vegetable, Parblefry, Mixed Dal, Salad, Water

Menu 6(570 taka):
Morong Polaw, Mutton Rezala, Kabab-1, Firni, Yoghurt, Salad, Water

User have to choose the event time and number of people who'll come. Event time must be after 2:00 pm. Cafeteria management can arrange events for up to 200 people. But if the

number of people is higher than 150, 10% of total cost will be added. Then User have to pay advance at least 50% of the total cost through payment methods to confirm the event. And the rest of the cost must be paid on the event date. If for any reason a user wants to cancel the event, there will be an option for the user to cancel the event. Cancellation must happen 72 hours before. Otherwise, cafeteria management will take 15% of the cost. For an event, 10% of service charge will be added.

Staff management:
Only the manager of the cafeteria can maintain the staff management part.
Joining date of the staff, attendance, posts, salary of the staff of the cafeteria will be stored in the system's database. Manager can see and modify their stuff's attendance, post and salary.

At least three of the staff must be assigned for food delivery.

Feedback and special item:

There'll be a rating and comment section for all of the food items. User can choose if he wants the update of the special food items and will get the update through this system.

Graphical Representation of Food Sale:

Daily sales graph will show hourly sales for each food items and graph of time ranged sales will show weekly or monthly total sales. Manager can choose which type of graph he wants to see.

Graphical Representation of Profit-loss:
The system will also show a graph of profit -loss to the manager. For this graph, the following formula will be used-

Profit = Earned money – expenses

The earned money and expenses will be calculated from the database. All money transactions will be recorded in the database. From the list of sold food items and event booking, earned money can be calculated. The cashier will upload expenses for each food items in the database and thus the system will calculate the expenses. The manager will verify daily expenses from the database.

3.2.4 Elicitation Work Product

At first, it has to be known whether the output of the Elicitation task may vary because of the dependency on the size of the system or the product to be built. Here, the Elicitation work product includes: -

- Making a statement of our requirements for the Dhaka University Club Management System
- Making a bounded statement of scope for our system.
- Making a list of customers, users and other stakeholders who participated in the requirements elicitation.
- Making a list of requirements that are organized by function and domain constraints that apply to each other.
- A set of usage scenarios that provide insight into the use of the system.
- Description of the system's technical environment.

Chapter 4

Scenario Based Modeling

This chapter describes the Scenario Based Model for the Cafeteria Management System.

4.1 Introduction

When developing software, user satisfaction is given the highest priority. If we understand how end users (and other actors) want to interact with a system, our software team will be better able to properly characterize requirements and build meaningful analysis and design models. Thus, requirements being with scenario generation in the form of use cases, activity diagrams and swim lane diagrams.

4.2 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.

4.3 Use case diagrams

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

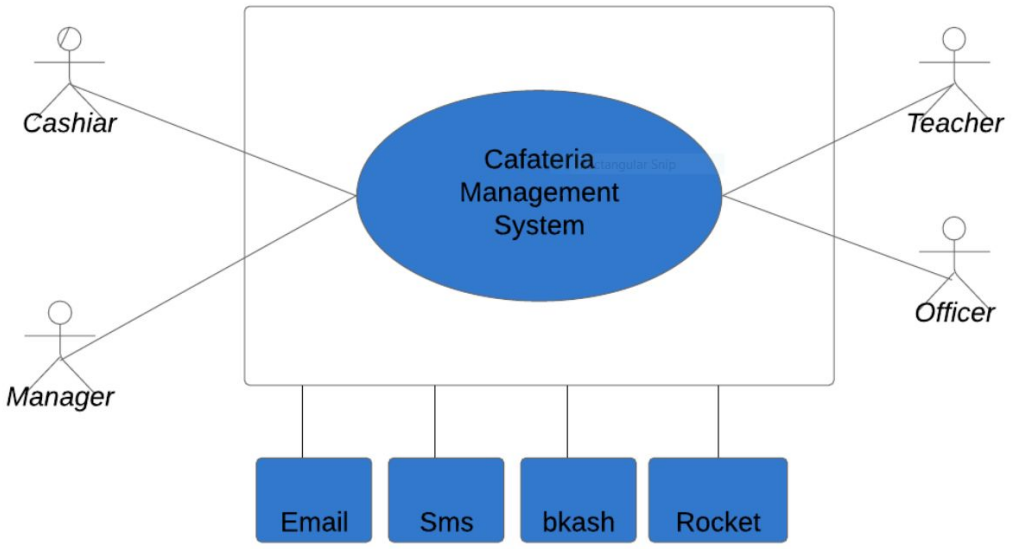


Figure – 1: Level 0 use case diagram – CMS.

Name:	CMS
Id:	CMS – L – 0
Primary actor:	Teacher, officer, manager, cashier.
Secondary actor:	Email, SMS, bkash, Rocket.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 0:

After analyzing user story, we found eight actors who will directly use the system as a system operator. Primary actors are those who play action and get reply from the system whereas secondary actors only produce or consume the information.

Following are the actors of Cafeteria Management System:

1. Teacher (Primary)
2. Officer (Primary)
3. Manager (Primary)
4. Cashier (Primary)
5. Email (Secondary)
6. SMS (Secondary)
7. Bkash (Secondary)
8. Rocket (Secondary)

4.3.2 LEVEL – 1 USE CASE DIAGRAM – Subsystem

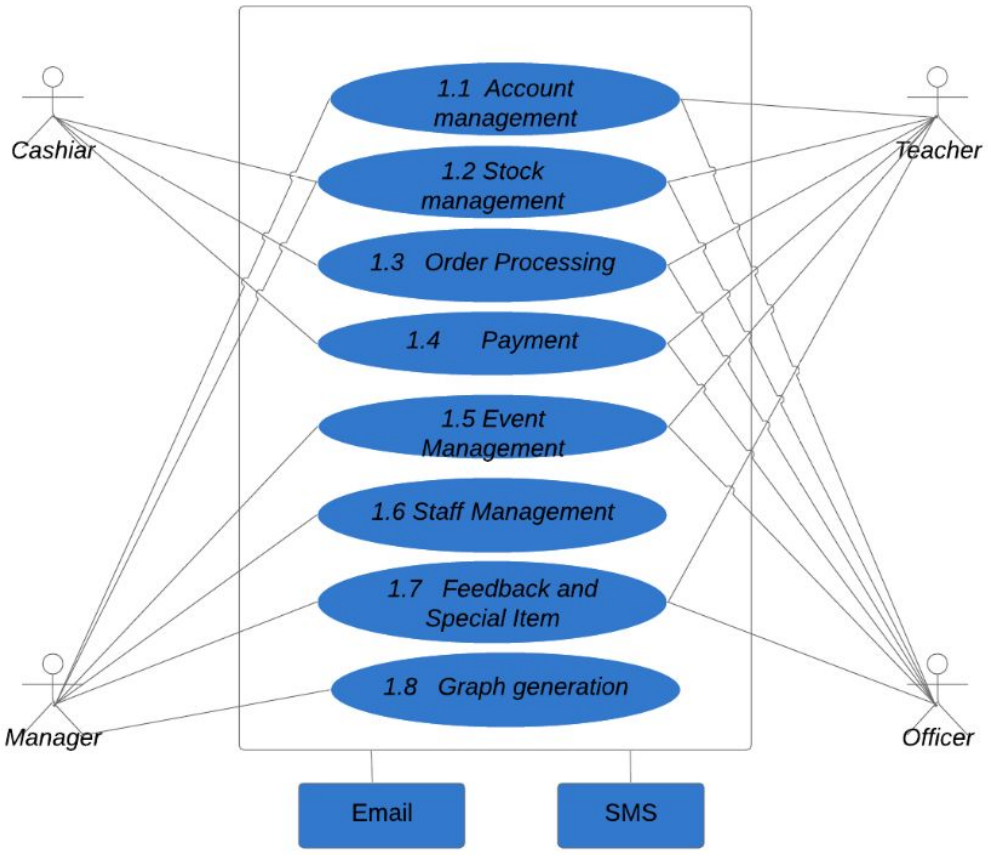


Figure – 2: Level 1 use case diagram – Subsystem.

Name:	Subsystem
Id:	CMS – L – 1
Primary actor:	Teacher, officer, manager, cashier.
Secondary actor:	Email, sms, Bkash, Rocket.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1:

There are eight subsystems in Pharmacy Management System. These are:

1. Account management.
2. Stock management.
3. Order processing.
4. Payment.
5. Event management.
6. Staff management.
7. Feedback and special item.
8. Graph generation.

4.3.3 LEVEL – 1.1 USE CASE DIAGRAM – Account management

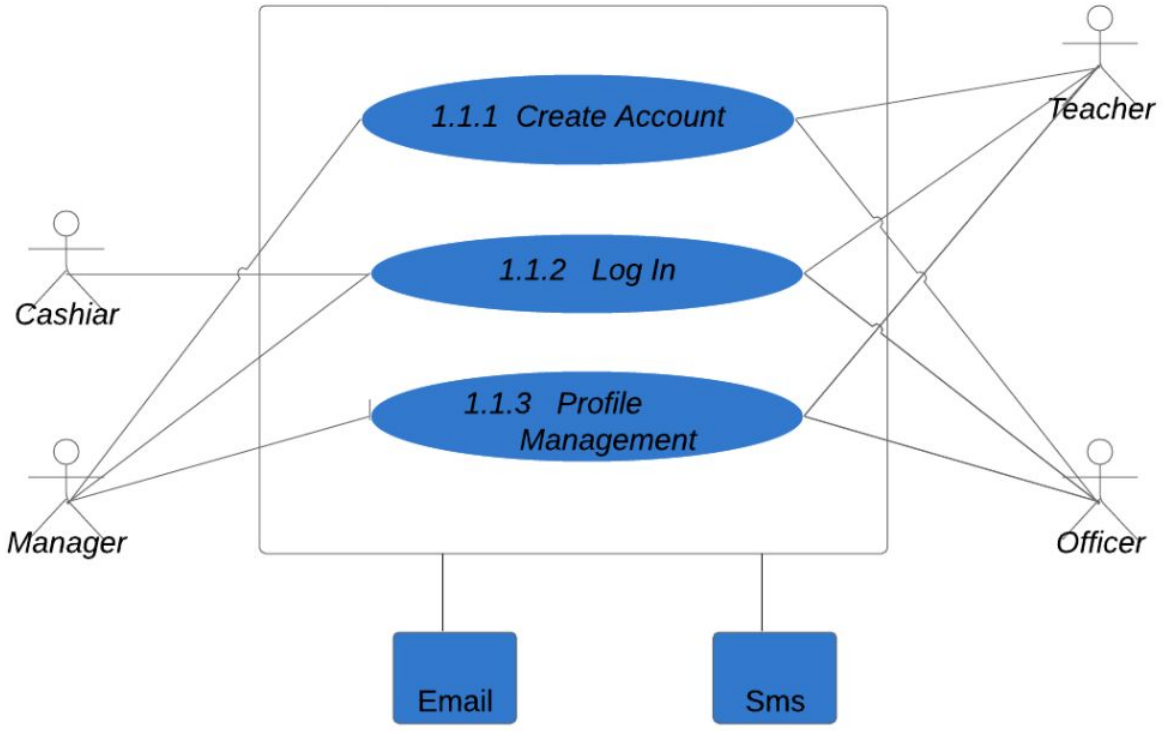


Figure – 3: Level 1.1 use case diagram – Account management.

Name:	Account management.
ID:	CMS – L – 1.1
Primary Actors:	Teacher, Officer, Manager, Cashier.
Secondary Actor:	Email, SMS,

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.1:

Account management has three parts. They are: -

1. Create account.

2. Log in.

3. Profile management.

4.3.4 LEVEL – 1.1.1 USE CASE DIAGRAM –Create account

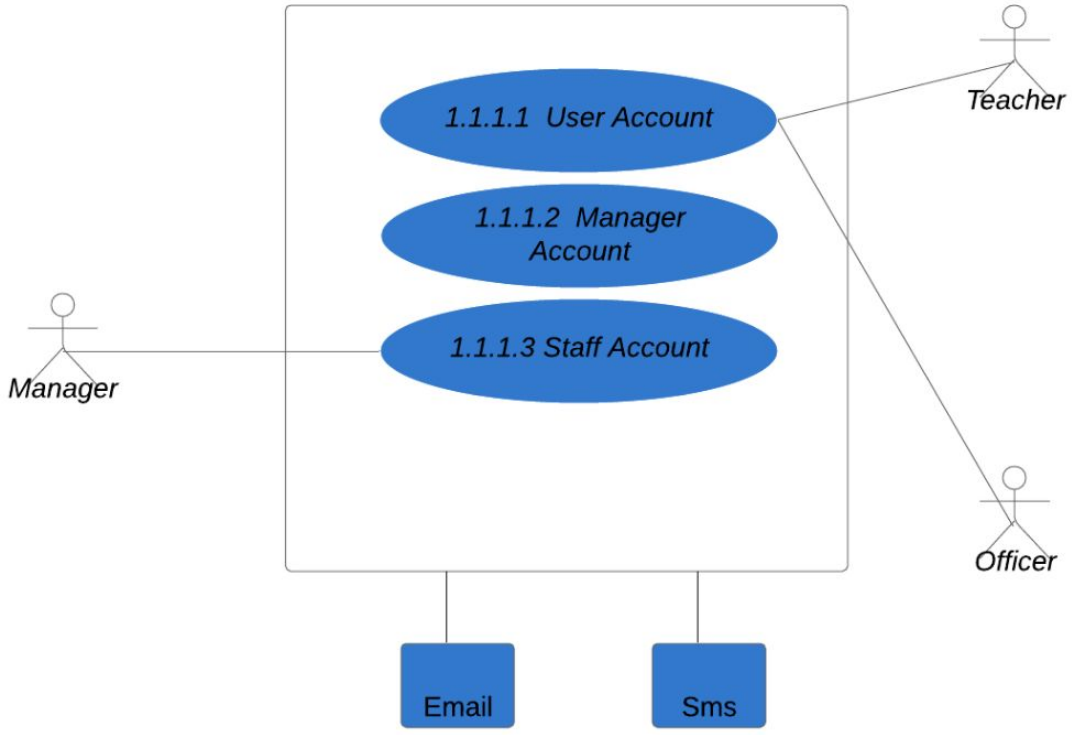


Figure – 4: Level 1.1.1 use case diagram – Create account.

Name:	Create account
ID:	CMS – L – 1.1.1
Primary Actors:	Teacher, Officer, Manager.
Secondary Actor:	Email, SMS.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.1.1:

There are three types of accounts. They are: -

1. User account
2. Manager account.
3. Staff account.

In level 1.1.1.2, manager's account will be created only by super admin before the launch of this system. Later on, the manager's account can only be edited and created by the manager. To create the manager's account username, password, email address is needed.

4.3.5 Level – 1.1.1.1 Use Case Diagram –User account

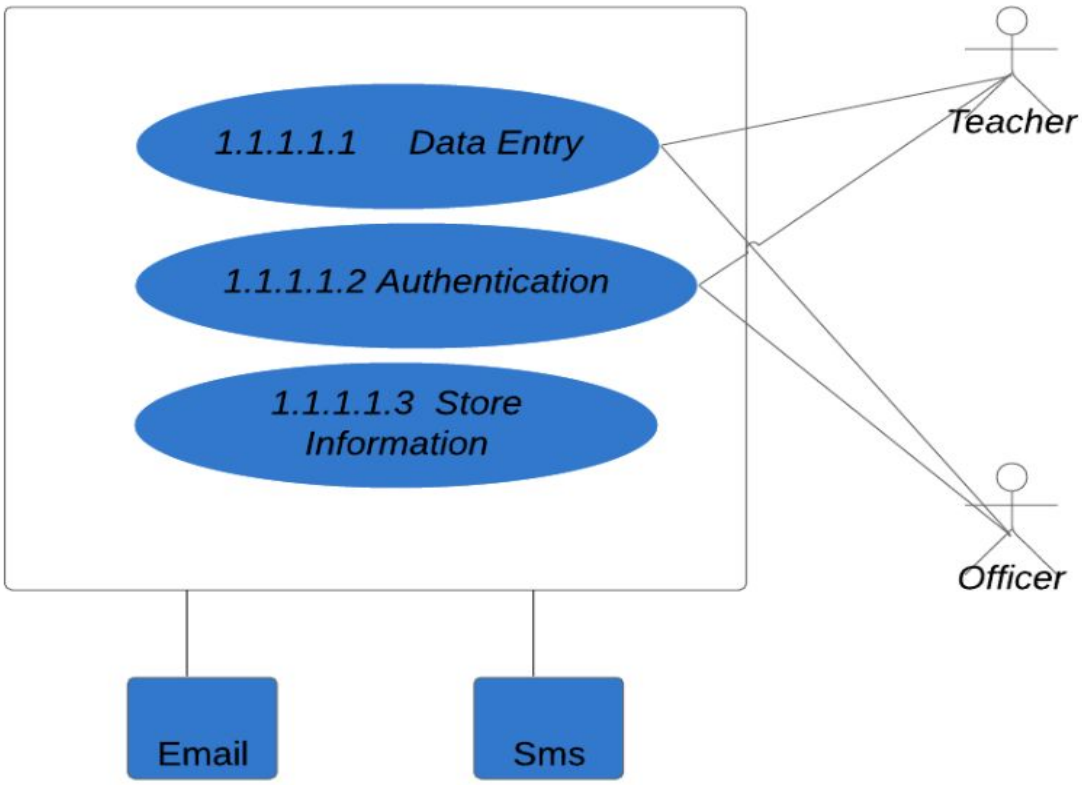


Figure – 5: Level 1.1.1.1 use case diagram – User account.

Name: User account

ID: CMS – L – 1.1.1.1

Primary Actors: Teacher, Officer.

Secondary Actor: Email, SMS.

Description Of Use Case Diagram Level – 1.1.1.1:

In level 1.1.1.1, there are two types of user-

· Teacher

· Officer

For creating user account, there are three steps. They are: -

1. Data entry.
2. Authentication.
3. Store information.

In level 1.1.1.1.1, for creating an account, user has to provide their full name, username, password, email address, mobile number, address of their respective department and room number.

Action: Users provide required data.

Reply: System checks validity of data.

In level 1.1.1.1.2, after providing data, authentication process comes. User has to provide an email which contains “du.ac.bd” domain. If the user does not provide that type of email, he must be referenced by any existing user. User can search for existing user and send him request for confirmation. After confirmation, authentication is done.

Action: User provides data for authentication.

Reply: System authenticates.

In level 1.1.1.1.3, after authentication, database stores the data.

Action: Database stores the data.

Reply: Successfully stored.

4.3.6 LEVEL – 1.1.2 USE CASE DIAGRAM –Login

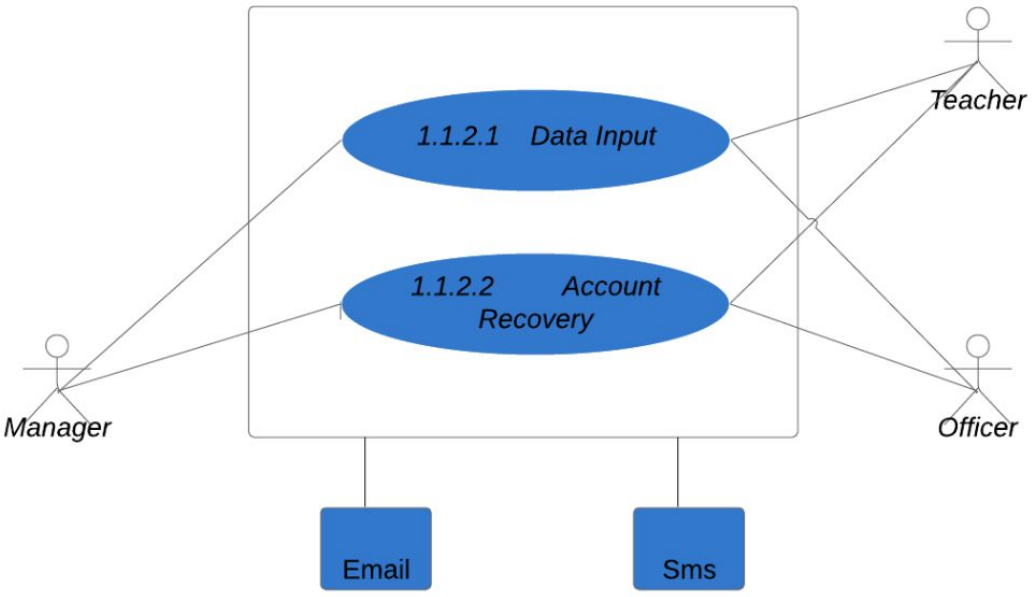


Figure – 6: Level 1.1.2 use case diagram – Login.

Name:	Login
ID:	CMS – L – 1.1.2
Primary Actors:	Teacher, Officer, Manager.
Secondary Actor:	Email, SMS.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.1.2:

In level 1.1.2, to login to the system, users have to provide their username and password. If the correct username-password pair is not provided, an error dialogue box will be shown. After 5 unsuccessful trials, an email will be sent to the user and he cannot get access to the system for 2 days. There are two parts. They are: -

1. Data input.

2. Account recovery.

In level 1.1.2.1, users have to provide username and password to log in.

Action 1: Users provide right username and password.

Reply 1: System verifies and allow users.

Action 2: Users provide wrong username and password.

Reply 2: System shows error message.

4.3.7 LEVEL – 1.1.2.2 USE CASE DIAGRAM –Account recovery

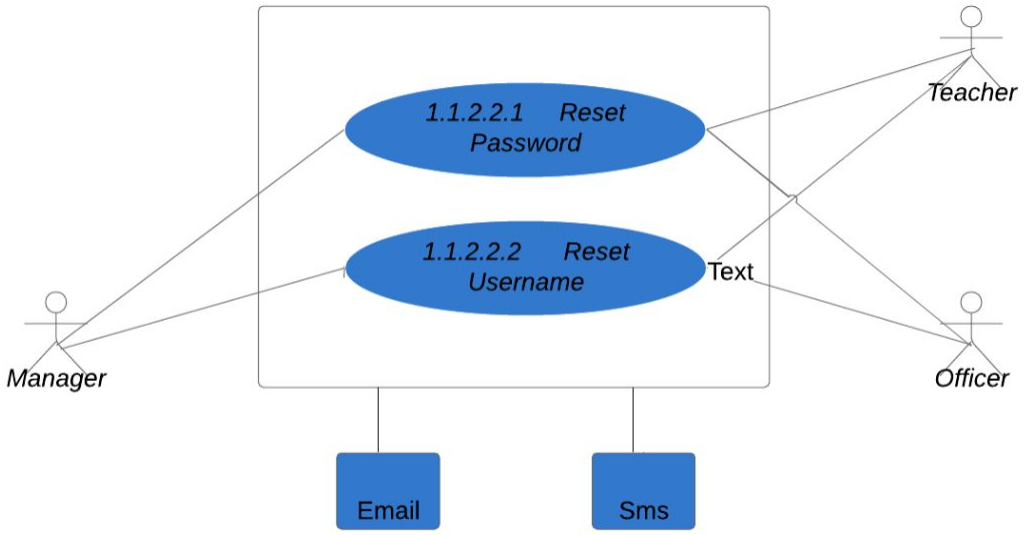


Figure – 7: Level 1.1.2.2 use case diagram – Account recovery.

Name:	Account recovery
ID:	CMS – L – 1.1.2.2

Primary Actors: Teacher, Officer, Manager.

Secondary Actor: Email, SMS.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.1.2.2:

In level 1.1.2.2, user can reset his password or username if he forgets the password or username.

Account recovery has two parts. They are: -

1. Reset password.
2. Reset username.

In level 1.1.2.2.1, If user wants to reset password, he has to choose reset password option. Then a link will be sent to the user's email where he will be asked to enter his new password.

Action: Users choose reset password option and enter new password.

Reply: System checks validity and confirms the new password.

In level 1.1.2.2.2, If user wants to reset username, he has to choose reset username option. After logging in, he can change username.

Action: Users request for changing username and enter new username.

Reply: System confirms after validation.

4.3.8 LEVEL – 1.1.3 USE CASE DIAGRAM –Profile management

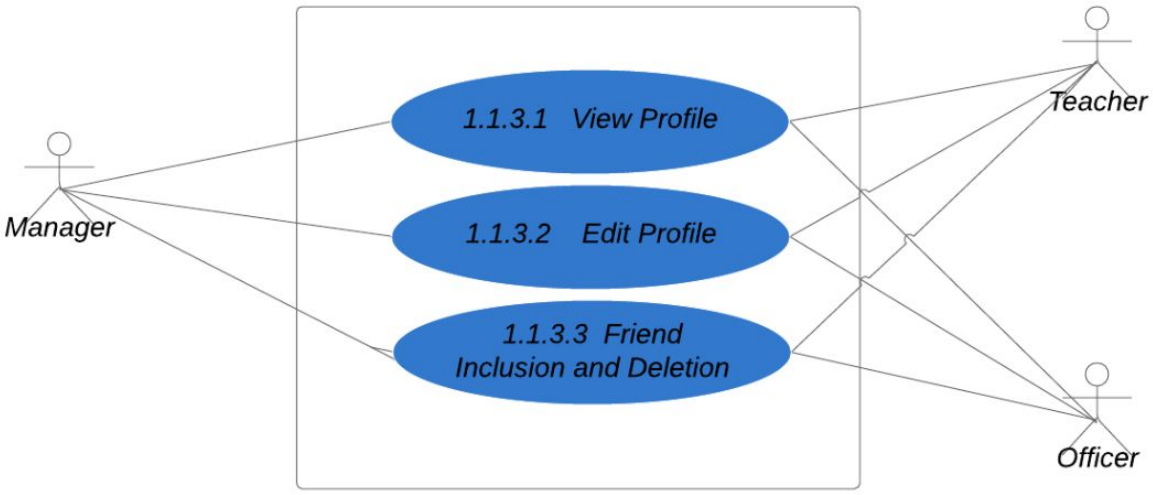


Figure – 8: Level 1.1.3 use case diagram – Profile management.

Name:	Profile management
ID:	CMS – L – 1.1.3
Primary Actors:	Teacher, Officer, Manager.
Secondary Actor:	N/A.

Description Of Use Case Diagram Level – 1.1.3:

In level 1.1.3, profile management has three parts. They are: -

1. View profile
2. Edit profile.
3. Friend inclusion and deletion.

In level 1.1.3.1, users can view details of their profiles.

Action: After logging in, users will choose viewing their profile.

Reply: System shows profile after verification.

In level 1.1.3.2, users can edit the details of their profiles.

Action: After logging in, users will choose edit profile option and enter new data.

Reply: System verifies and permit the users to make changes to the profile.

In level 1.1.3.3, users can search friends. After searching, users can add and delete friends. Then users can suggest food items to their friends.

Action: Users search for friends, add and delete them from friend list.

Reply: System responses according to the request.

4.3.9 LEVEL – 1.2 USE CASE DIAGRAM –Stock management

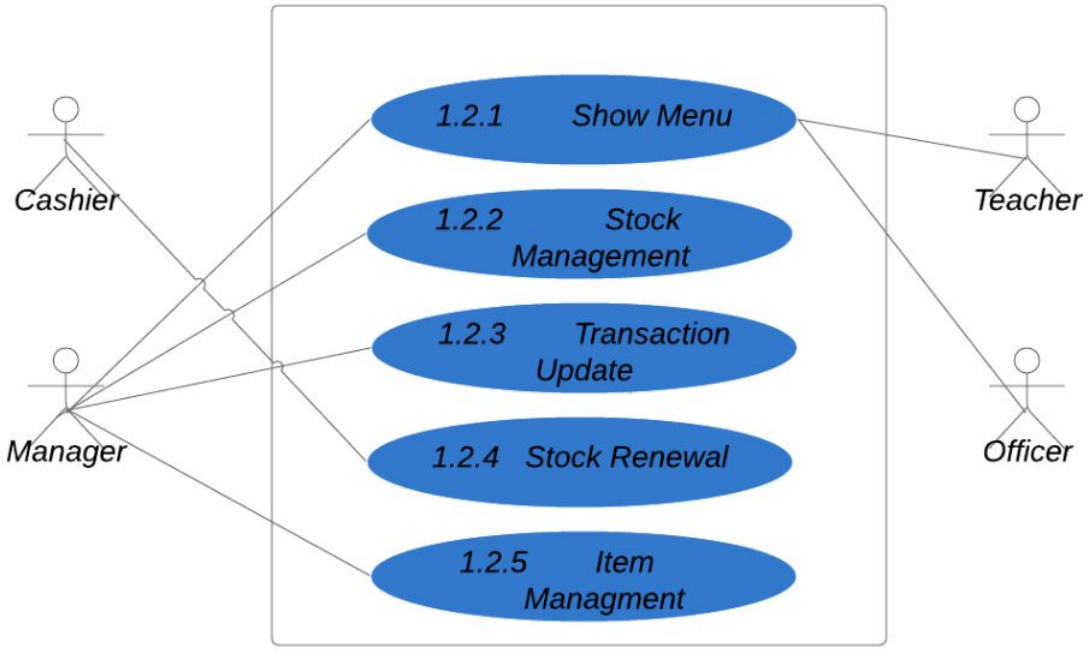


Figure – 9: Level 1.2 use case diagram – Stock management.

Name:	Stock management
ID:	CMS – L – 1.2
Primary Actors:	Teacher, Officer, Manager, Cashier.
Secondary Actor:	N/A.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.2:

In level 1.2, stock management has two perspectives. One is for user and the other one is for cafeteria. The whole system can be divided into five parts. They are: -

1. Show menu.
2. Stock reservation.

3. Transection update.
4. Stock renewal.
5. Item management.

In level 1.2.2, manager maintains stock reservation. He has to keep count for the raw materials needed for cooking and the amount of food which will be updated in the database.

Action: Manager performs the initial update.

Reply: Successfully updated.

In level 1.2.3, manager maintains transactional updates. He updates the database after every transaction.

Action: After every transaction, manager updates the database.

Reply: Database successfully updated.

In level 1.2.4, all the records of stock reservation are stored in the database. So, whenever the stock is getting shorter than required, an automatic signal will be given to the system for stock renewal.

Action: Database will check the quantity of the stock.

Reply: System gives a signal for renewal if the stock is short.

4.3.10 LEVEL – 1.2.1 USE CASE DIAGRAM –Show menu

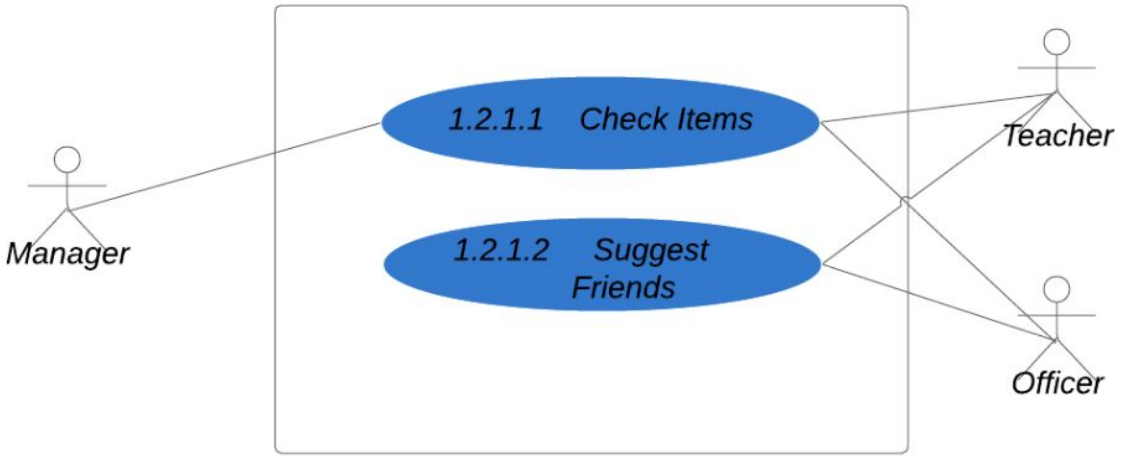


Figure – 10: Level 1.2.1 use case diagram – Show menu.

Name: Show menu.

ID: CMS – L – 1.2.1

Primary Actors: Teacher, Officer, Manager.

Secondary Actor: N/A.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.2.1:

There are two parts. They are: -

1. Check items.
2. Suggest friends.

In level 1.2.1.1, users can check items.

Action: Users check items.

Reply: System shows the menu.

In level 1.2.1.2, users can suggest food items to their friends using the same system.

Action: Users suggest food item to friends.

Reply: System sends notification.

4.3.11 LEVEL – 1.2.5 USE CASE DIAGRAM –Item management

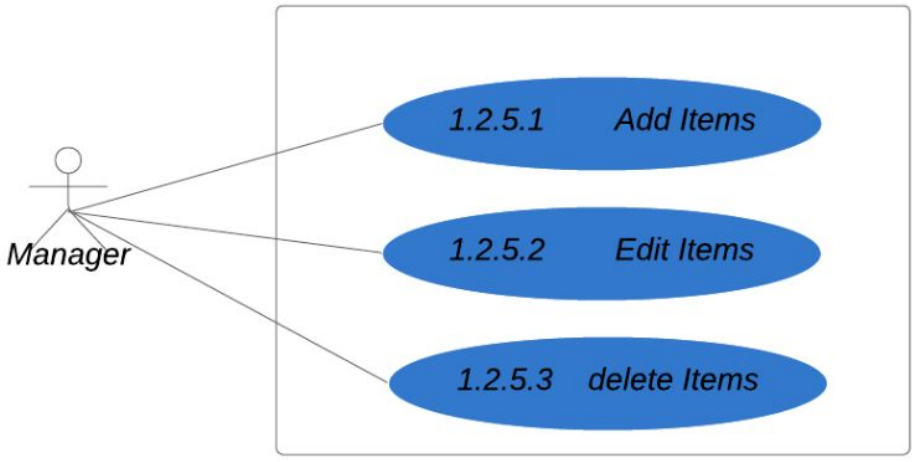


Figure – 11: Level 1.2.5 use case diagram – Item management.

Name: Item management

ID: CMS – L – 1.2.5

Primary Actors: Manager.

Secondary Actor: N/A.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.2.5:

In level 1.2.5, there are three parts. They are: -

1. Add items.
2. Edit items.
3. Delete items.

In level 1.2.5.1, manager can add items to the menu.

Action: Manager add items to the menu.

Reply: System updates the change to the database.

In level 1.2.5.2, manager can edit items to the menu.

Action: Manager edits items to the menu.

Reply: System updates the change.

In level 1.2.5.3, manager can delete items from the menu.

Action: Manager deletes items from the menu.

Reply: System updates the changes.

4.3.12 LEVEL – 1.3 USE CASE DIAGRAM –Order processing

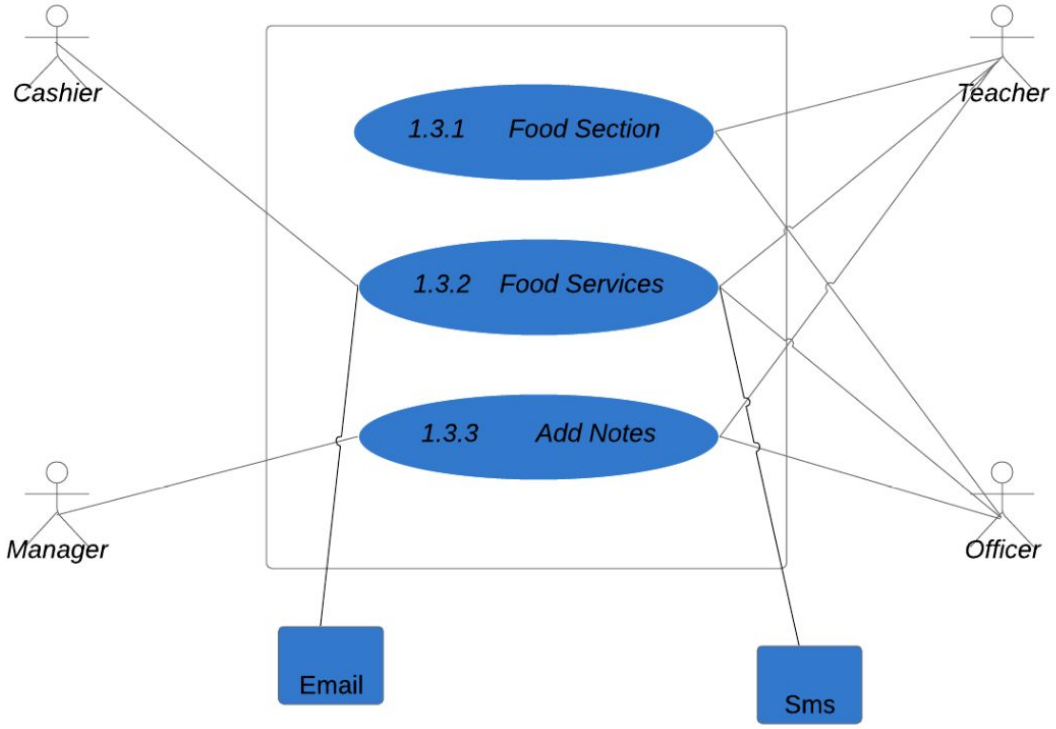


Figure – 12: Level 1.3 use case diagram – Order processing.

Name: Order processing

ID: CMS – L – 1.3

Primary Actors: Teacher, Officer, Manager, Cashier.

Secondary Actor: Email, SMS.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.3:

In level 1.3, there are three sub sections. They are: -

1. Food selection.
2. Food services.
3. Add notes.

In level 1.3.1, users can choose food items from the menu.

Action: User selects food from the menu.

Reply: System stores the selection.

In level 1.3.3, Users can add notes about the food.

Action: User adds notes about food.

Reply: System stores the note.

4.3.13 LEVEL – 1.3.2 USE CASE DIAGRAM –Food services

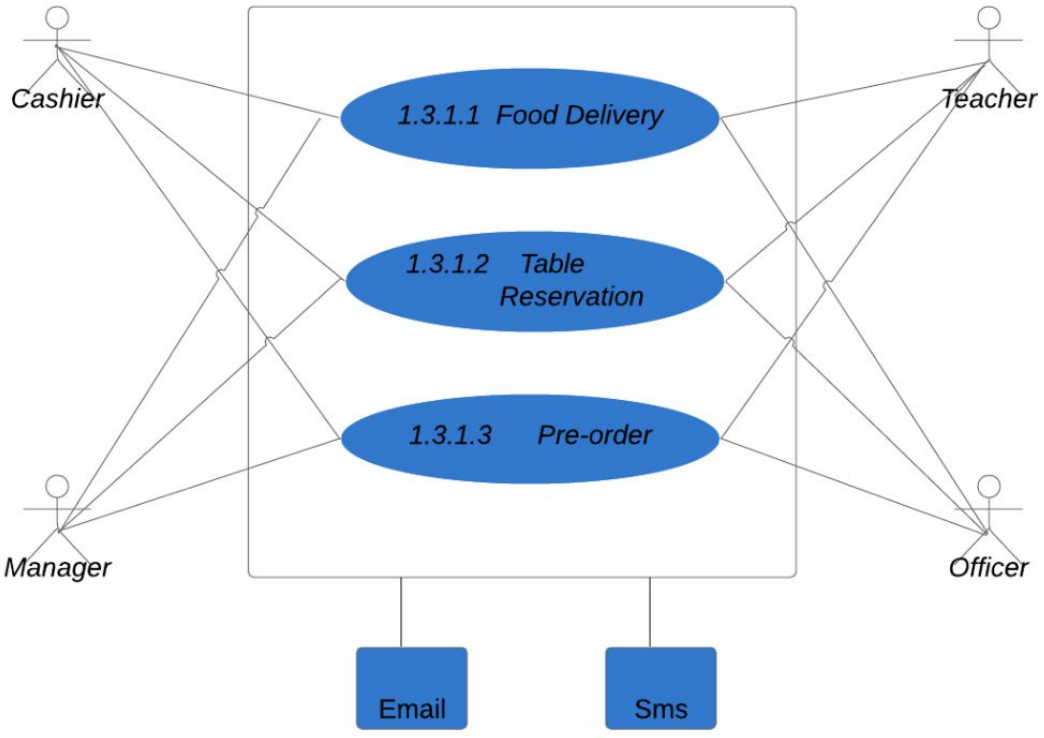


Figure – 13: Level 1.3.2 use case diagram – Food services.

Name: Food services.

ID: CMS – L – 1.3.2

Primary Actors: Teacher, Officer, Manager, Cashier.

Secondary Actor: Email, SMS.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.3.2:

In level 1.3.2, there are three parts. They are: -

1. Food delivery.
2. Table reservation.
3. Pre-order.

In level 1.3.2.1, user orders his preferred food for delivery, a cashier will print a memo and give it to the available delivery-man. The delivery-man will deliver the food to the given address. Both the delivery-man and the cashier's name will be stored in the database. Thirty takas will be charged for each delivery.

Action: User orders food online.
Reply: System stores the data and the delivery-man delivers the food to the given address.

In level 1.3.2.2, user can reserve a table for Dine-in. For booking a table, a user has to order at least 40 minutes earlier. A user will be given an order-code through email and SMS. When the user comes to the cafeteria, he has to show the code to the cashier to get the food. If the user does not come within 30 minutes from the booking time, his booking will be cancelled.

Action: User requests for table reservation.
Reply: System stores the response.

In level 1.3.2.3, Users can just order the food with the system without table reservation and come at his preferred time. He will be given an order-code through email and SMS. When he comes to the cafeteria, he needs to show his order-code before having the meal.

Action: Users can make pre-order and get a code after confirmation.
Reply: System responses according to the request.

4.3.14 LEVEL – 1.4 USE CASE DIAGRAM – Payment

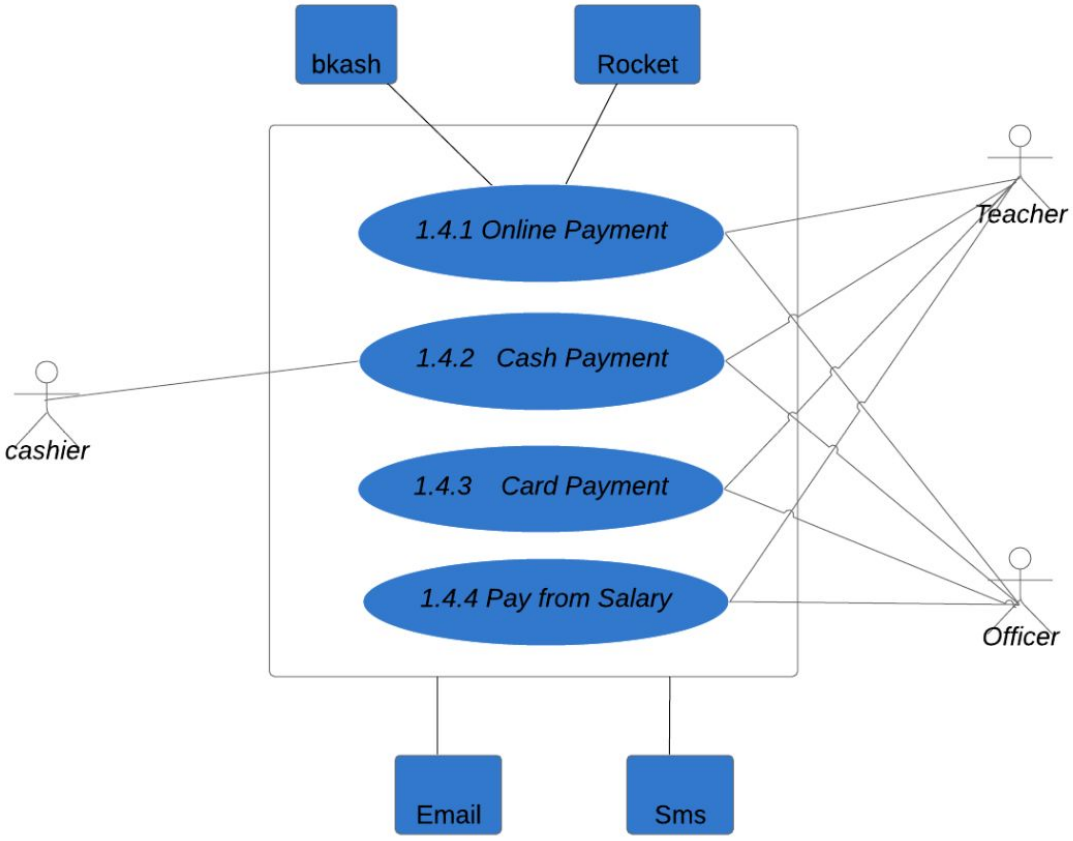


Figure – 14: Level 1.4 use case diagram – Payment.

Name: Payment

ID: CMS – L – 1.4

Primary Actors: Teacher, Officer, Cashier.

Secondary Actor: Email, SMS, BKash, Rocket.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.4:

In level 1.4, payment methods are included here. They are: -

1. Online payment.
2. Cash payment.
3. Card payment.
4. Pay from salary.

In level 1.4.1, users can pay through BKash and Rocket account. They have to log in to their accounts and the total cost of the order will be shown there. After confirmation, the payment will be completed and the user will be notified through email and sms.

Action: Users choose online payment option.

Reply: System confirms if the payment is done or not through notification and stores the information.

In level 1.4.2, users can pay bill in the counter. Cashier will store the customer's name to the database. After receiving the money, the cashier confirms the payment completion.

Action: Users pay cash to the cashier and cashier enters the data to the system.

Reply: System stores the data in the database.

In level 1.4.3, users can pay through cards. After logging in and ordering food, the system will show the cost to the user. Then users can pay by swiping the card in the EDC machine. After confirmation, the payment will be completed and users will be notified through email and sms.

Action: Users choose card payment method.

Reply: System confirms if the payment is done or not and stores the information.

In level 1.4.4, users can choose pay from salary. Every day, after ordering food, a notification will be sent to the user and he will be asked for confirmation. At the end of the month, the total bill will be sent to the University authority and the bill will be cut from the user's salary. The money will be transferred to the cafeteria.

Action: Users choose pay from salary option.

Reply: System stores data and sends the bill to the customer for confirmation.

4.3.15 LEVEL – 1.5 USE CASE DIAGRAM –Event management

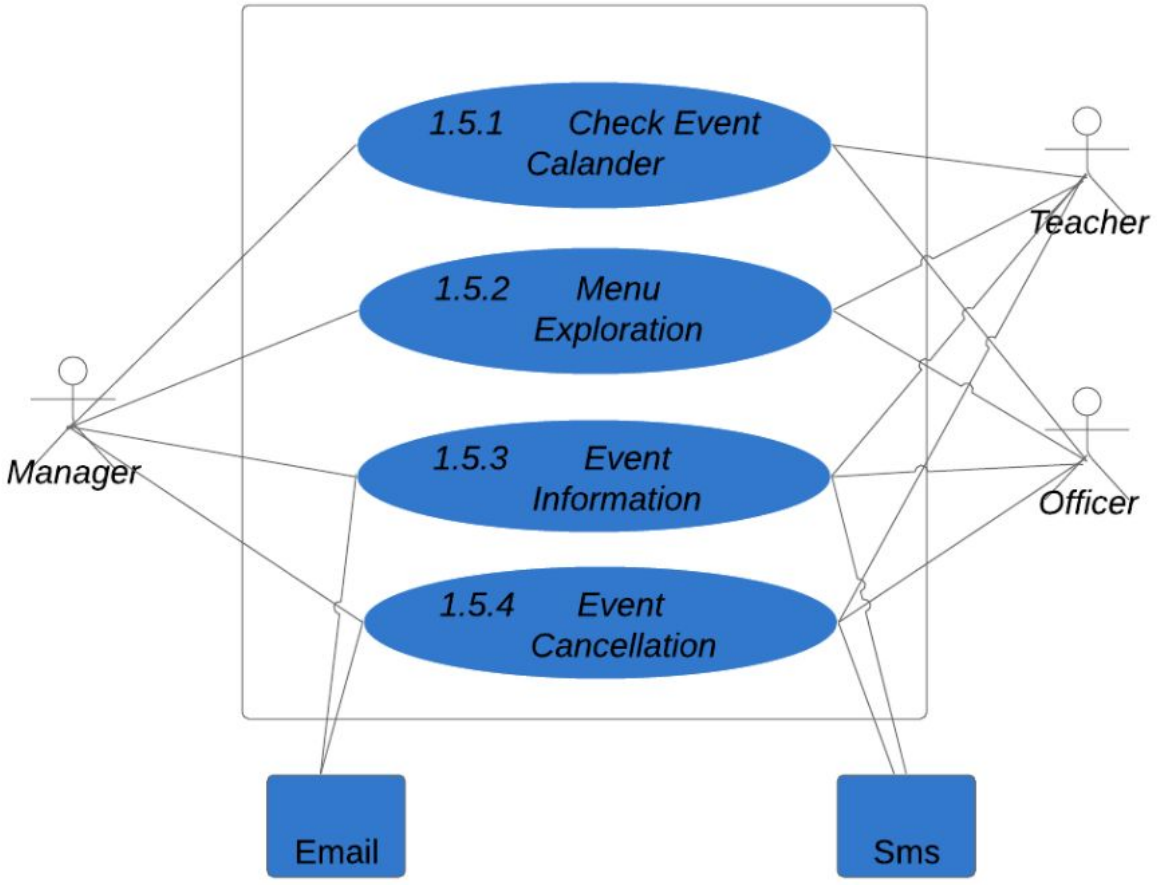


Figure – 15: Level 1.5 use case diagram – Event management.

Name: Event management.

ID: CMS – L – 1.5

Primary Actors: Teacher, Officer, Manager.

Secondary Actor: Email, SMS.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.5:

In level 1.5, there are four parts. They are: -

1. Check event calendar.
2. Menu exploration.
3. Event information.
4. Event cancellation.

In level 1.5.1, there will be an event calendar for the manager to see and edit. Manager can see which dates are booked for an event. The user has to let cafeteria management know about the event at least 3 days ago.

Action: User chooses a date for event and manager checks the event calendar.

Reply: System stores the data.

In level 1.5.4, for any reason if a user wants to cancel the event, there will be an option for the user to cancel the event. Cancellation must happen 72 hours before. Otherwise, cafeteria management will take 15% of the cost. For an event, 10% of service charge will be added.

Action: User cancels an event booking.
Reply: System stores it and manager supervises it.

4.3.16 LEVEL – 1.5.2 USE CASE DIAGRAM –Menu exploration

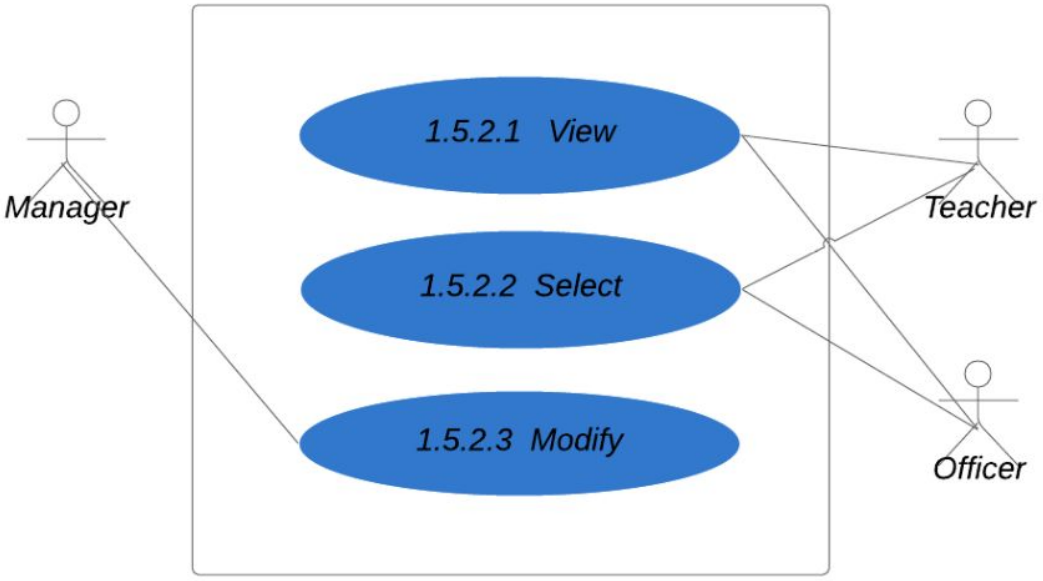


Figure – 16: Level 1.5.2 use case diagram – Menu exploration.

Name: Menu exploration.

ID: CMS – L – 1.5.2

Primary Actors: Teacher, Officer, Manager.

Secondary Actor: N/A.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.5.2:

In level 1.5.2, user can view and choose any package from the menu for the event. Manager can modify menu.

Action1: User selects menu for the event.

Reply1: System stores the data.

Action2: Manager modifies the menu.

Reply2: System stores the changes.

4.3.17 LEVEL – 1.5.3 USE CASE DIAGRAM –Event information

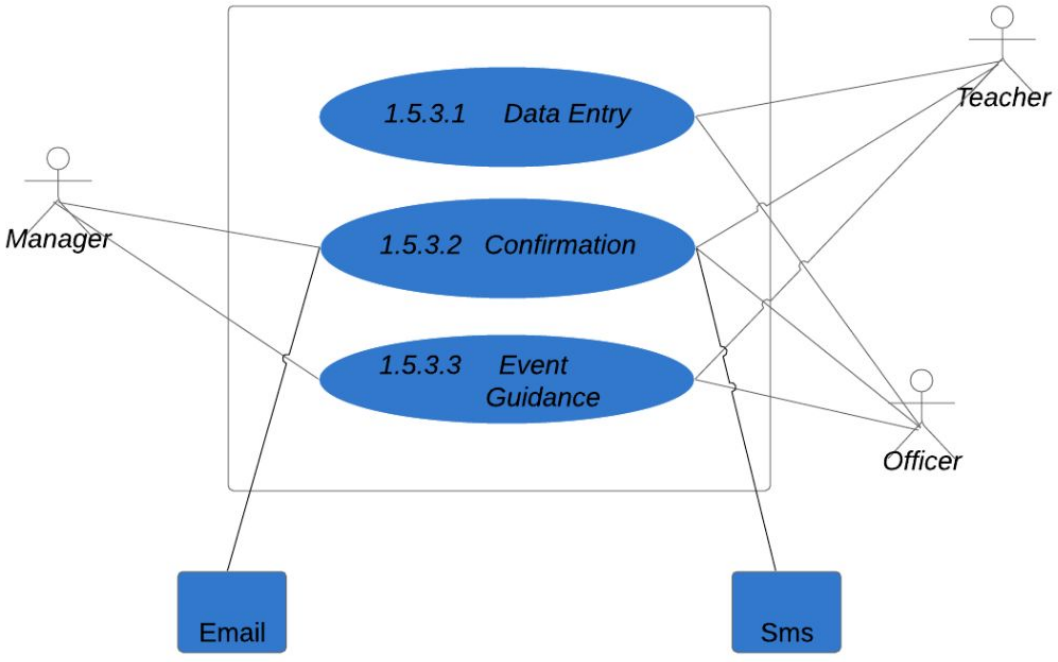


Figure – 17: Level 1.5.3 use case diagram – Event information.

Name: Event information.

ID: CMS – L – 1.5.3

Primary Actors: Teacher, Officer, Manager.

Secondary Actor: Email, SMS.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.5.3:

In level 1.5.3, user can enter event data to the system. After the confirmation, system will store the event information. User can also know about the event guidelines.

Action: User enters event data.

Reply: System confirms if possible and stores data.

4.3.18 LEVEL – 1.6 USE CASE DIAGRAM – Staff management

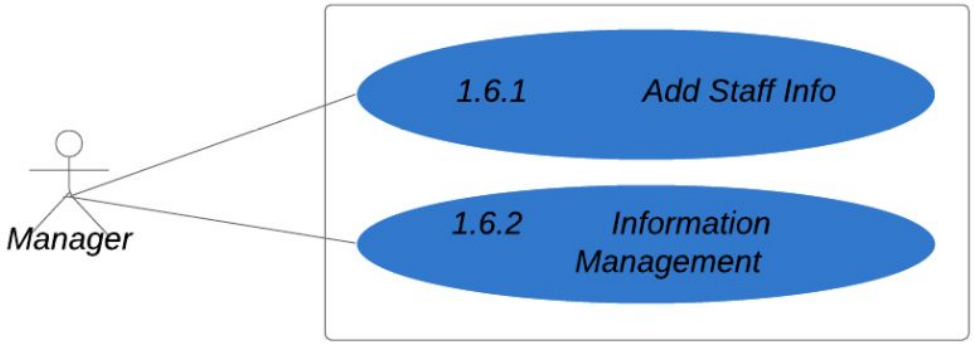


Figure – 18: Level 1.6 use case diagram – Staff management

Name: Staff management

ID: CMS – L – 1.6

Primary Actors: Manager.

Secondary Actor: N/A.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.6:

In level 1.6, there are two parts of stall management.

1. Add staff information.

2. Information management.

In level 1.6.1 manager maintains the information of the staff. Joining date of the staff, attendance, posts, salary of the staff of the cafeteria will be stored in the system's database. Manager can see and modify their stuff's attendance, post and salary. He also chooses at least three of the staff for food delivery.

Action: Manager enters data of the staff.

Reply: Database stores the data of the staff.

4.3.19 LEVEL – 1.6.2 USE CASE DIAGRAM –Information management

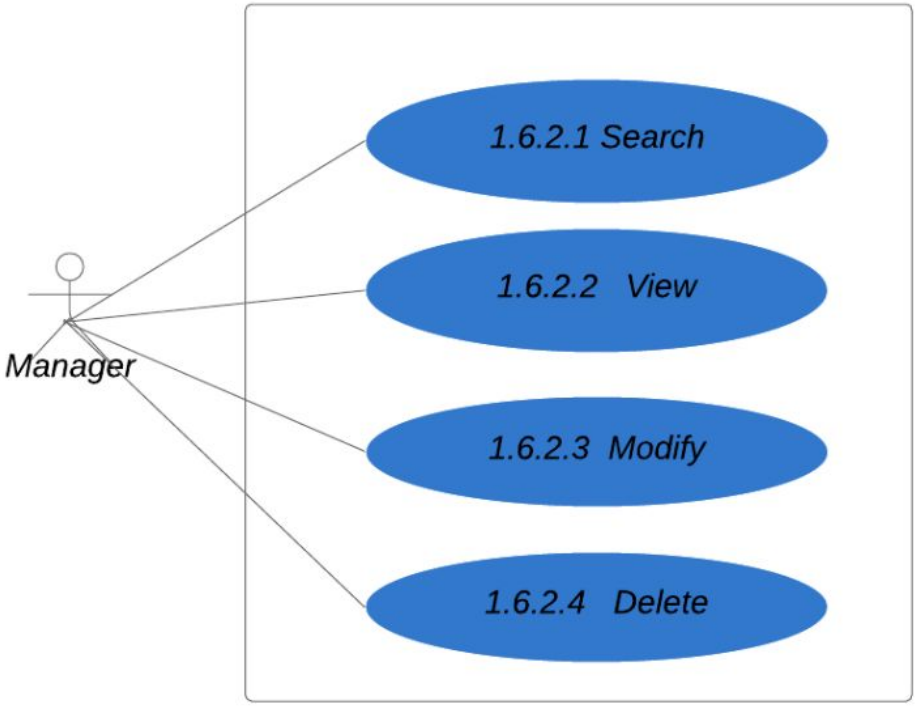


Figure – 19: Level 1.6.2 use case diagram – Information management

Name: Information management

ID: CMS – L – 1.6.2

Primary Actors: Manager.

Secondary Actor: N/A.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.6.2:

In level 1.6.2.1, manager can search the staff's information.

Action: Manager searches the staff's information.

Reply: System searches for that information in the database if it is there or not.

In level 1.6.2.2, manager can view the staff's information.

Action: Manager requests for viewing staff's information.

Reply: System searches for the result and show it if it is found.

In level 1.6.2.3, manager can modify the staff's information.

Action: Manager requests for modification.

Reply: System permits after validation.

In level 1.6.2.4, manager can delete staff's information.

Action: Manager requests for deletion.

Reply: System permits after validation.

4.3.20 LEVEL – 1.7 USE CASE DIAGRAM – Feedback and special item

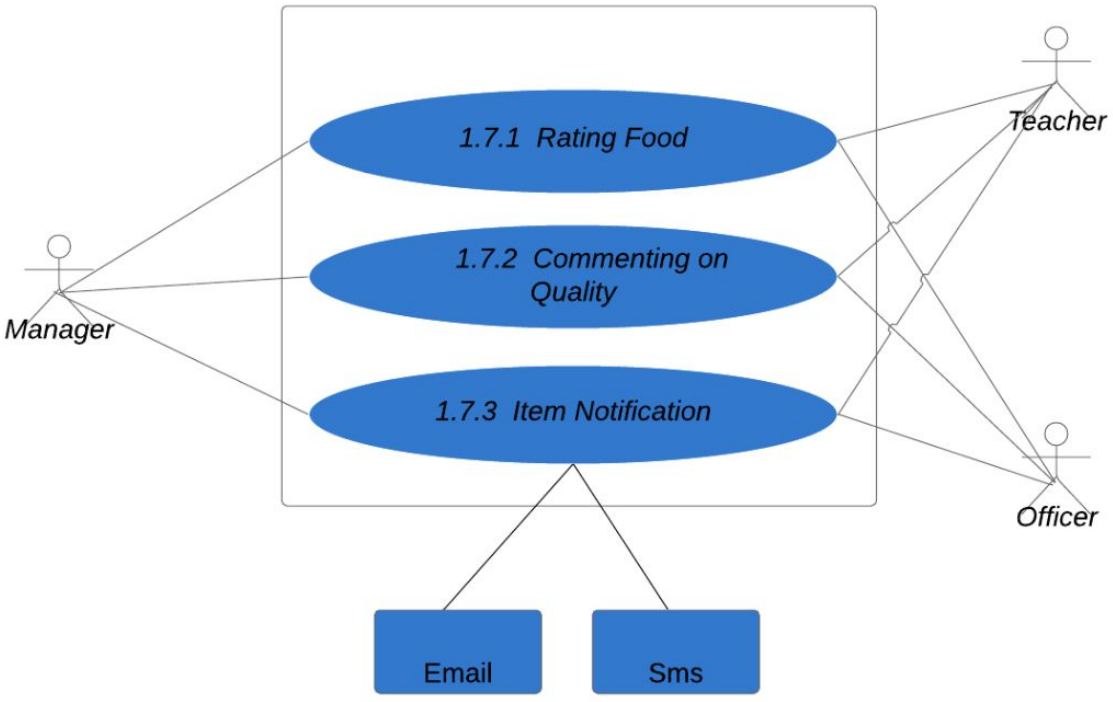


Figure – 20: Level 1.7 use case diagram – Feedback and special item.

Name: Feedback and special item

ID: CMS – L – 1.7

Primary Actors: Teacher, Officer, Manager.

Secondary Actor: Email, SMS.

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.7:

This section has three parts.

1. Rating food.
2. Commenting on quality.
3. Item notification.

In 1.7.1, user can rate the food items and manager will monitor the ratings.

Action: User provides ratings for food items.

Reply: System accepts it and store it.

In 1.7.2, user can comment on the food quality and manager will monitor it.

Action: User provides comments on food quality.

Reply: System stores the comments for further betterment.

In 1.7.3, user can choose if he wants the update of the special food items or not.

Action: User provides his choice if he wants to get update or not.

Reply: System keeps updating the user according to his choice.

4.3.21 LEVEL – 1.8 USE CASE DIAGRAM – Graph generation

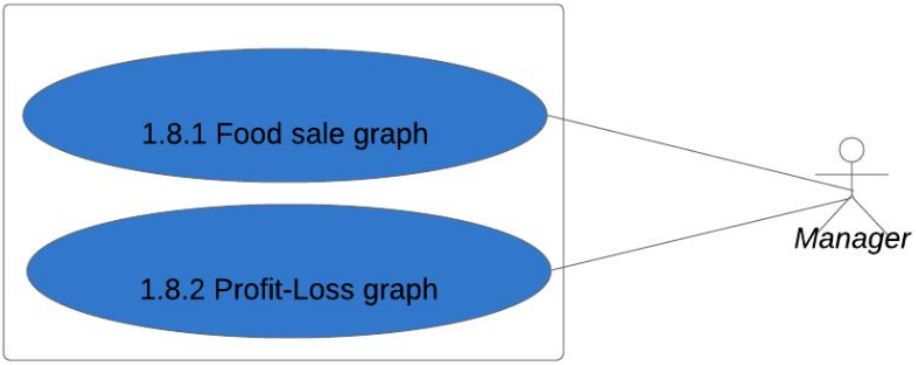


Figure – 21: Level 1.8 use case diagram – Graph generation.

Name: Graph generation

ID: CMS – L – 1.8

Primary Actors: Manager

Secondary Actor: N/A

DESCRIPTION OF USE CASE DIAGRAM LEVEL – 1.8:

In 1.8, the system will generate two types of graphs.

· Food sale

· Profit-loss.

In 1.8.1, the system represents a graph of food sale. Here we can get two types of sale graphs. One is daily sales graph. It will show hourly sales for each food items. Time ranged sales graph will represent weekly or monthly total sales. Manager can choose which graph he wants to see. As all the sale's information are stored in the database, so it is easy for manager to generate graph from that information.

Action: Manager requests for sales graph generation and provides required information.

Reply: Subsystem generates graph if the information is valid.

In 1.8.2, the system shows a profit-loss graph. For this graph, some calculations are needed, so there is a formula: -

Profit=Earned money-expenses.

Earned money and expenses will be calculated from the database. All money transactions will be recorded in the database. From the list of sold food items and event booking, earned money can be calculated. Cashier will upload all information to the database. Manager will verify that information.

Action: Manager requests for profit-loss graph generation and provides required information.

Reply: Subsystem generates graph after validating the information and calculation of profit.

4.4 ACTIVITY DIAGRAMS

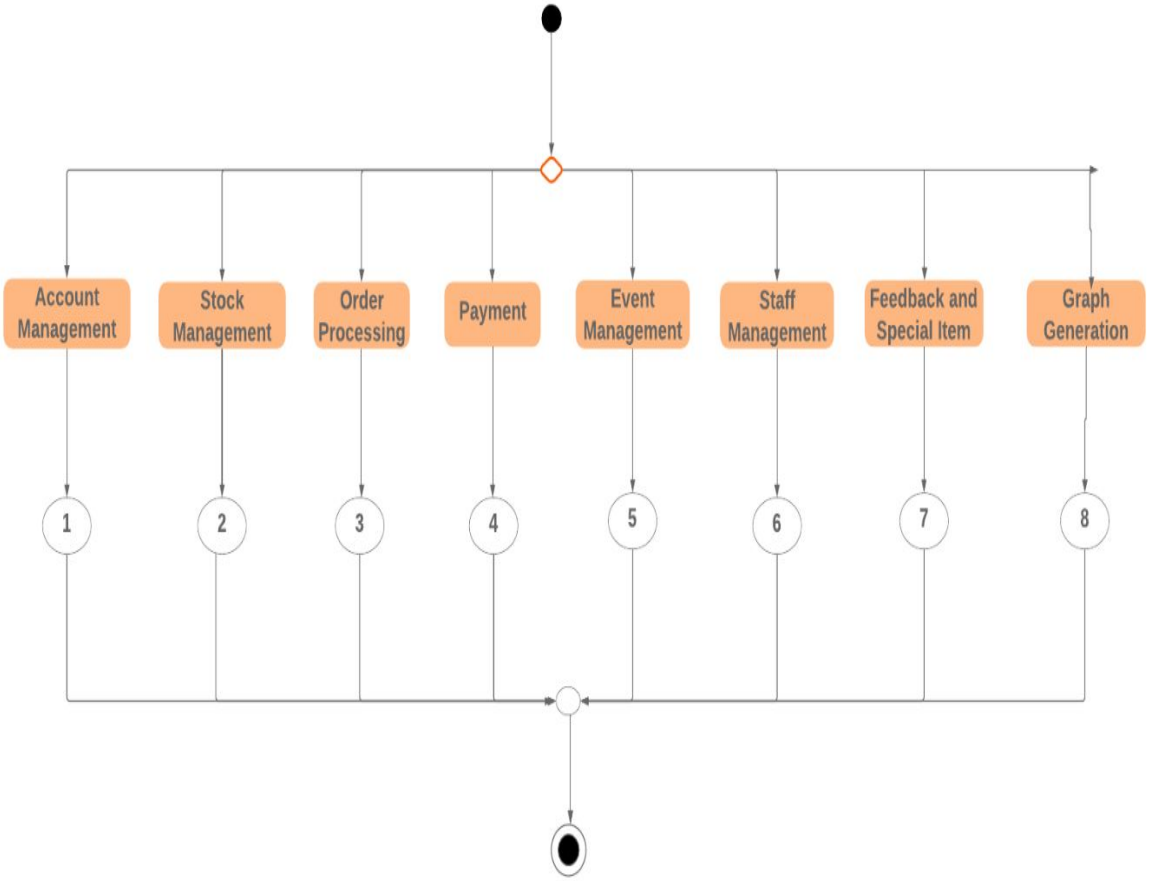


Figure 22: Activity Diagram 1: Cafeteria Management System.

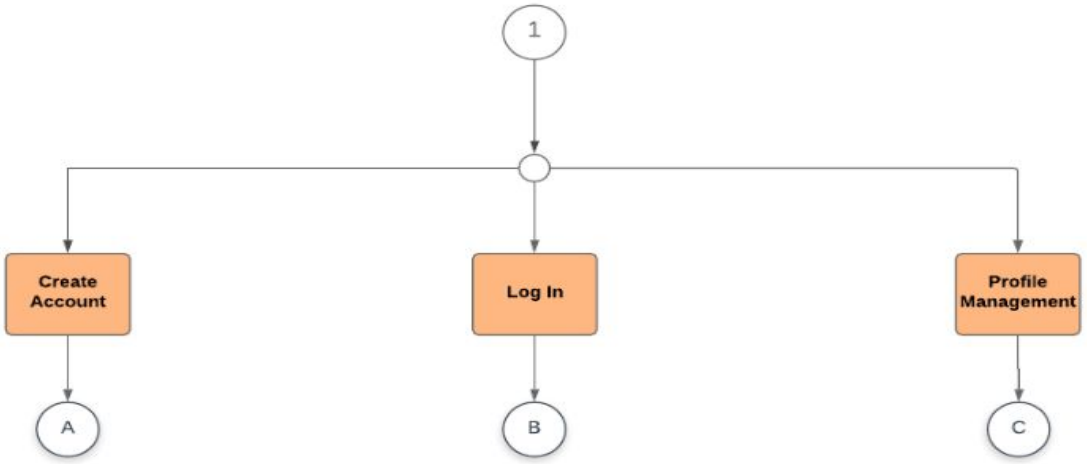


Figure 23: Activity Diagram 2: Account management.

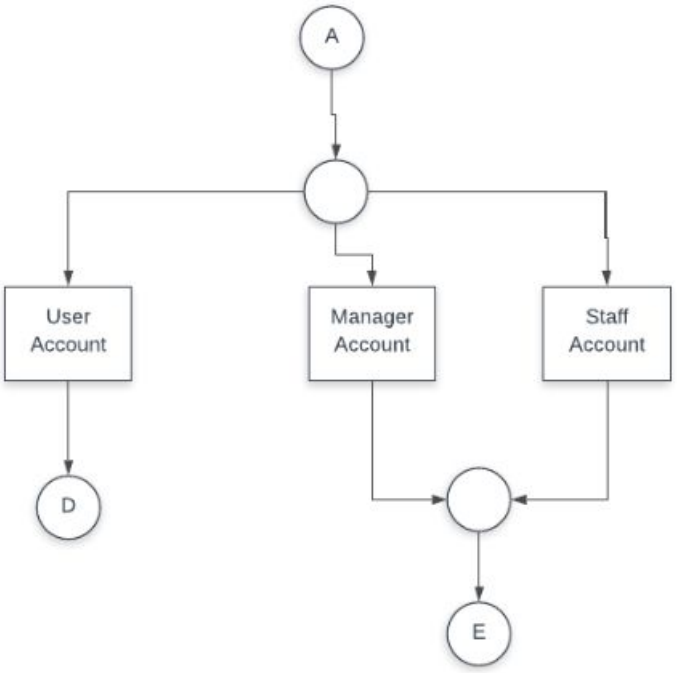


Figure 24: Activity Diagram 3: Create account.

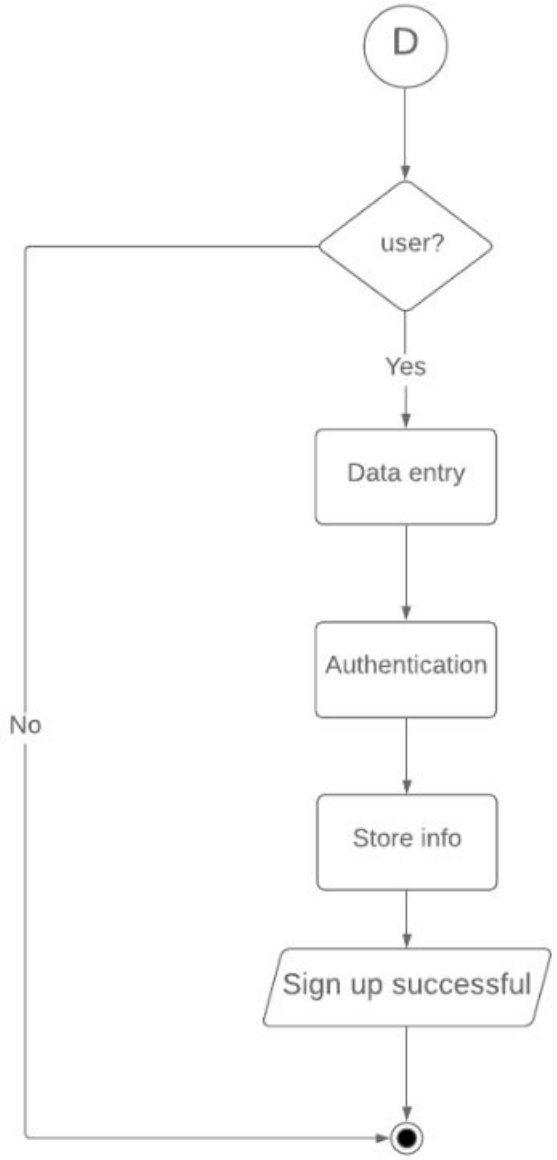


Figure 25: Activity Diagram 4: User account.

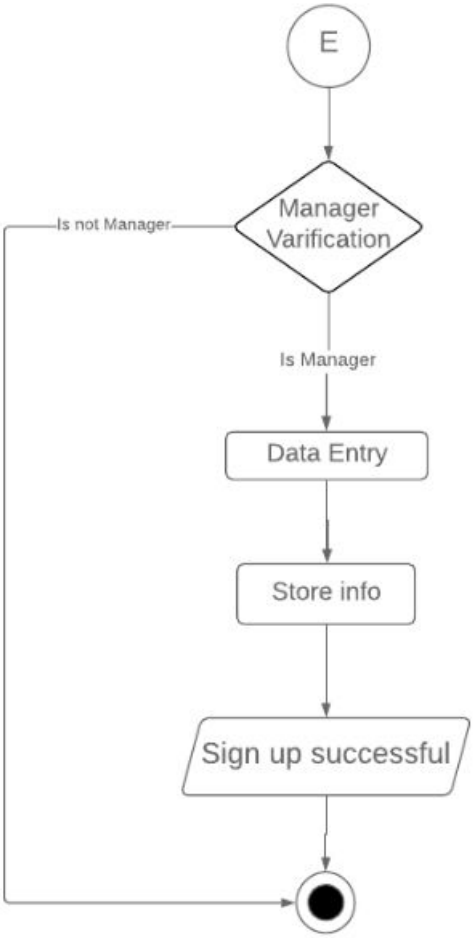


Figure 26: Activity Diagram 5: Manager account.

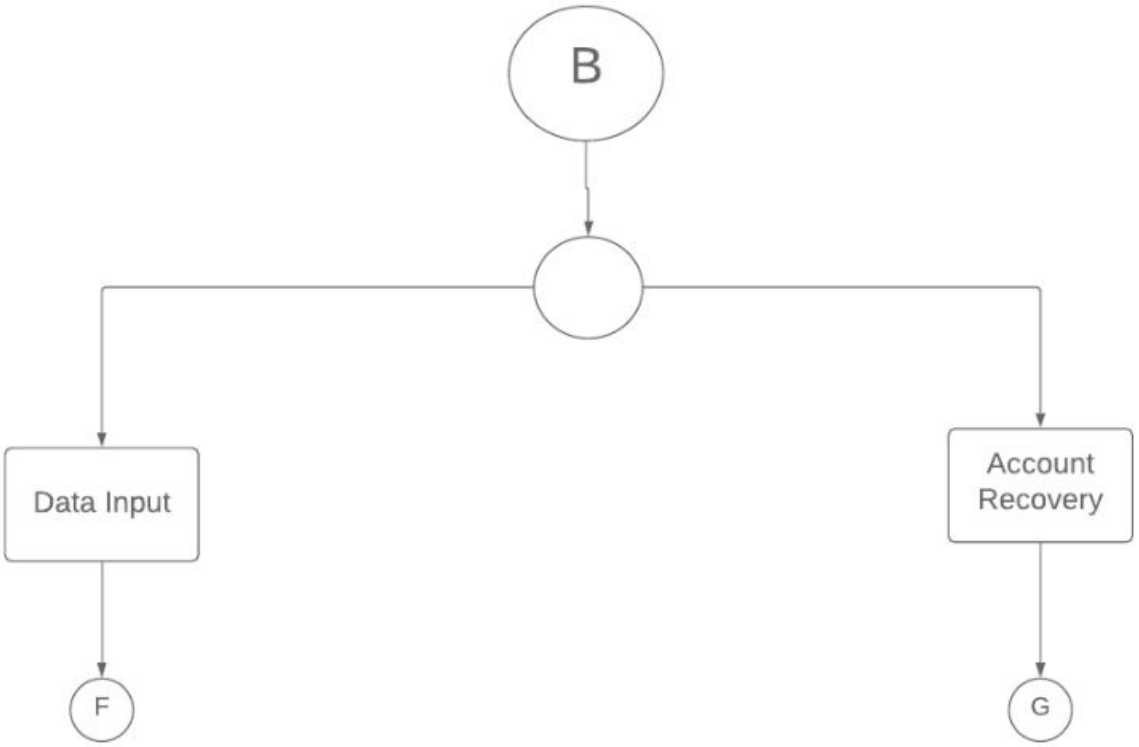


Figure 27: Activity Diagram 6: Login.

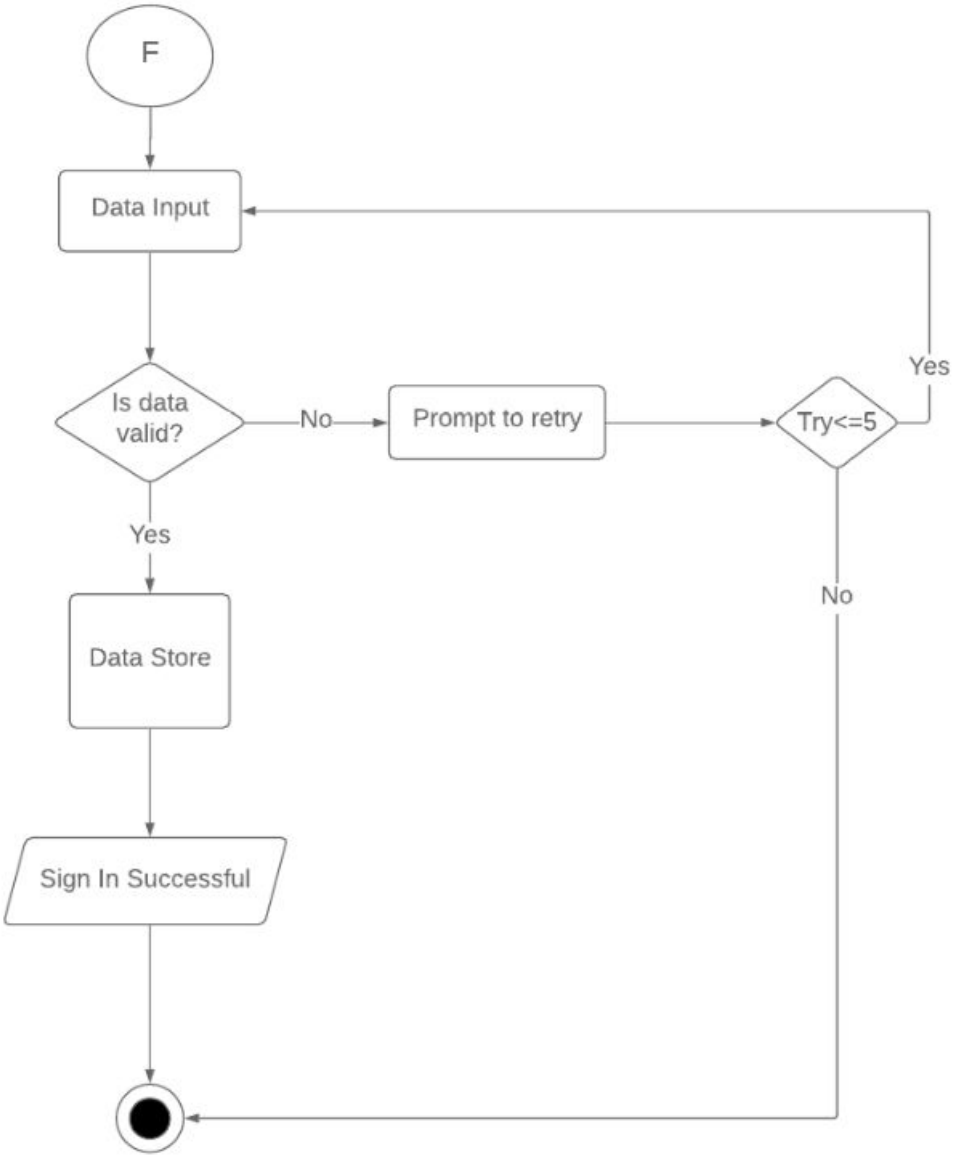


Figure 28: Activity Diagram 7: Data input.

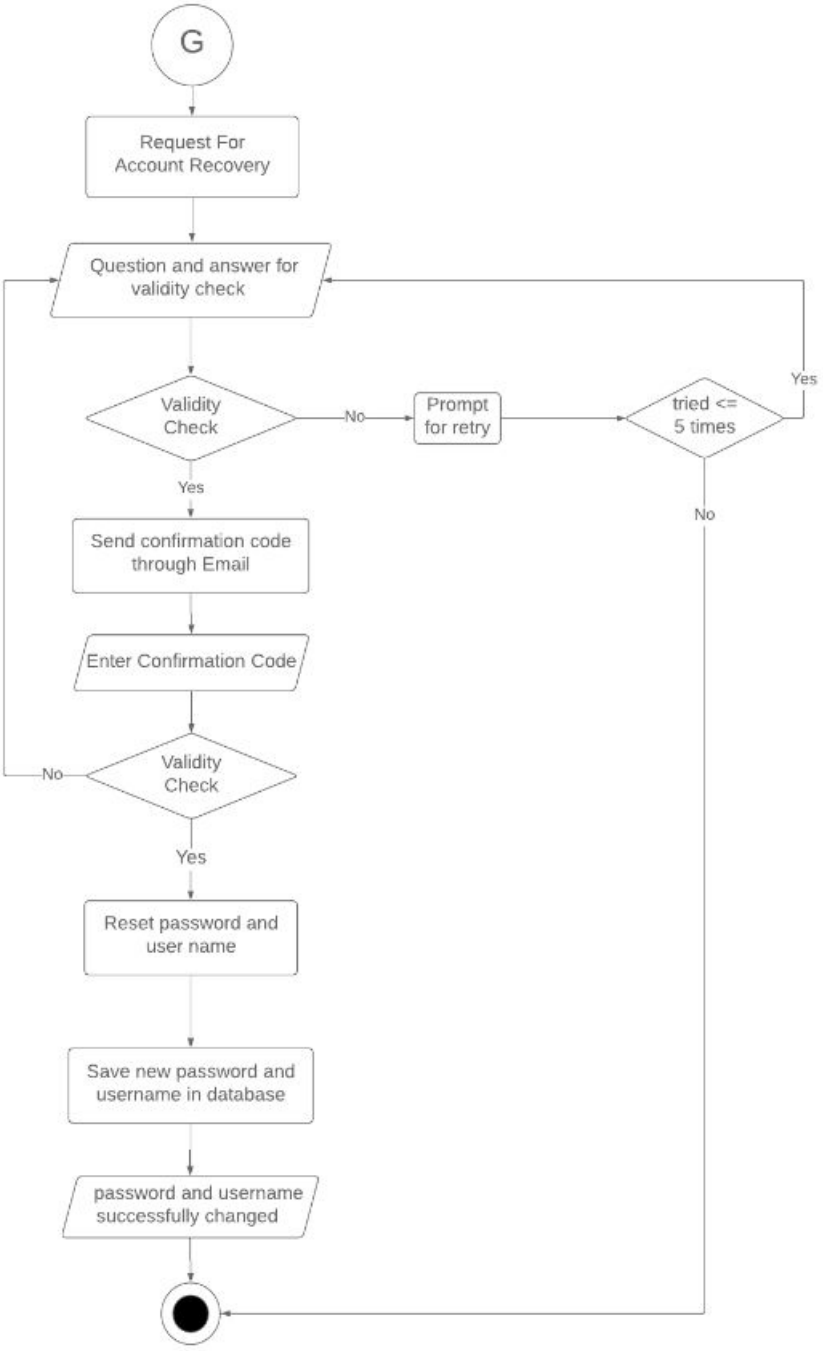


Figure 29: Activity Diagram 8: Account recovery.

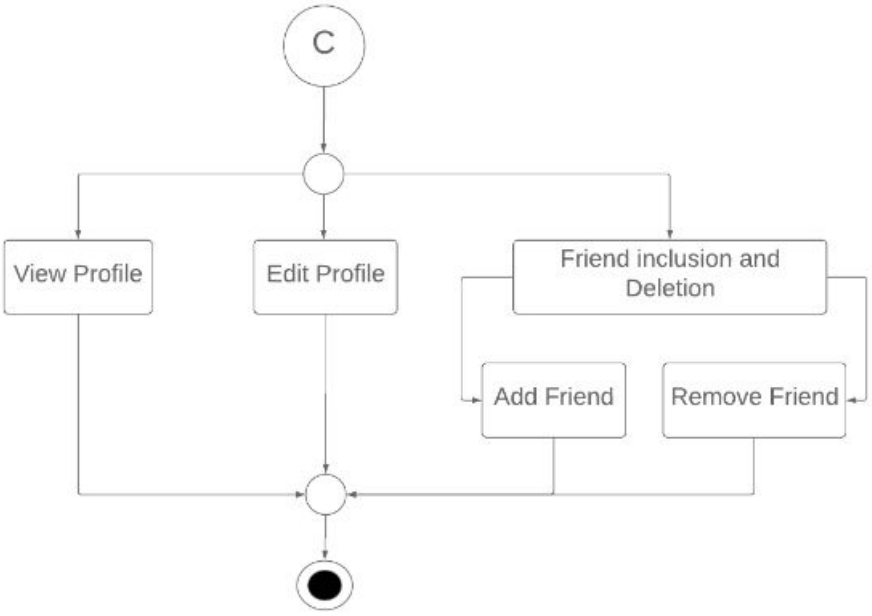


Figure 30: Activity Diagram 9: Profile management.

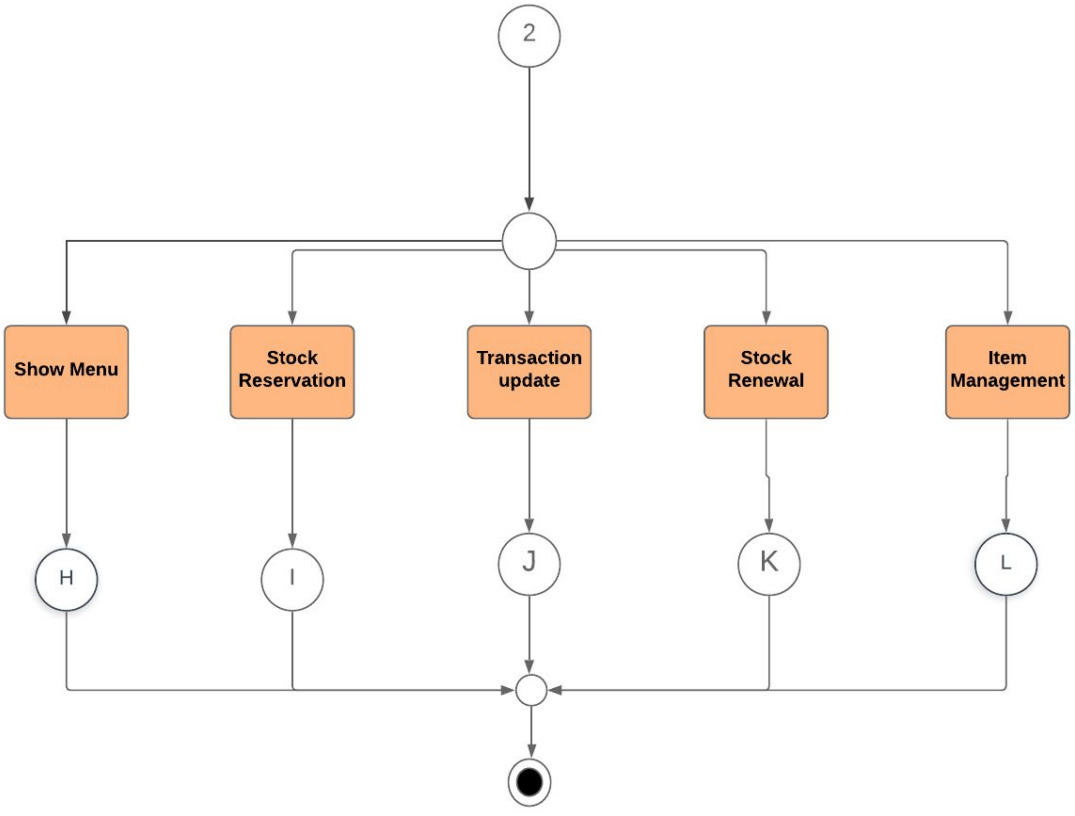


Figure 31: Activity Diagram 10: Stock management.

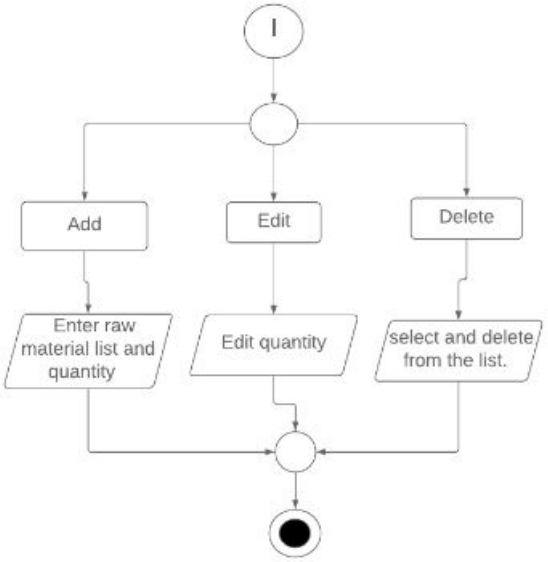


Figure 32: Activity Diagram 11: Stock reservation.

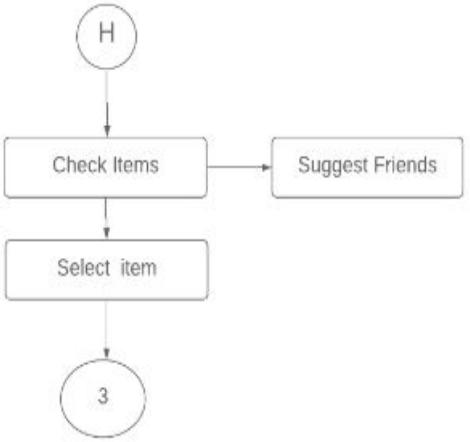


Figure 33: Activity Diagram 12: Show menu.

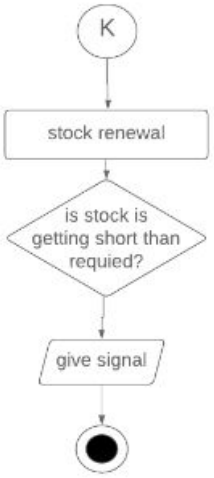


Figure 34: Activity Diagram 13: Stock renewal.

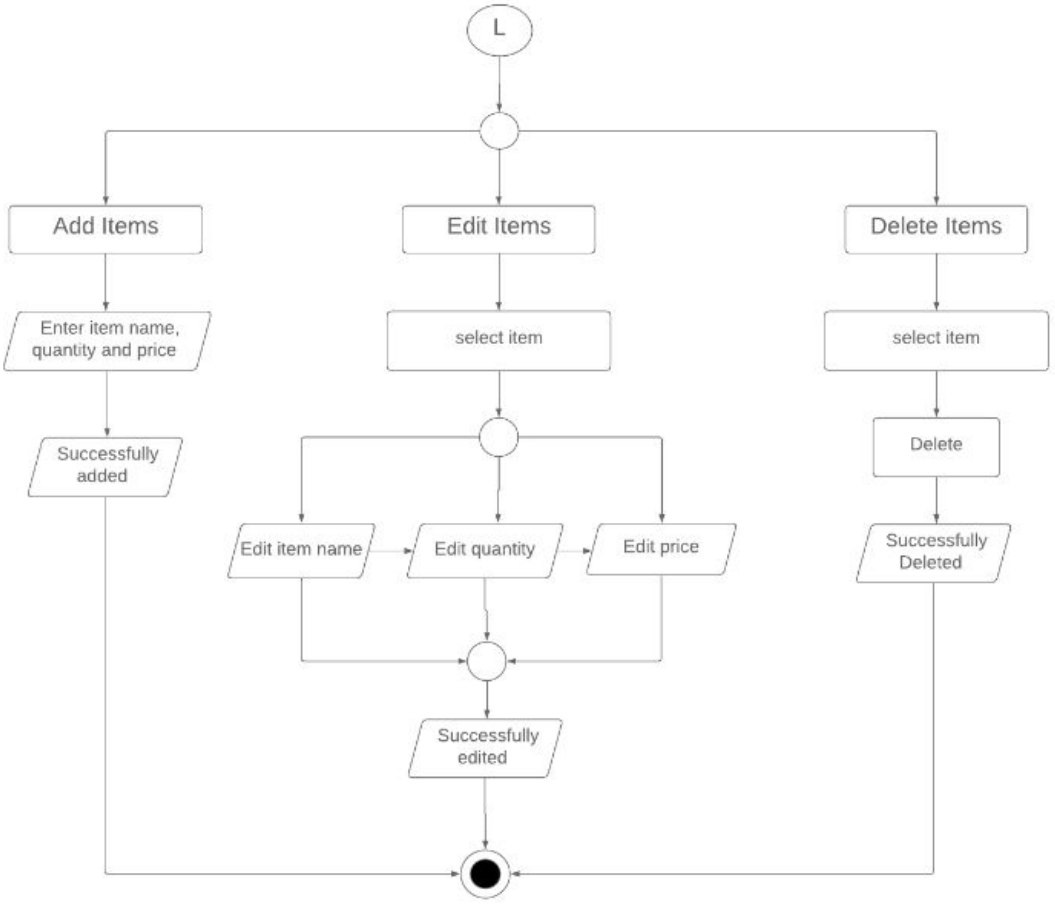


Figure 35: Activity Diagram 14: Item management.

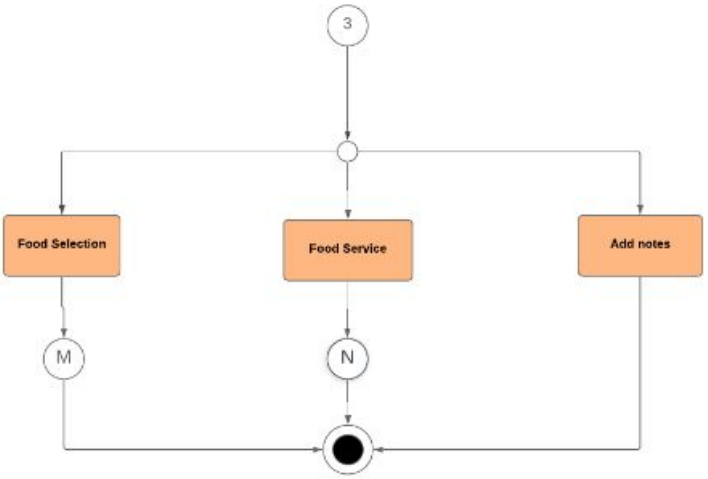


Figure 36: Activity Diagram 15: Order processing.

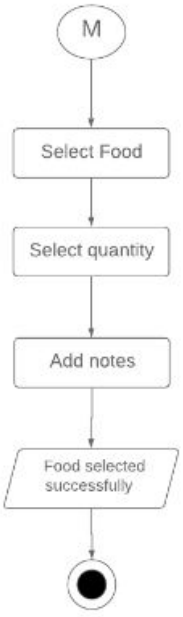


Figure 37: Activity Diagram 16: Food selection.

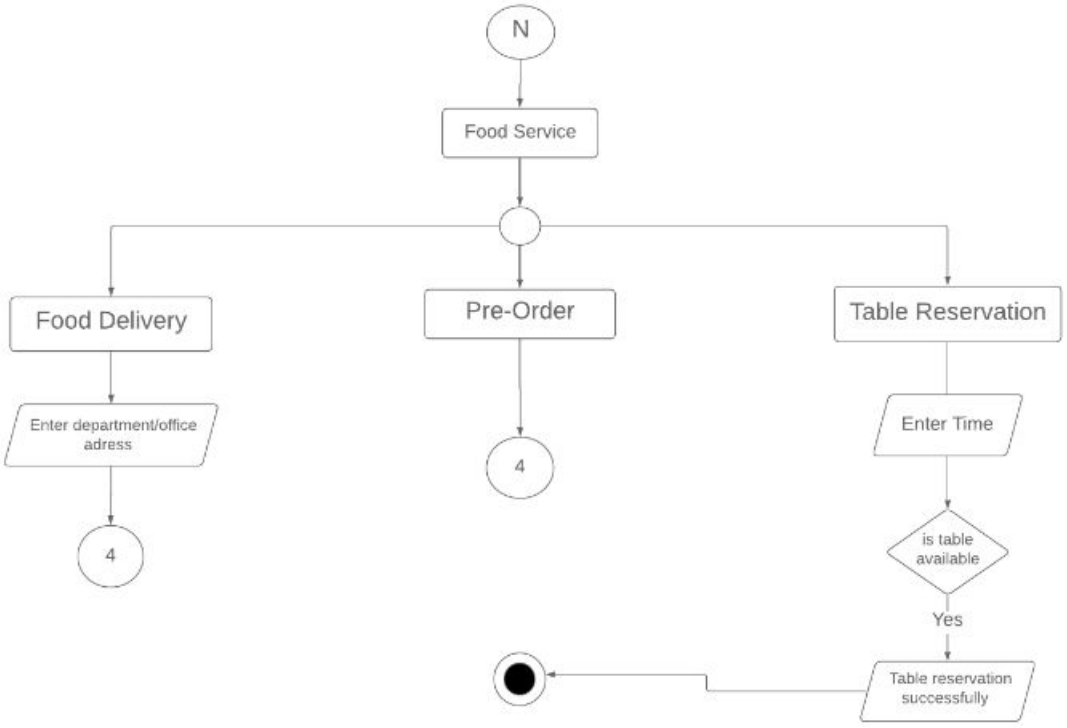


Figure 38: Activity Diagram 17: Food services.

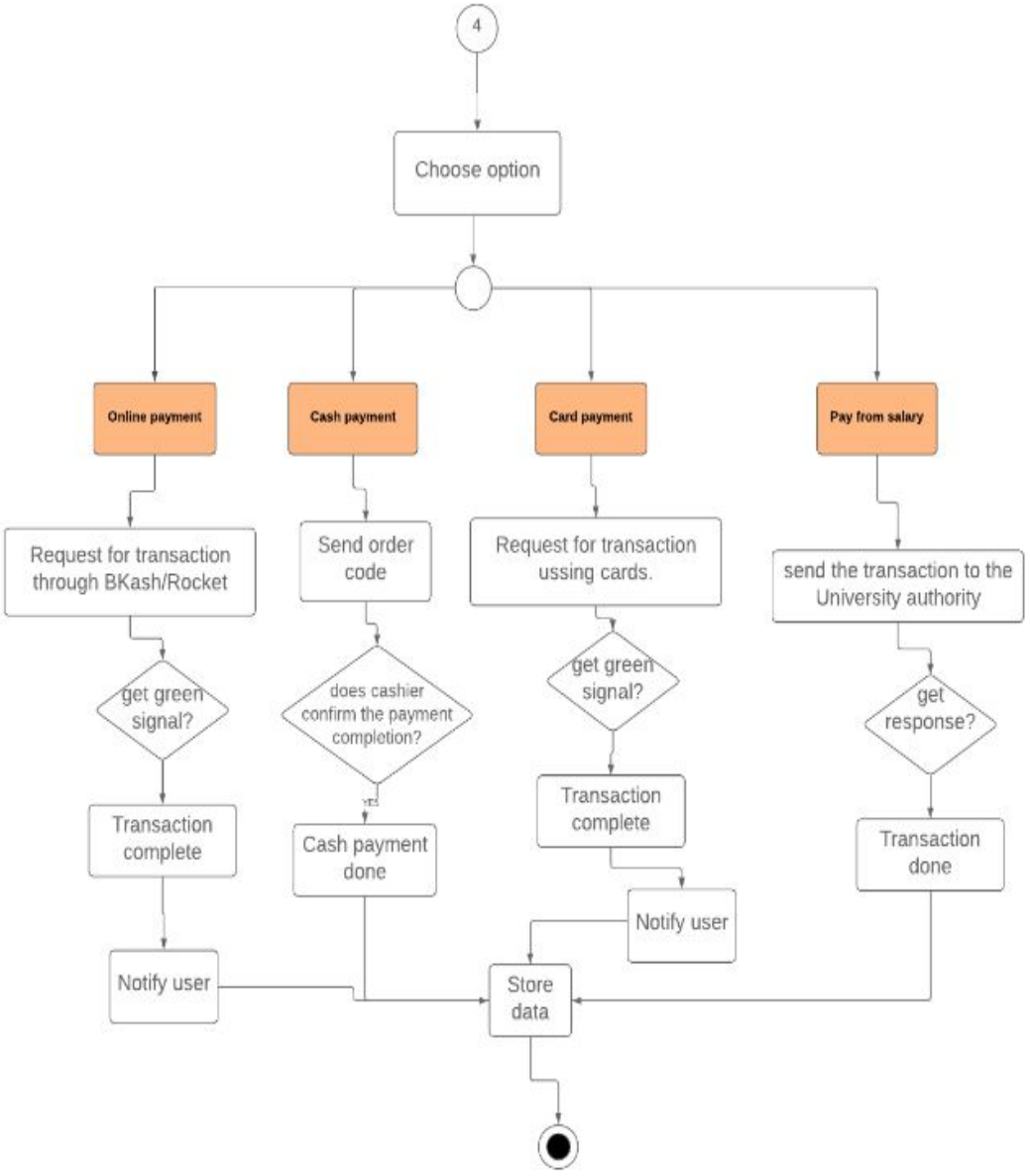


Figure 39: Activity Diagram 18: Payment.

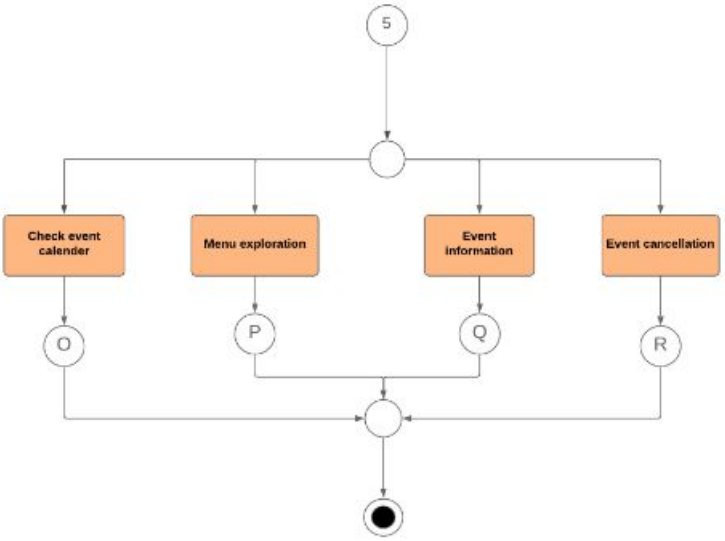


Figure 40: Activity Diagram 19: Event management.

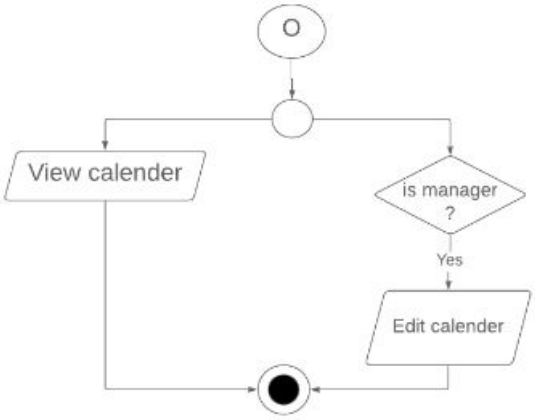


Figure 41: Activity Diagram 20: Check event calendar.

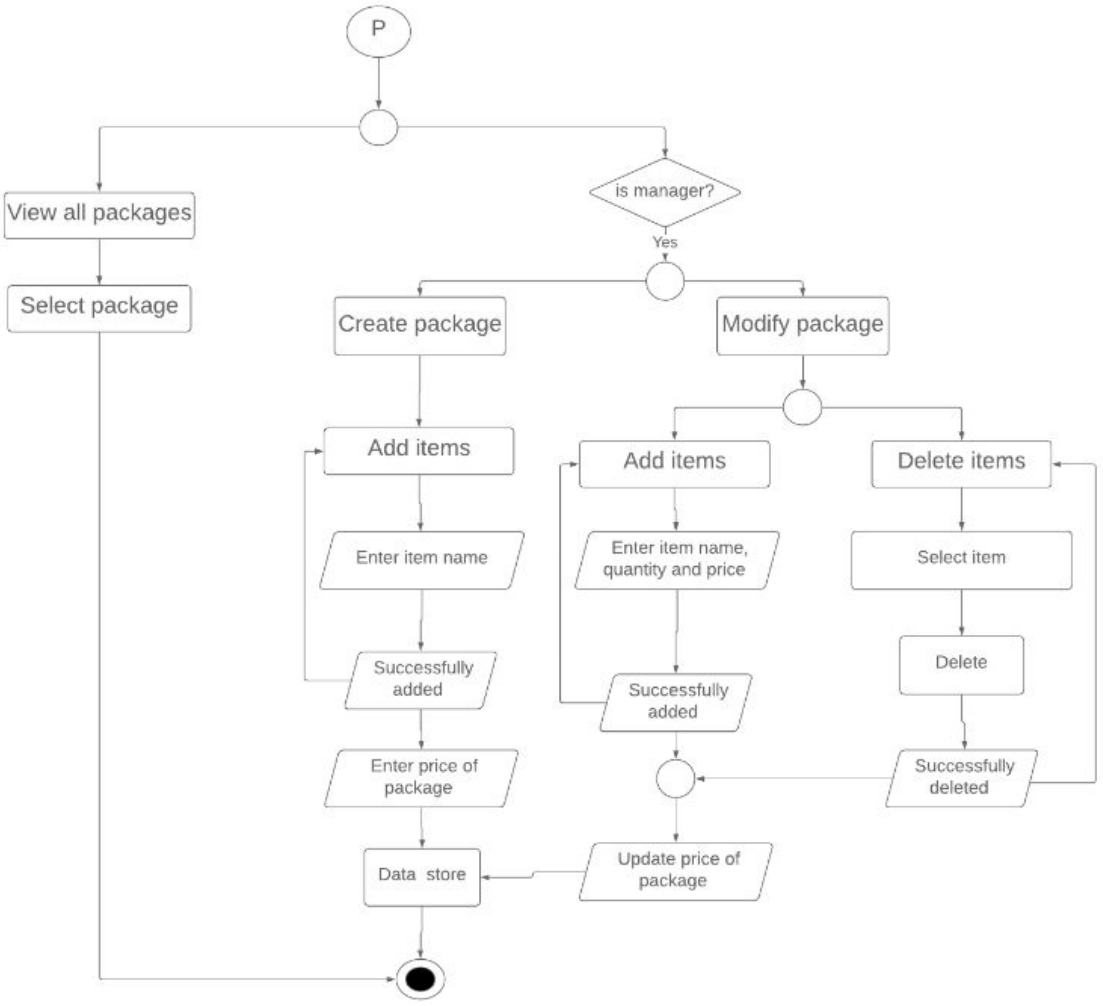


Figure 42: Activity Diagram 21: Menu exploration.

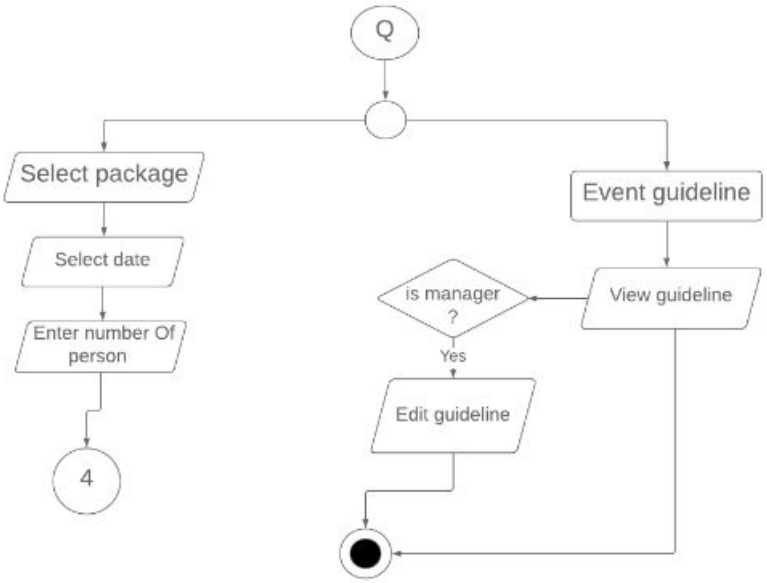


Figure 43: Activity Diagram 22: Event information.

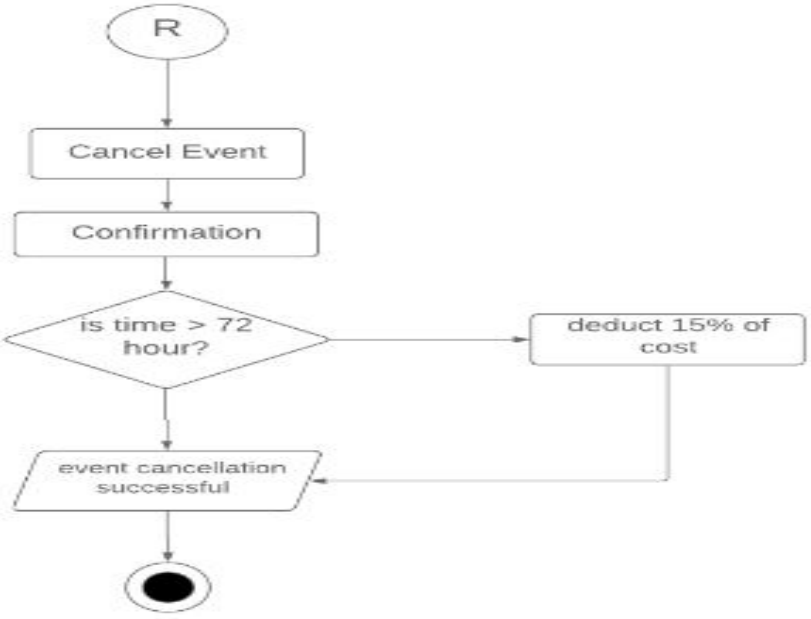


Figure 44: Activity Diagram 23: Event cancellation.

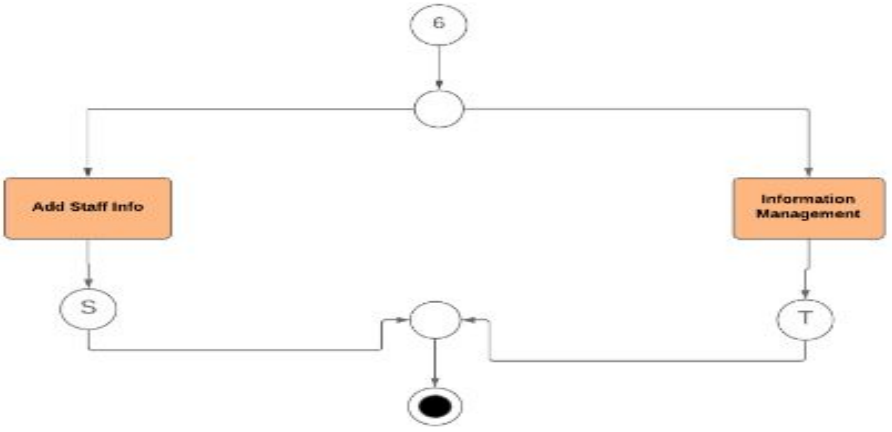


Figure 45: Activity Diagram 24: Staff management.

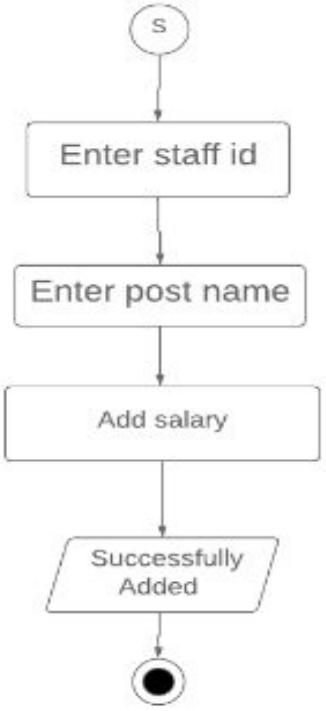


Figure 46: Activity Diagram 25: Add staff information.

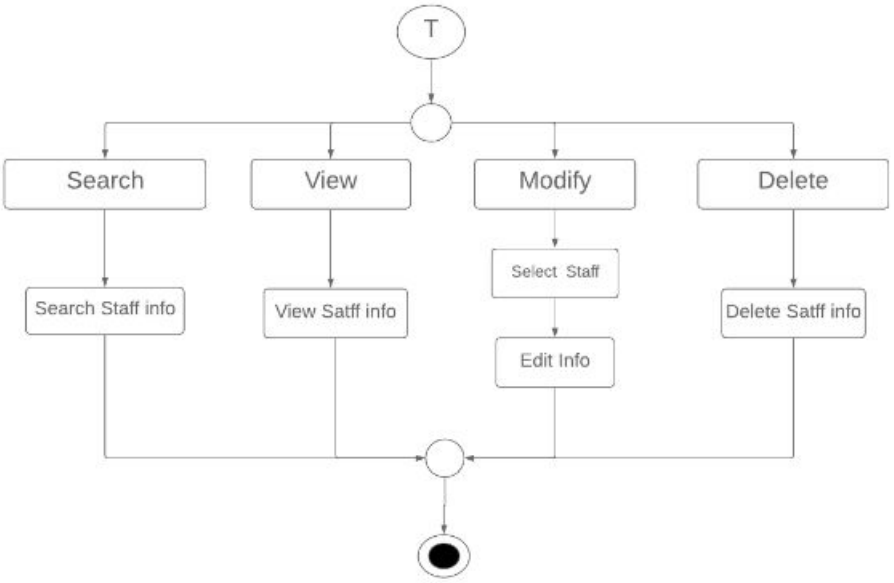


Figure 47: Activity Diagram 26: Information management.

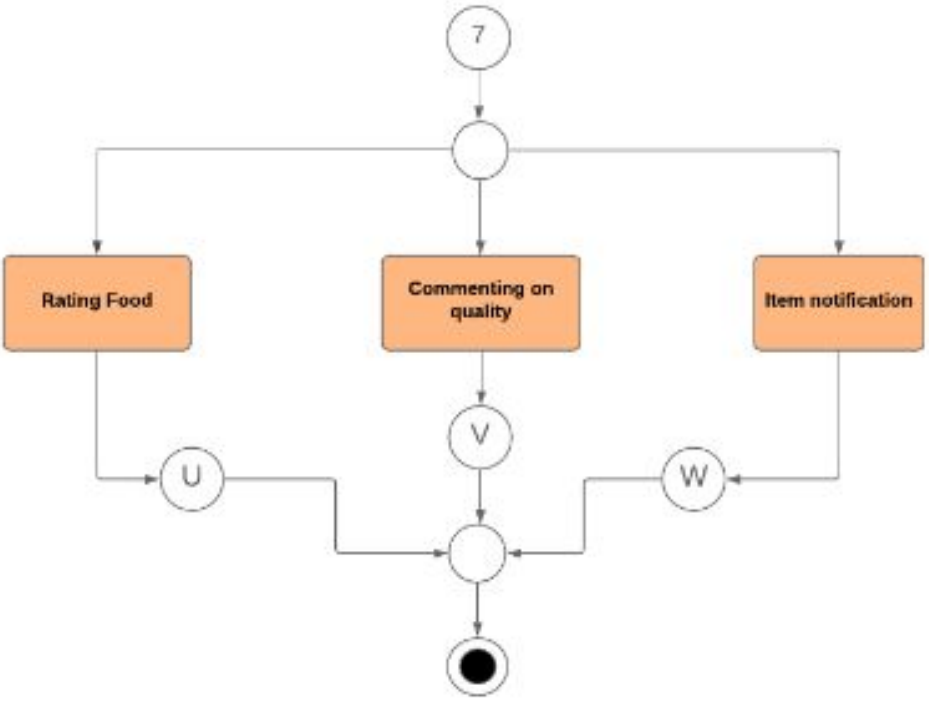


Figure 48: Activity Diagram 27: Feedback and special item.

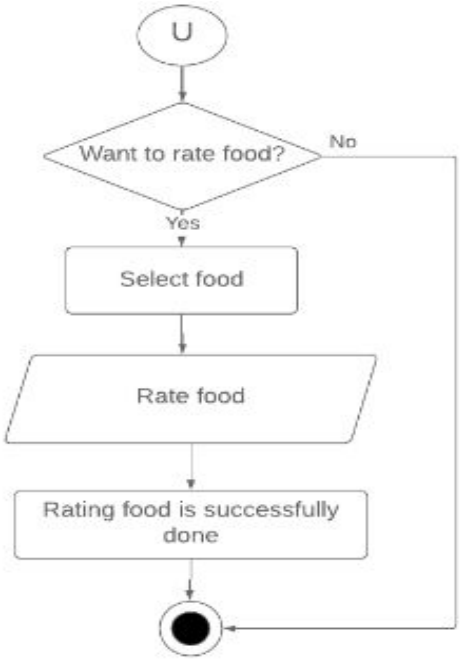


Figure 49: Activity Diagram 28: Rating food.

Figure 50: Activity Diagram 29: Commenting on quality.

Figure 51: Activity Diagram 30: Item notification.

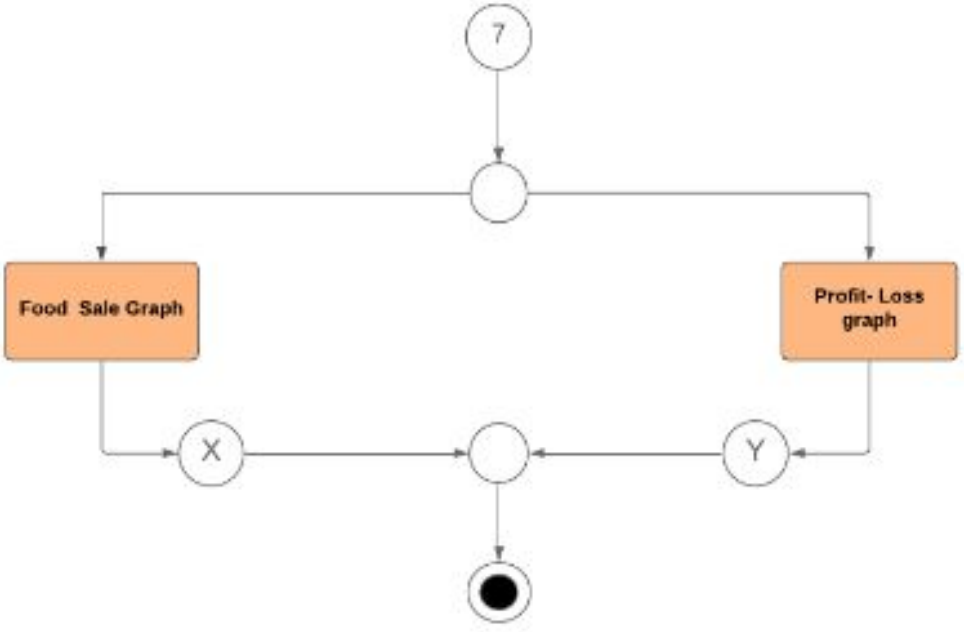


Figure 52: Activity Diagram 31: Graph generation.

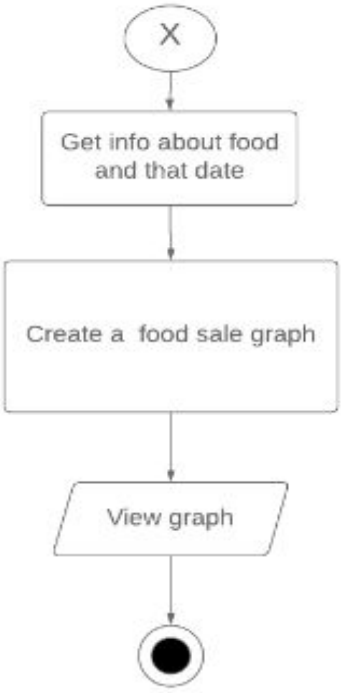


Figure 53: Activity Diagram 32: Food sale graph.

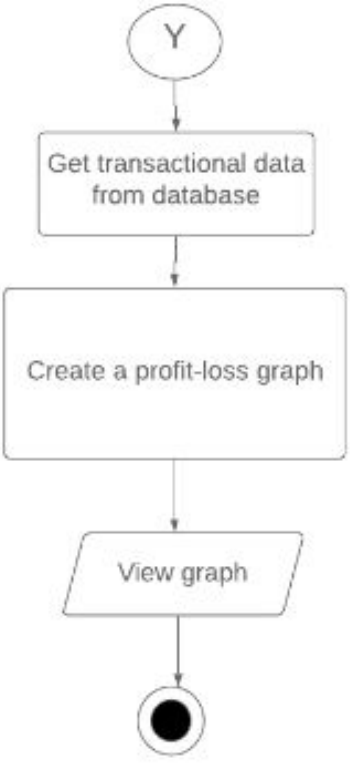


Figure 54: Activity Diagram 33: Profit-loss graph.

4.5 Swim lane Diagram

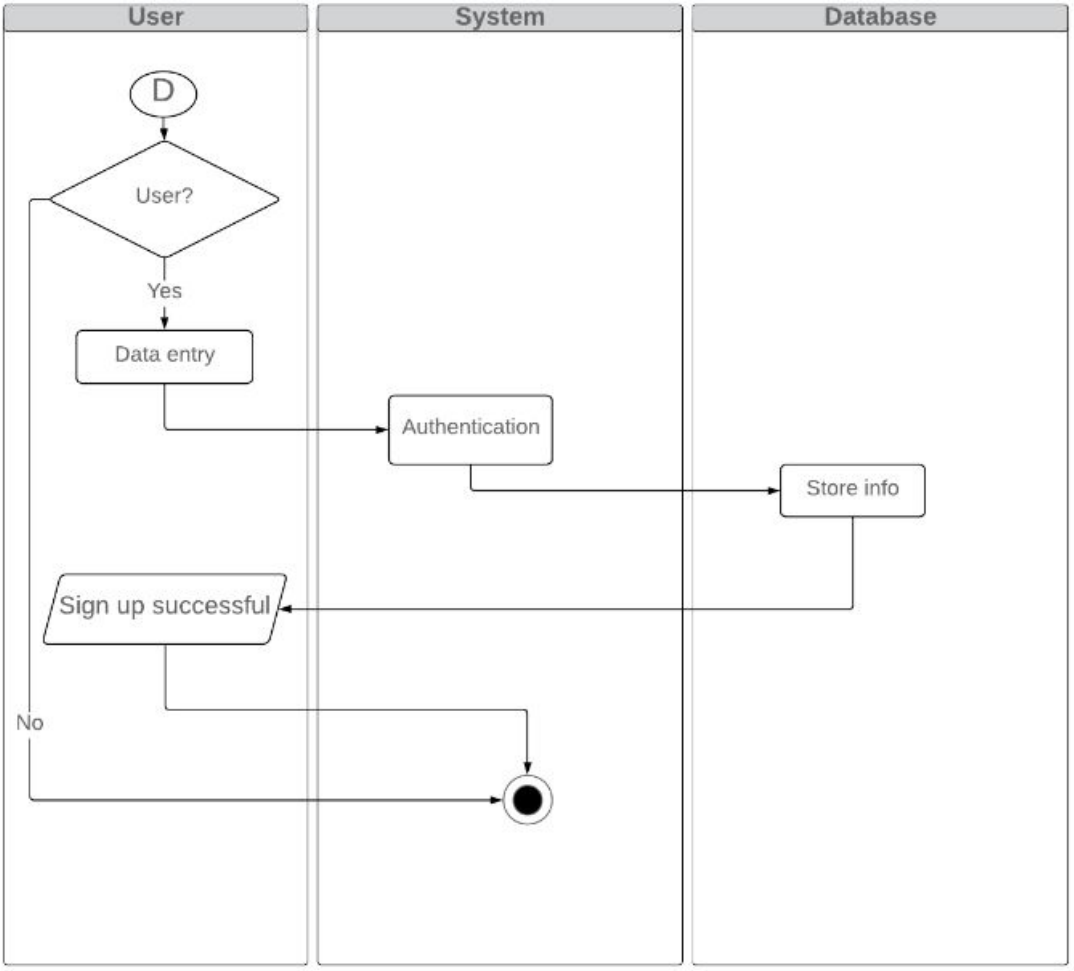


Figure 55: Swim Lane Diagram User account

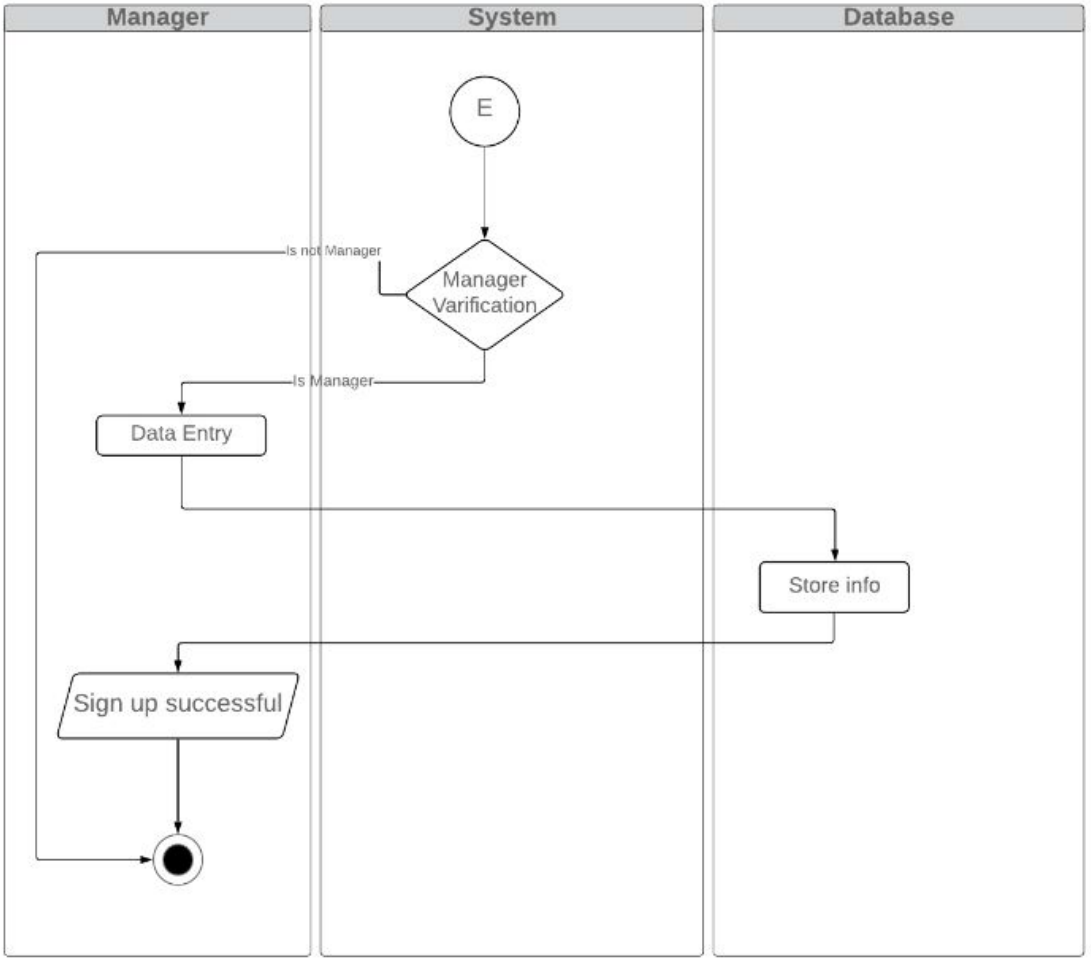


Figure 56: Swim Lane Diagram Manager account.

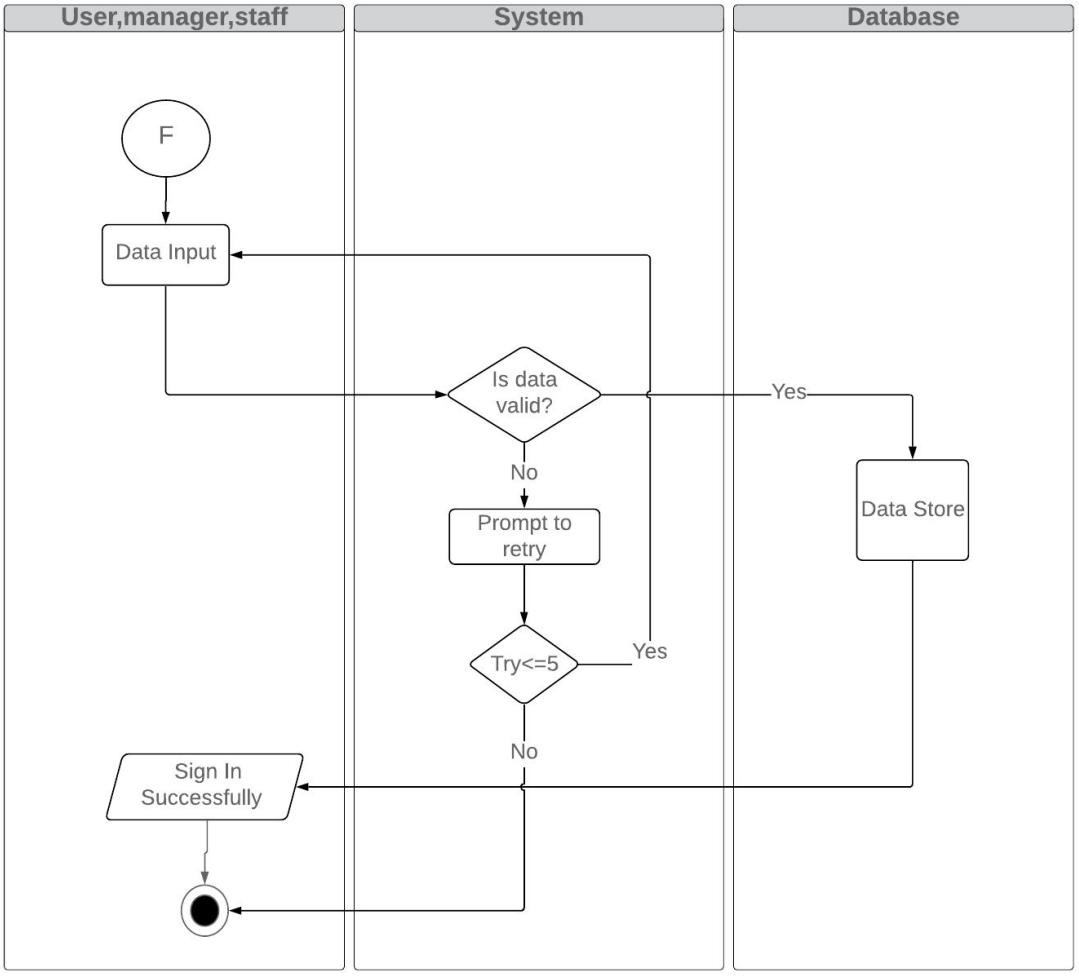


Figure 57: Swim Lane Diagram: Log in.

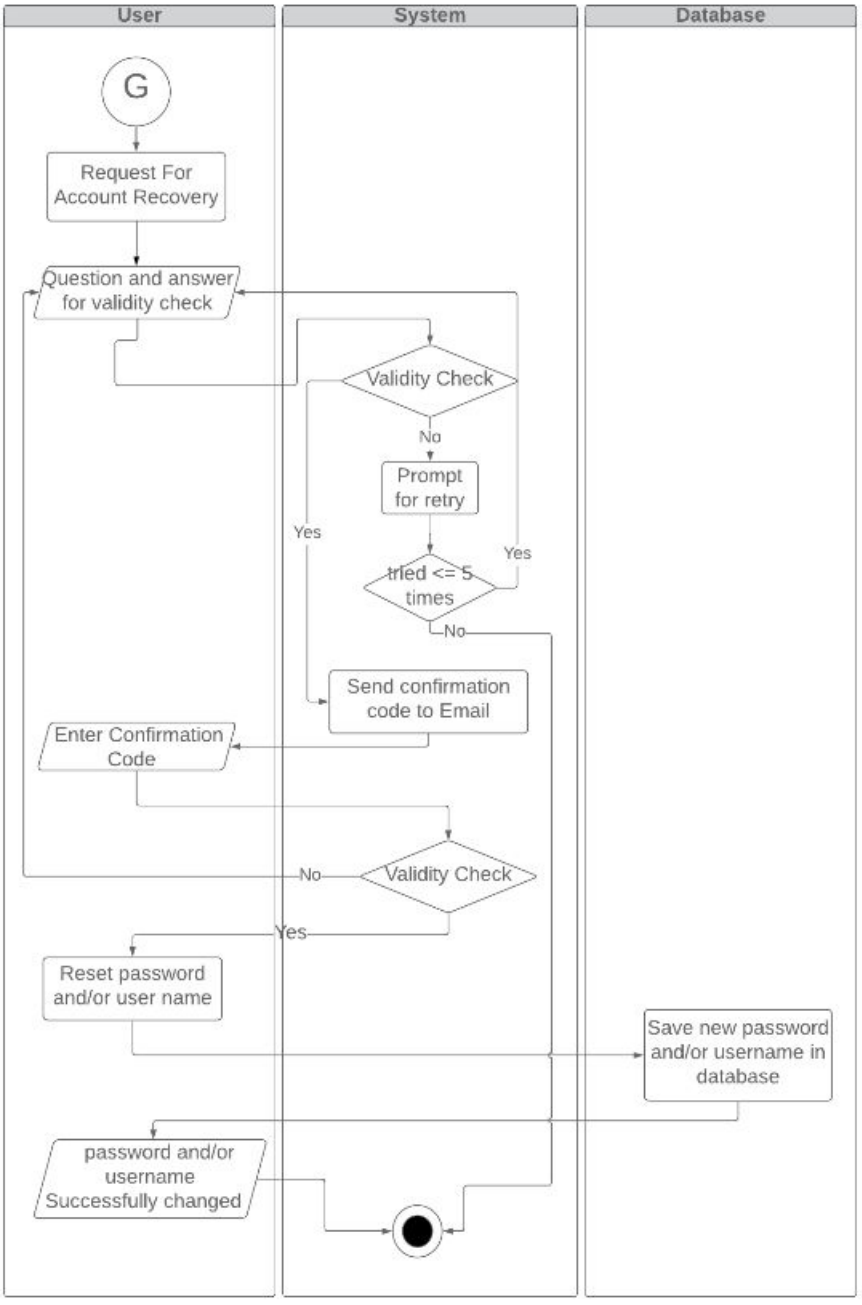


Figure 58: Swim Lane Diagram: Account recovery.

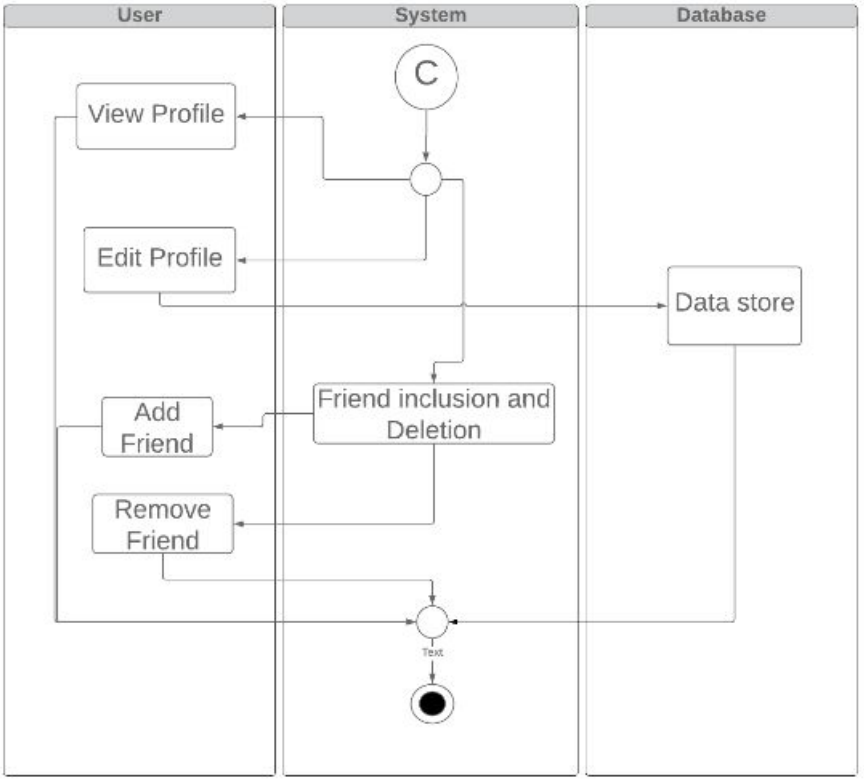


Figure 59: Swim Lane Diagram: Profile management

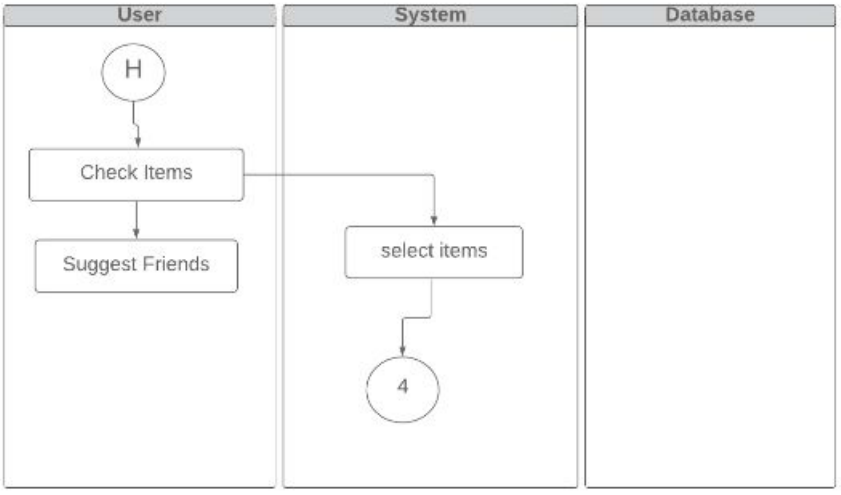


Figure 60: Swim Lane Diagram: Show menu

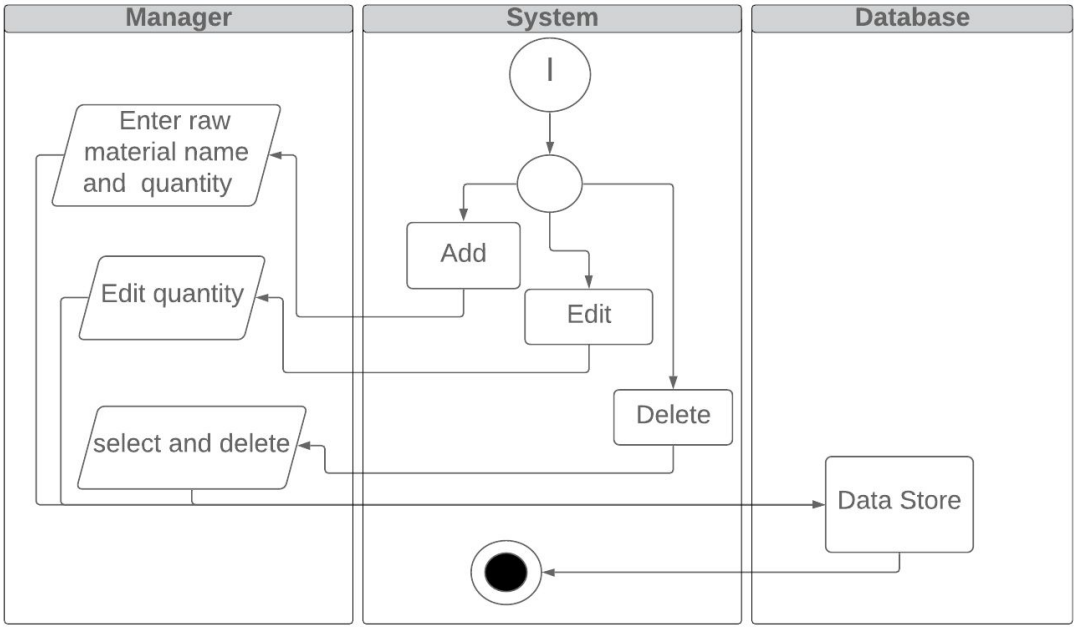


Figure 61: Swim Lane Diagram: Stock reservation

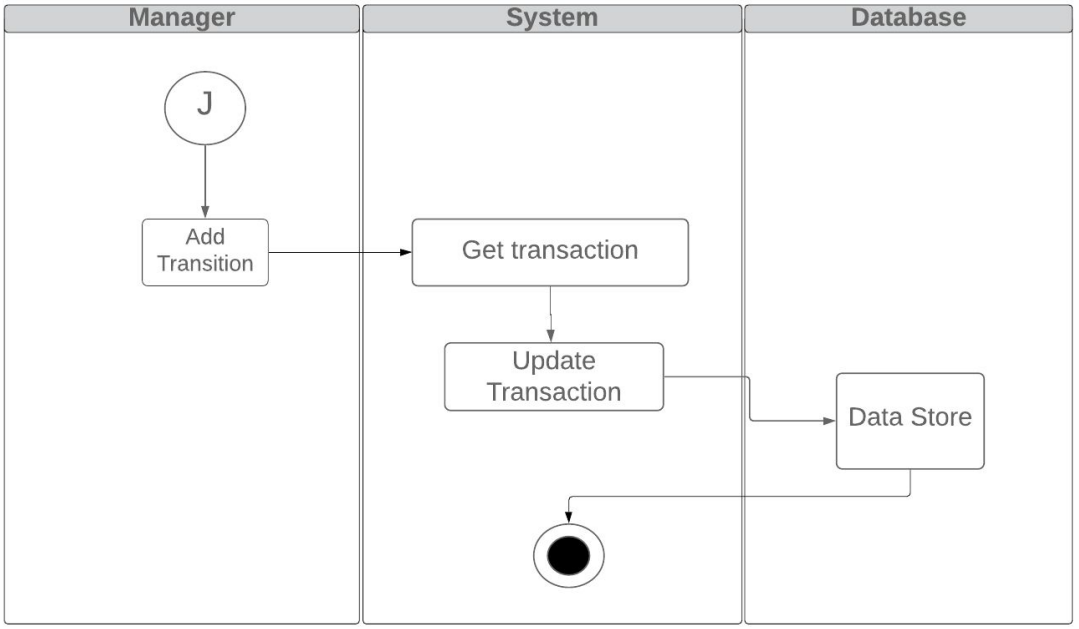


Figure 62: Swim Lane Diagram: Transaction update

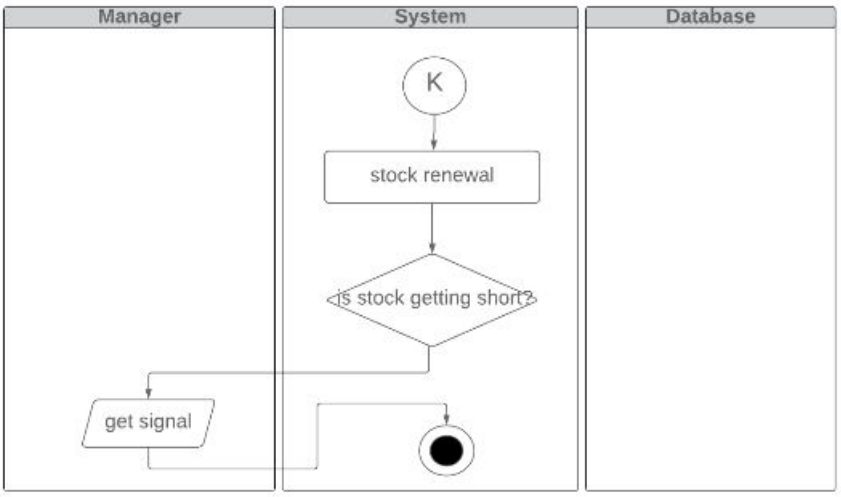


Figure 63: Swim Lane Diagram: Stock renewal

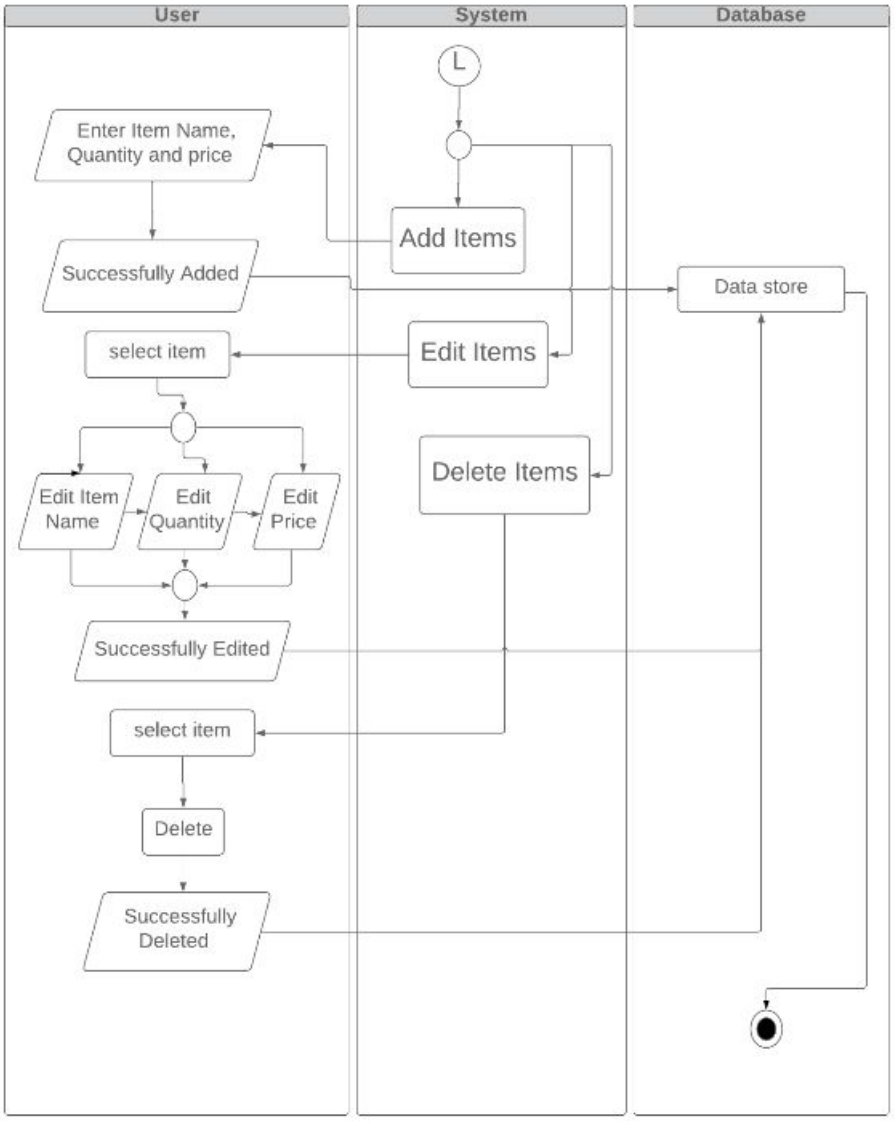


Figure 64: Swim Lane Diagram: Item management.

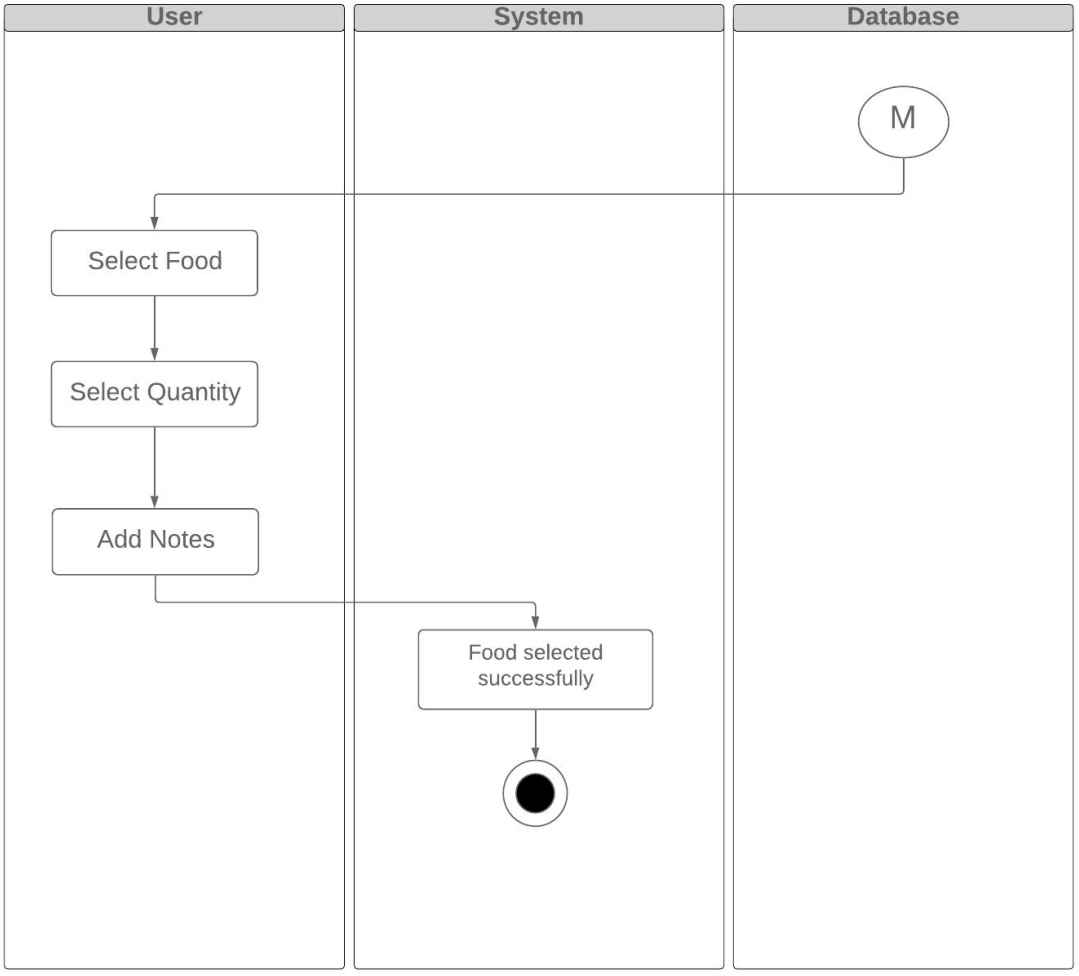


Figure 65: Swim Lane Diagram: Food selection.

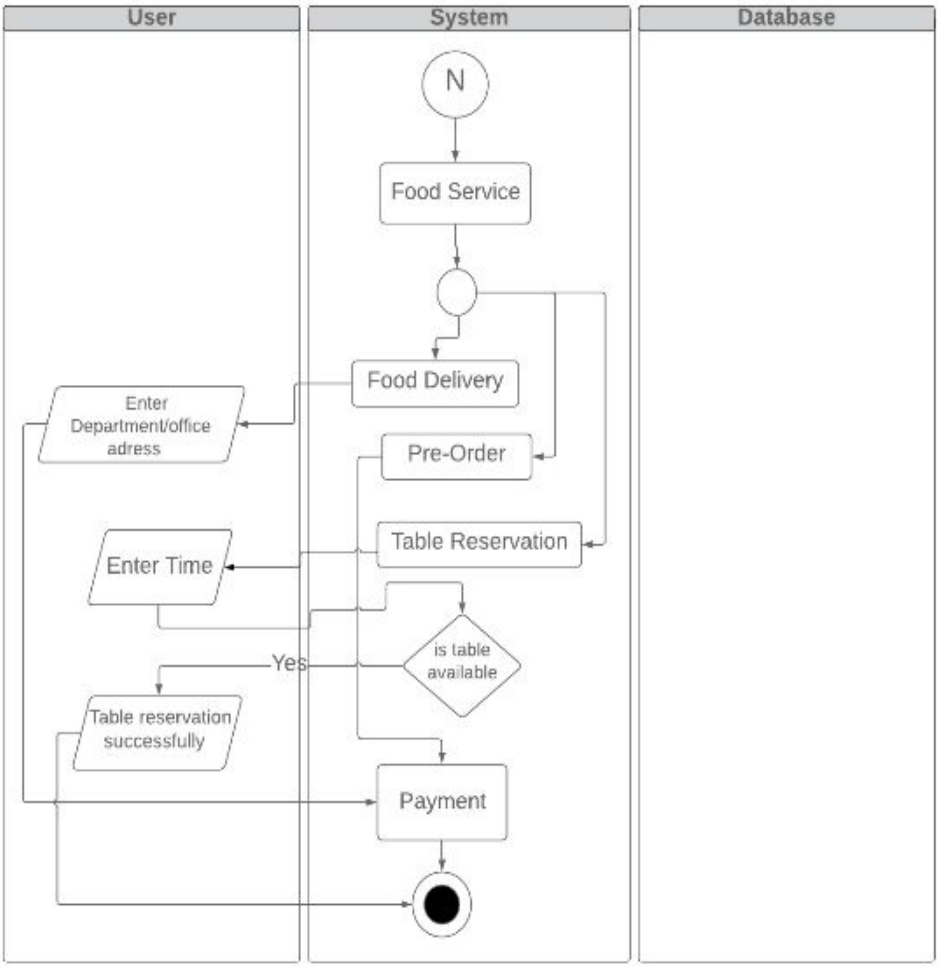


Figure 66: Swim Lane Diagram: Food services.

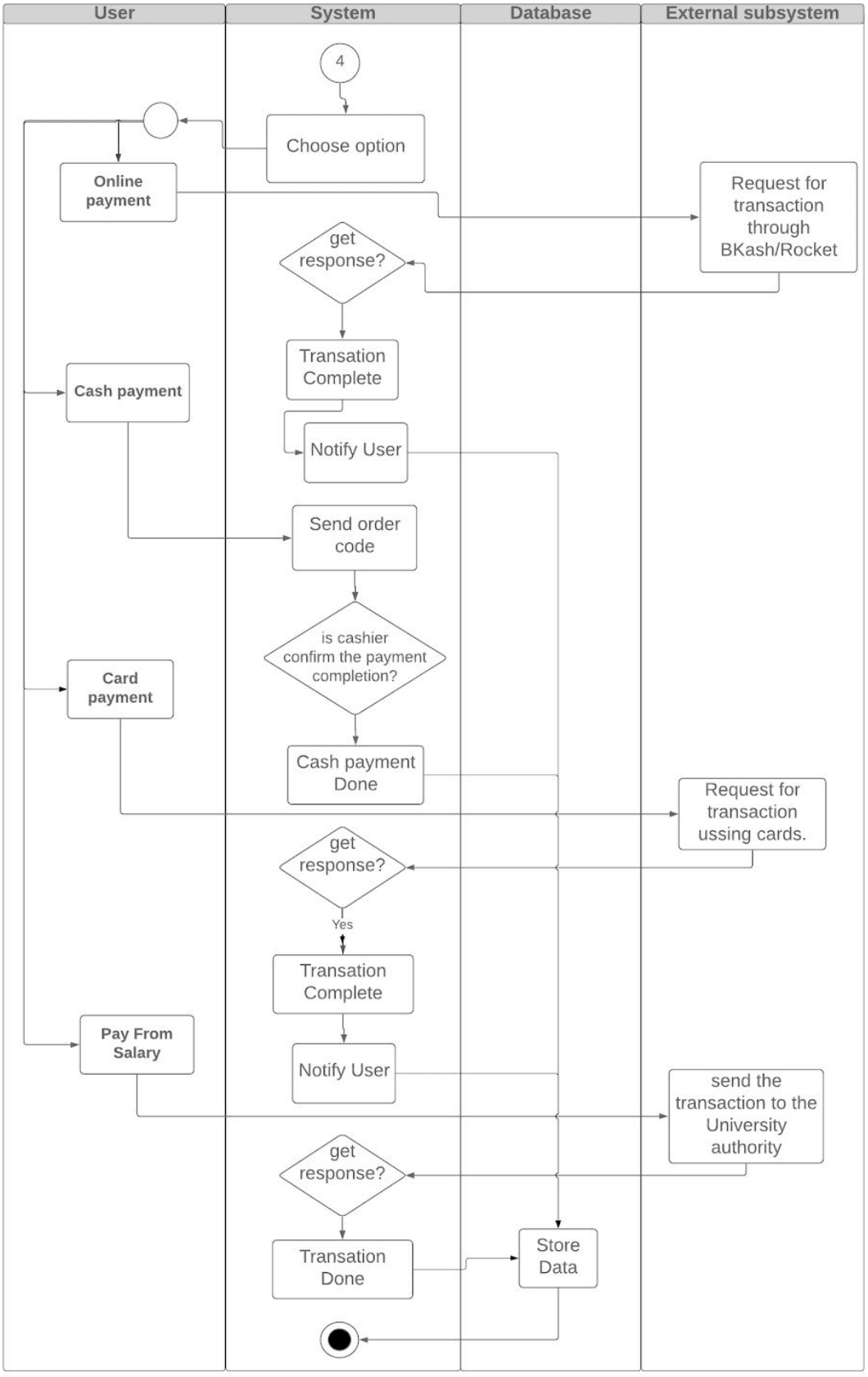


Figure 67: Swim Lane Diagram: Payment.

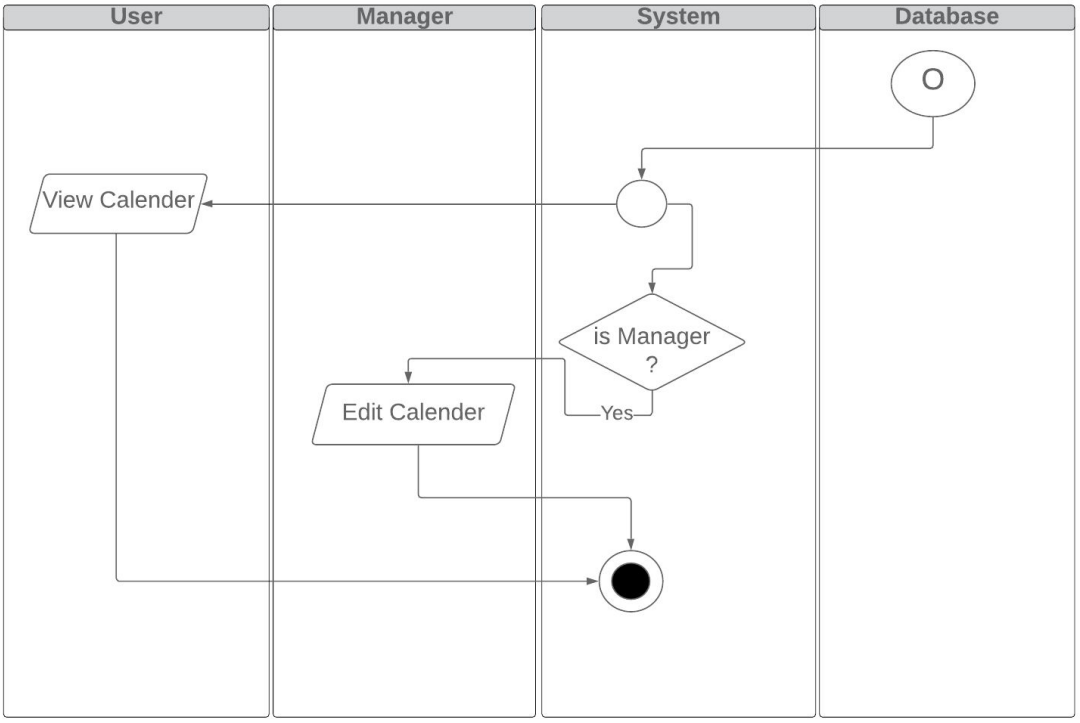


Figure 68: Swim Lane Diagram: Check event calendar

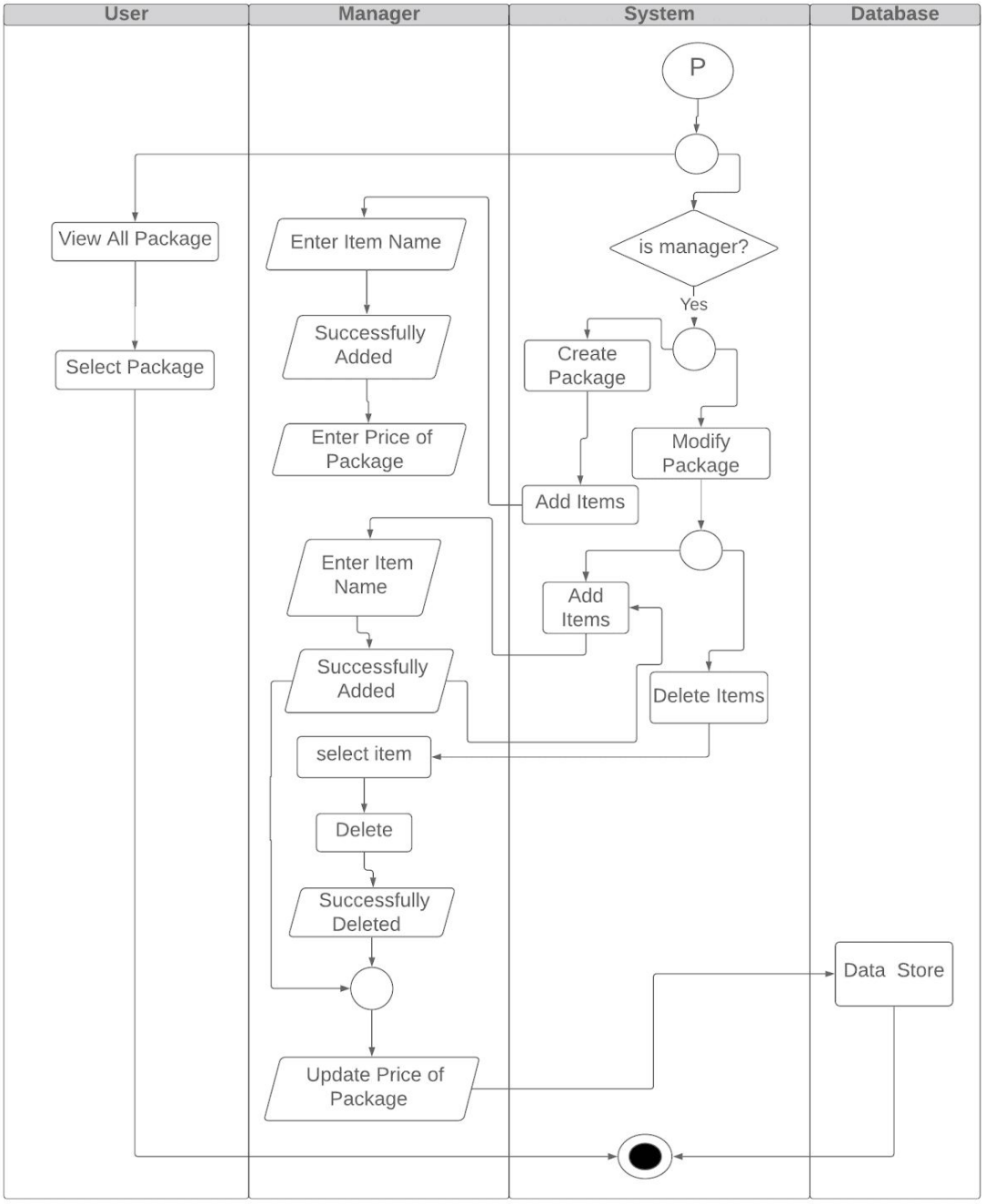


Figure 69: Swim Lane Diagram: MENU EXPLORATION

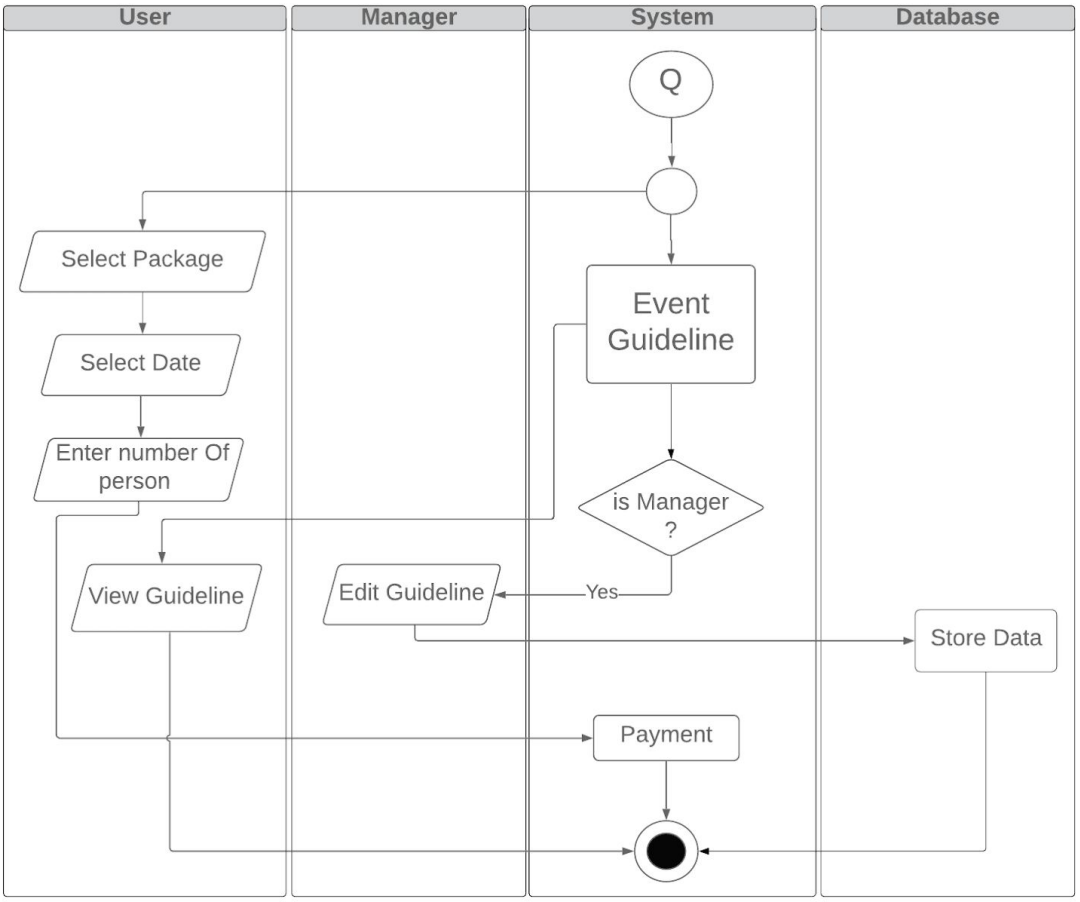


Figure 71: Swim Lane Diagram: Event information.

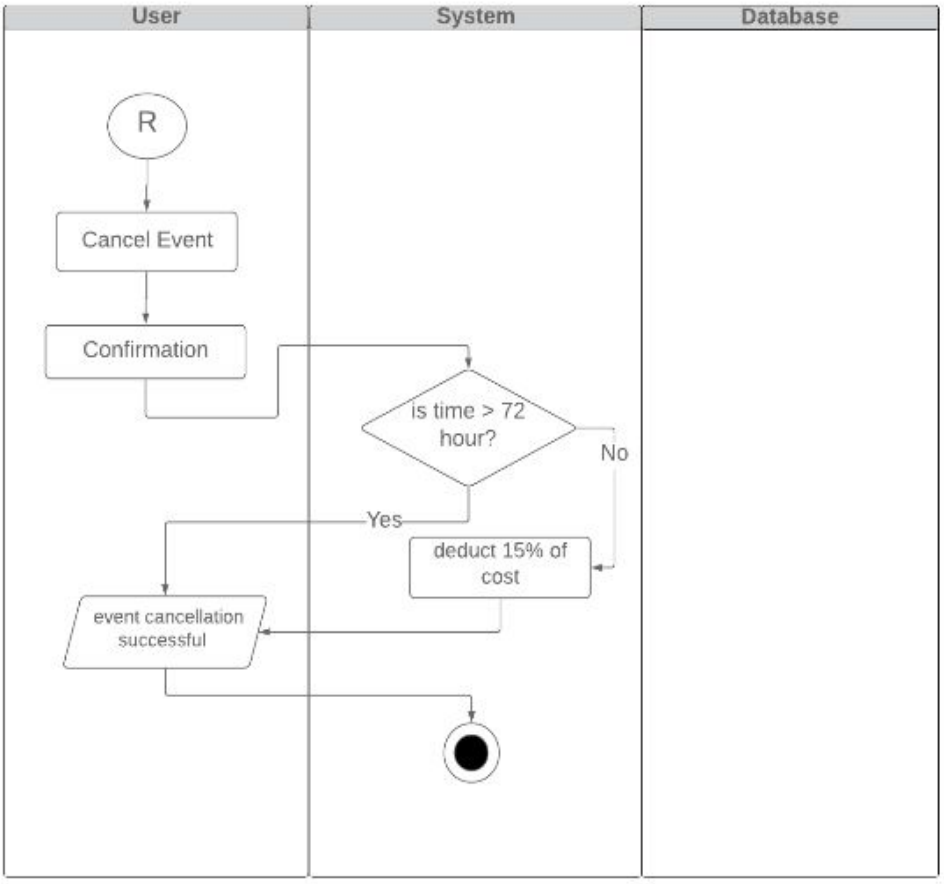


Figure 72: Swim Lane Diagram: Event cancellation

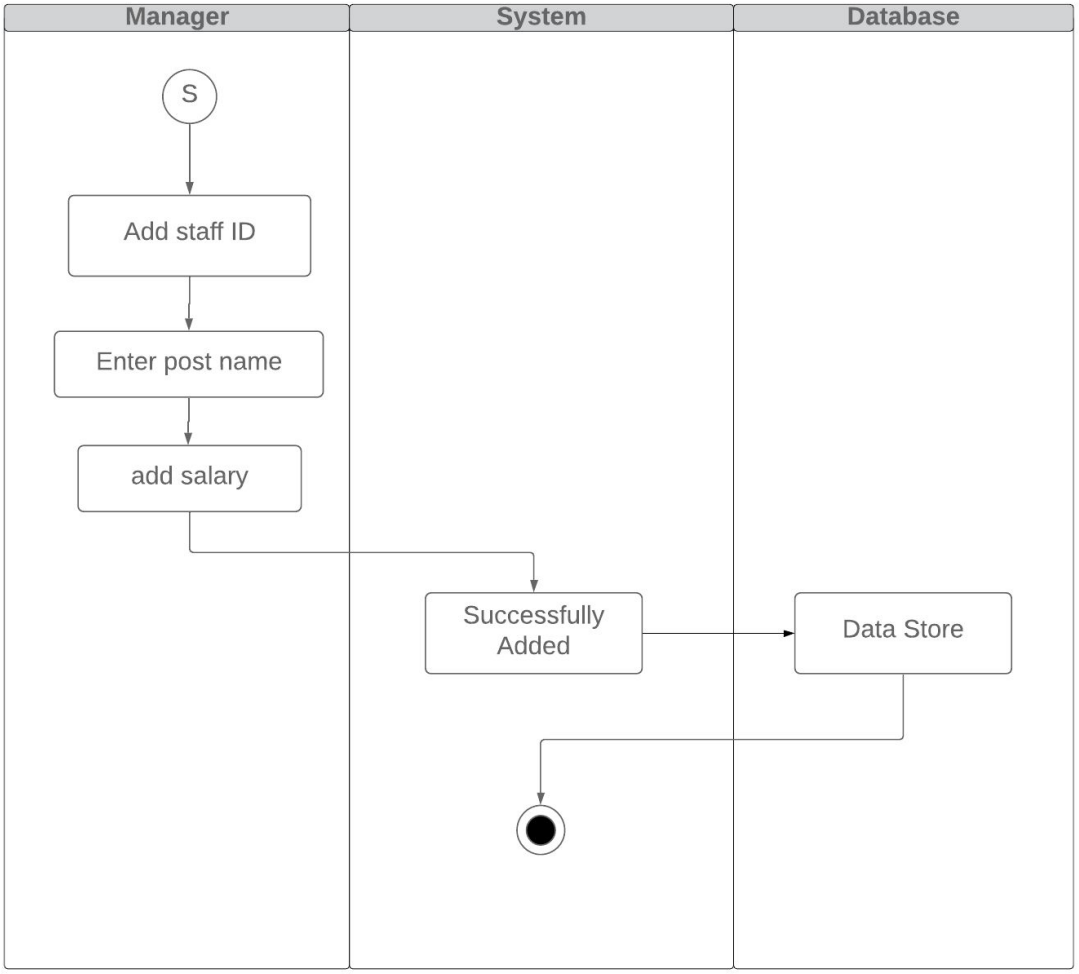


Figure 73: Swim Lane Diagram: Add staff information.

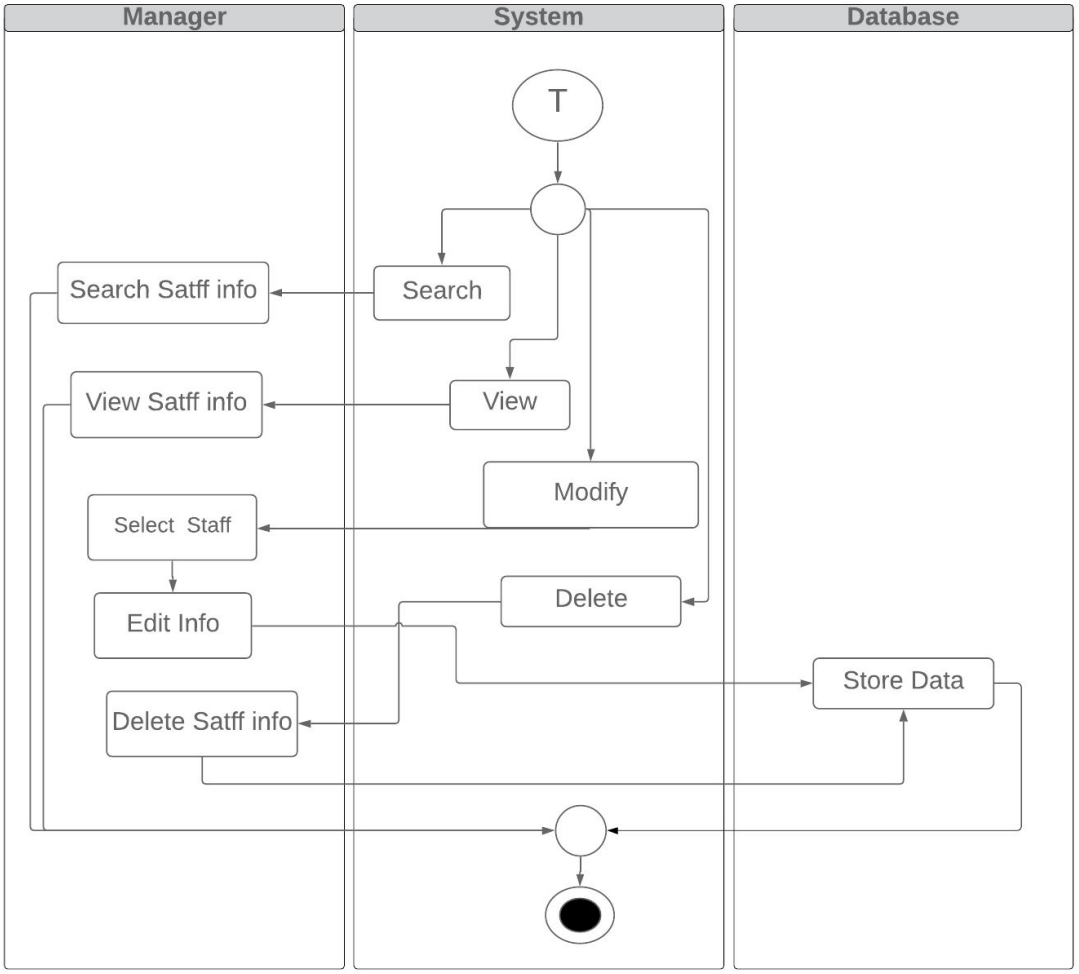


Figure 74: Swim Lane Diagram: Information management.

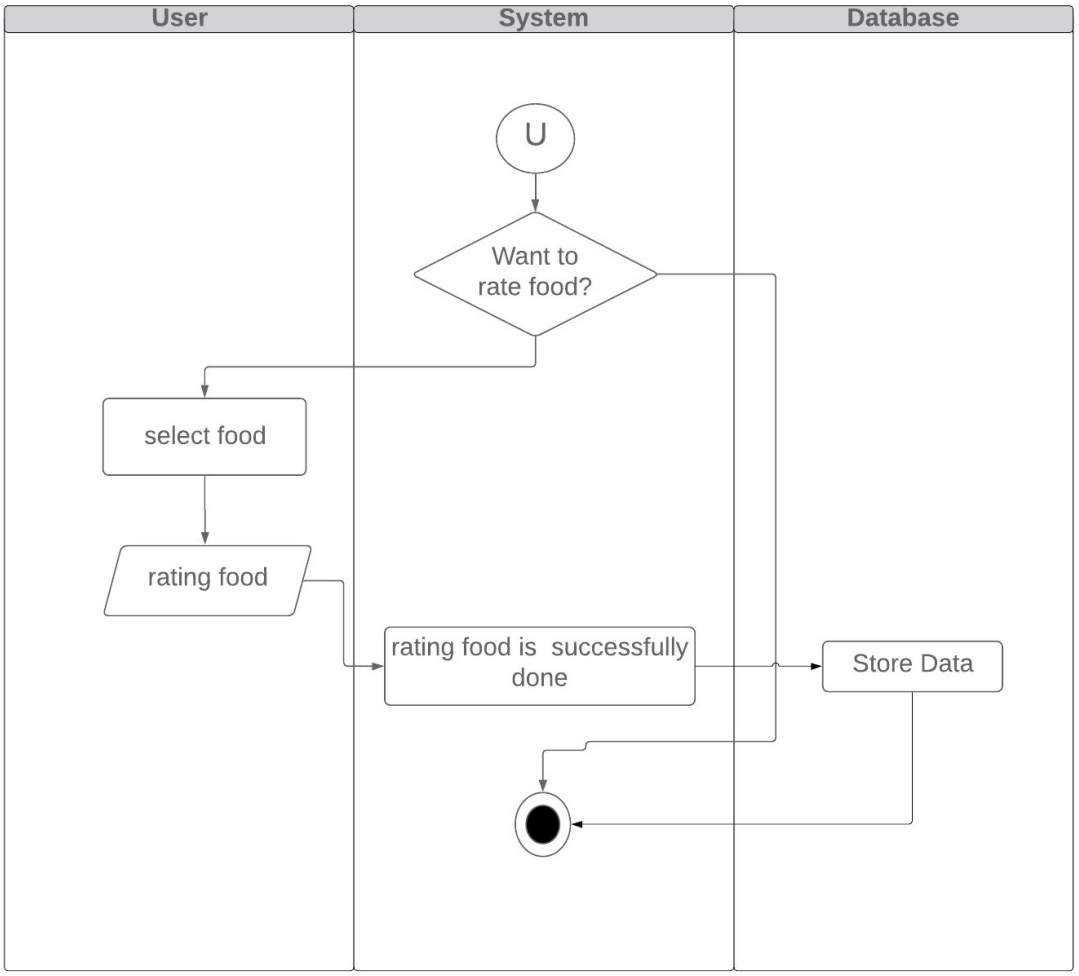


Figure 75: Swim Lane Diagram: Rating food

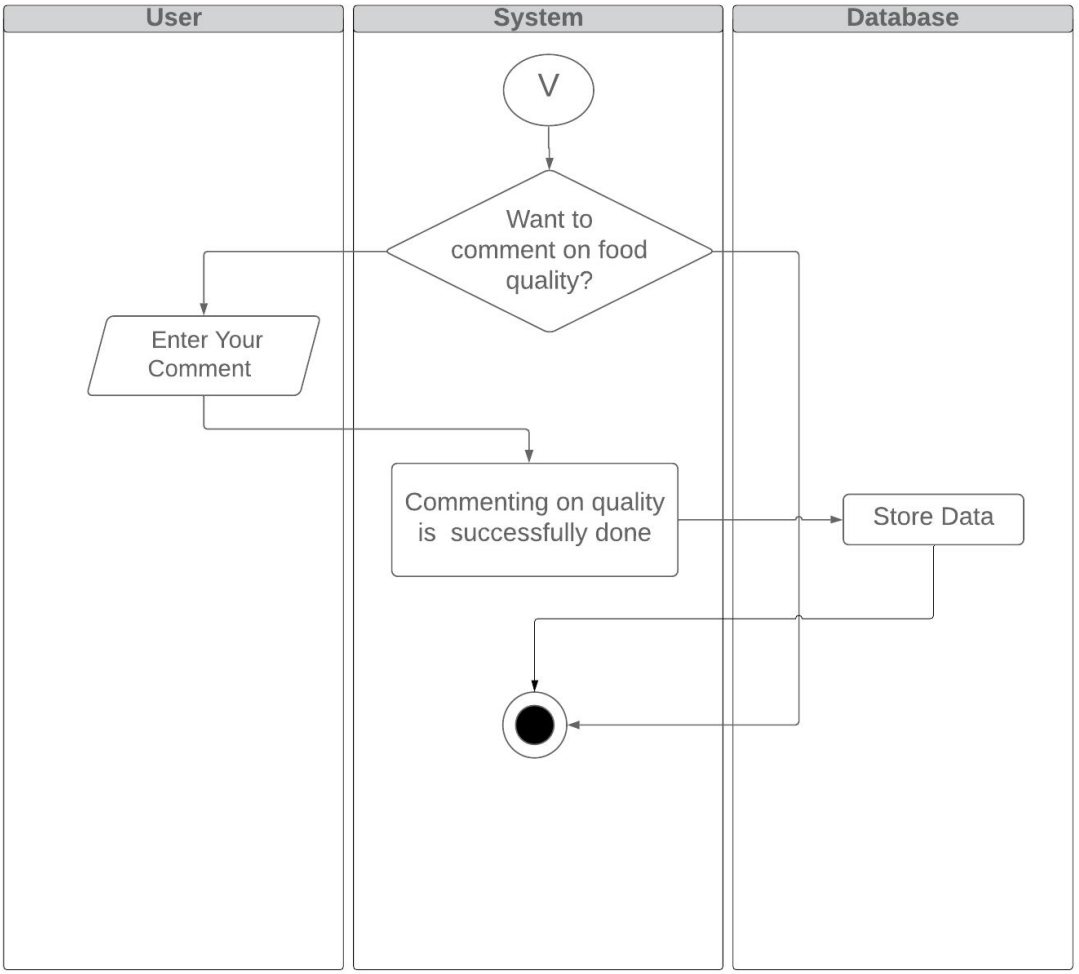


Figure 76: Swim Lane Diagram: Commenting on quality.

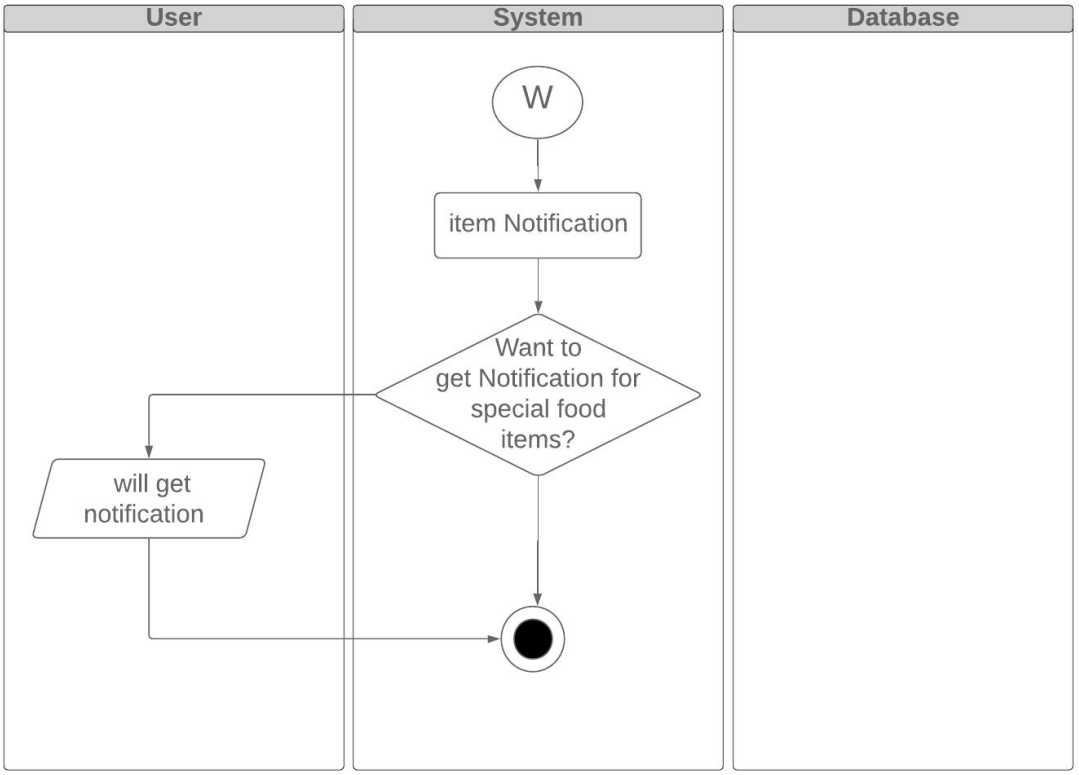


Figure 77: Swim Lane Diagram: Item notification

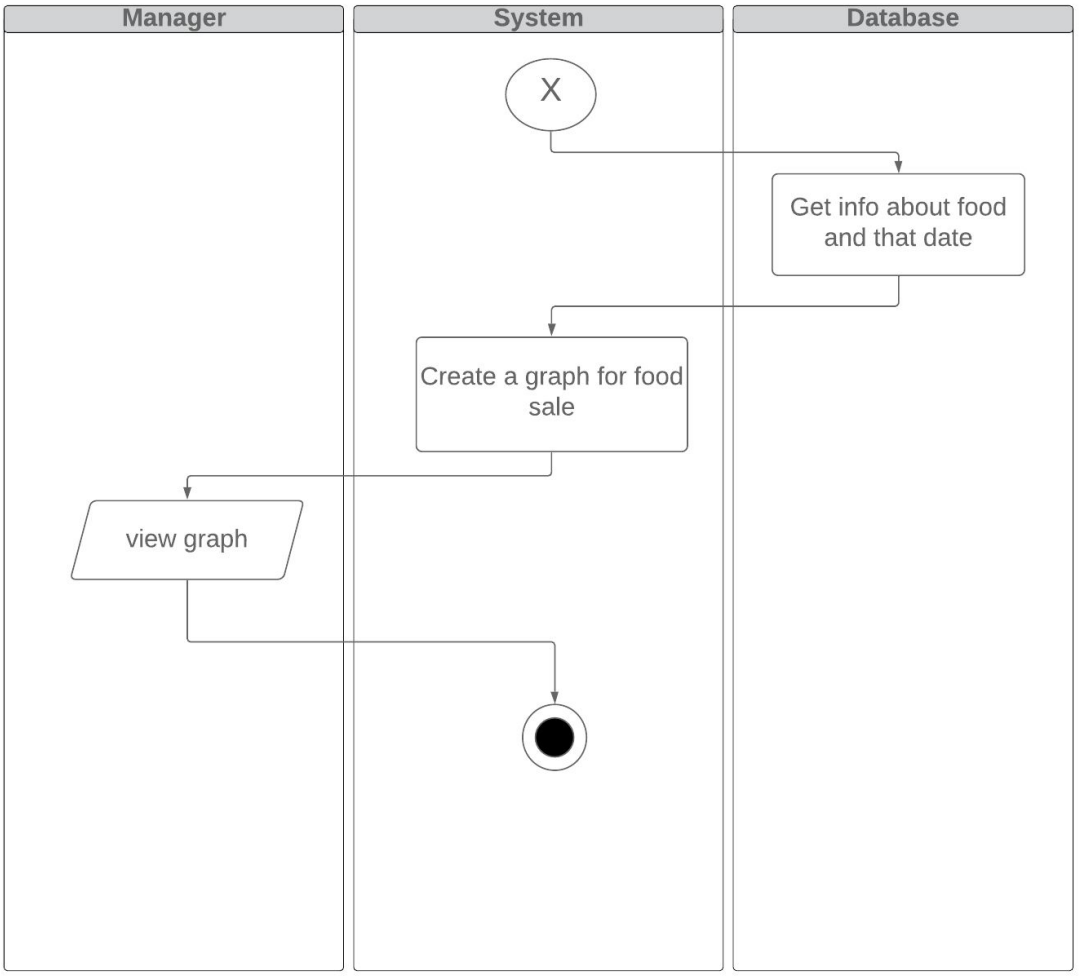


Figure 78: Swim Lane Diagram: Food sale graph.

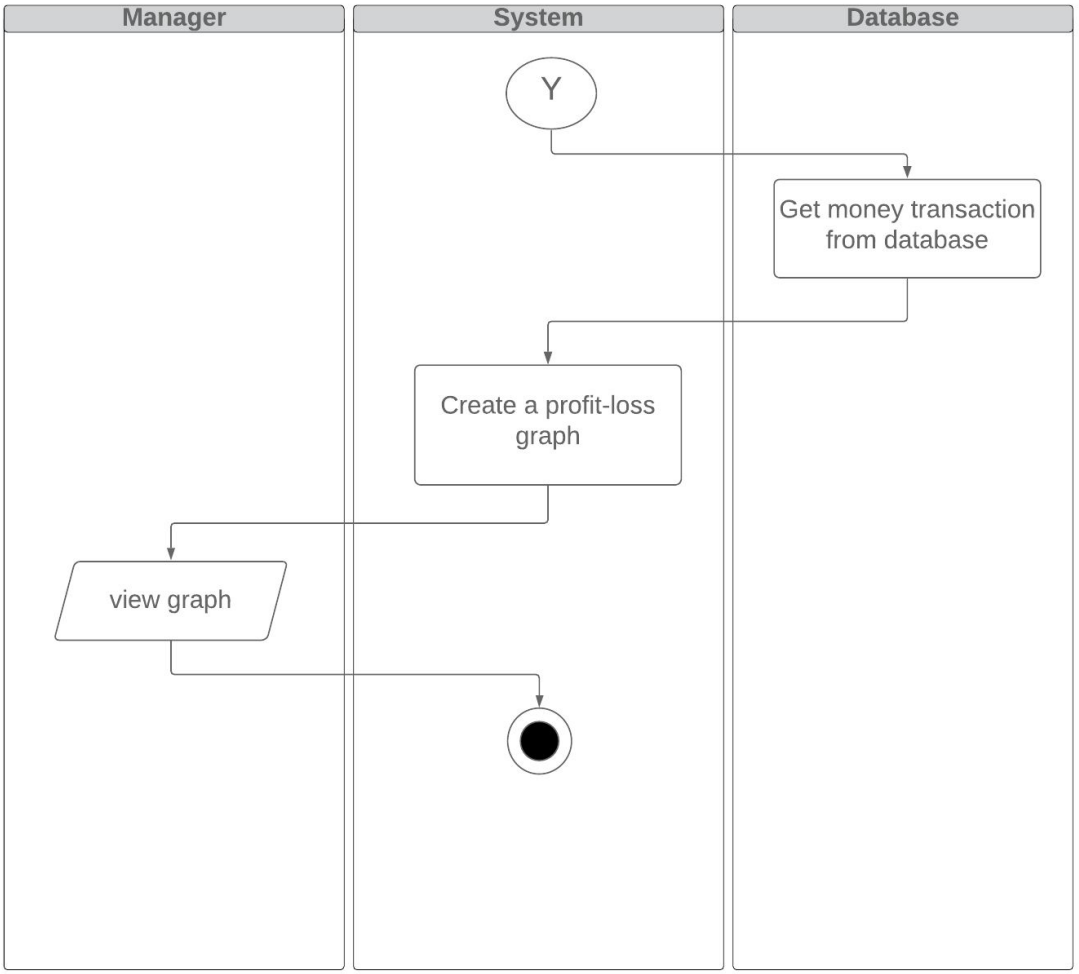


Figure 79: Swim Lane Diagram: Profit-loss graph

Chapter 5

Data Based Modeling

5.1 Introduction

Sometimes software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated. The software team chooses to create data models as a part of overall requirements modelling. 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.

5.2 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.

5.2.1 Noun identification

We identified all the nouns whether they are in problem space or in solution space from our usage scenario.

Table 1: Noun Identification for Data Modelling

Serial	Noun	P/S	Attributes
1	Cafeteria Management System	P	
2	Automated system	P	
3	Authority	P	
4	Online assist	P	
5	Teacher	S	8, 17, 18, 26-31
6	Officer	S	8, 17, 18, 26-31

7	University of Dhaka	P	
8	Food delivery	S	
9	Order processing	S	
10	Event management	S	
11	Features	P	
12	Account management	S	
13	User	S	8, 17, 18, 26-31
14	Account	S	17, 18, 26-31
15	Type	P	
16	Account creation	S	
17	Username	S	
18	Password	S	
19	User account	S	17, 18, 26-31
20	Manager's account	S	17, 18, 26-29, 31,32
21	Manager	S	17, 18, 26, 27, 29, 31
22	Staff account	S	17, 18, 26-29, 31, 34
23	Staff	S	17, 18, 26, 31, 34
24	Cafeteria	P	
25	Cashier	S	17, 18, 26-29, 31
26	Full name	S	
27	Email address	S	
28	Mobile number	S	
29	Address	S	
30	Department	P	
31	Room number	S	
32	Shift	S	
33	Name	P	
34	Post	S	
35	Cook	P	
36	Waiter	P	
37	Delivery-man	P	
38	Link	S	
39	Character	P	
40	Combination	P	
41	Letter	P	
42	Number	P	
43	Warning	S	
44	Information	P	
45	Code	S	
46	Notification	S	54, 98, 99, 43
47	Authentication	S	
48	Dhaka University Database	S	5, 6, 13
49	Dhaka University Authority	S	
50	Request message	S	
51	Confirmation	S	
52	Login	S	

53	Error dialogue box	S	
54	Screen	P	
55	Unsuccessful trial	S	
56	Reset password	S	
57	option	P	
58	Stock management	S	60, 61, 63, 64, 65-82
59	User perspective	P	
60	Menu	S	
61	Dhaka	P	
62	Price	S	
63	Food	S	
64	General food item	S	
65	Chicken	P	
66	Rice	P	
67	Daal	P	
68	Vegetable	P	
69	Rohit fish	P	
70	Beef	P	
71	Omlet	P	
72	Ruti	P	
73	Special food item	S	
74	Sunday	P	
75	Khicuri	P	
76	Monday	P	
77	Tuesday	P	
78	Wednesday	P	
79	Thursday	P	
80	Chinese food	P	
81	Polao	P	
82	Salad	P	
83	Cafeteria perspective	P	
84	Record	S	
85	Daily purchase	S	
86	Amount	S	
87	Ordered food	S	
88	Job	P	
89	Customer	S	
90	Product	S	
91	Order processing	S	95, 96, 92, 96-99
92	Food service	S	
93	Memo	S	
94	Taka	S	
95	Table reservation	S	
96	Table	P	
97	Dine-in	S	
98	Order-code	S	

99	SMS	S	
100	Booking time	S	
101	Pre-order	S	
102	Meal	P	
103	Add notes	S	
104	Payment	S	
105	BKash	S	
106	Rocket	S	
107	Card	S	
108	EDC machine	S	
109	Cash payment	S	
110	Bill	S	
111	Counter	P	
112	Money	P	
113	Procedure	P	
114	Pay from salary	S	
115	Month	P	
116	University	P	
117	Salary	S	
118	Event management	S	119, 120,121, 122-125
119	Event calendar	S	
120	Date	S	
121	Time	S	
122	Party menu	S	
123	Event time	S	
124	People	P	
125	Advance	S	
126	Method	P	
127	Cancellation	S	
128	Service charge	S	
129	Staff management	S	130-133
130	Joining date	S	
131	Attendance	S	
132	Feedback	S	
133	Rating	S	
134	Comment section	S	
135	Graphical Representation of food sell	S	136-139, 141-145
136	Food sell	P	
137	Graph	S	
138	Weekly sale	S	
139	Monthly sale	S	
140	Graphical Representation of profit-loss	S	136-139, 141-145
141	Formula	P	
142	Profit	S	
143	Earned money	S	
144	Expenses	S	

5.2.2 POTENTIAL DATA OBJECTS

- Teacher
- Officer
- User
- Account
- User Account
- Manager Account
- Staff Account
- Cashier
- Notification
- Database
- Stock Management
- Order Processing
- Event Management
- Staff Management
- Food sell Graph
- Profit-loss Graph

5.2.3 ANALYSIS FOR FINAL DATA OBJECT

- **Teacher and officers** are all users of Cafeteria Management System and have common attributes stored as data object **User.**
- **User Account** keeps username, full name, password, Email Address, Department, Room no of the user.
- **Manager Account** keeps full name, password, Email Address of the manager.
- **Staff Account** keeps full name, password, post of the staff.
- Staffs' recruitment date, salary, staff availability is kept stored in **Staff Management.**
- **Stock Management** holds all food items information and the information of earned money from the food items.
- **Cashier** has common attributes stored as data object **Staff.**
- **Event management** holds all information about the event and earned money from the event management.
- **Order Processing** contains Order No Table, Table reservation and memo.
- **Database** contains information about teachers and officers.
- **Food sell Graph** and **Profit-loss Graph** have the same attribute. So, we merge this two data object as **Profit-loss Graph.**
- **Notification** keeps notification Message, notification id.

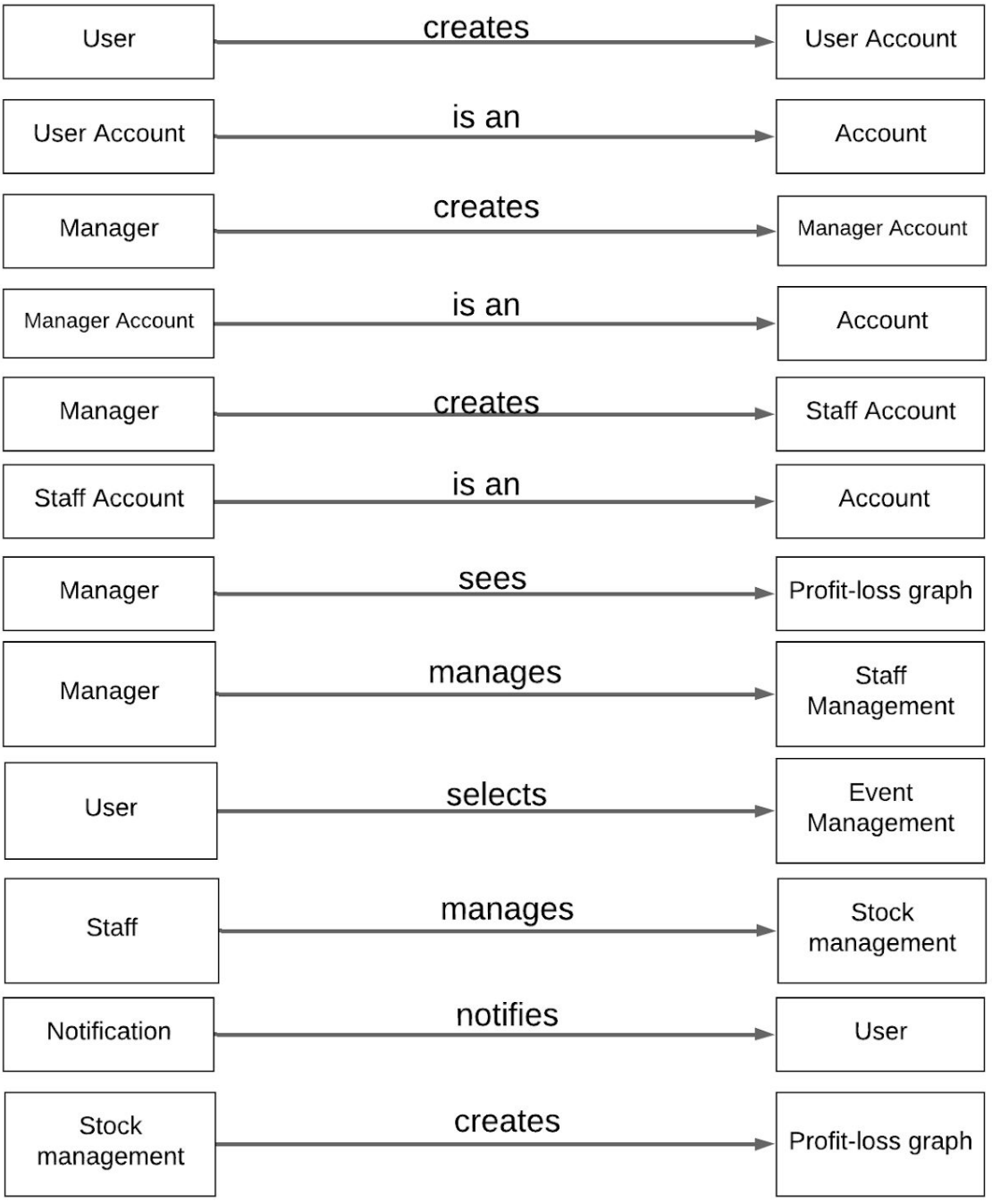
5.2.4 FINAL DATA OBJECTS

Table 2: Final Data Objects

1	**User**: User ID, Full name, Password, Email, Mobile number
2	**Manager:** fullName, password emailAddress
3	**User Account:** fullName, userName, User ID, password, emailAddress, mobileNumber, department, roomNumber
4	**Staff:** fullName fullname password emailAddress post mobileNumber
5	**Manager Account:** fullName, fullName, password emailAddress shift
6	**Staff Account:** National ID (NID), Permanent address,
7	**Staff management:** Staff_id Joining date, Attendance, Feedback, Rating
8	**Database**:
9	**Order Processing**: Order No, Table, Table reservation, memo

10	**Event Management:** Event_time, Event_Date, Party menu, Advance
11	**Stock Management:** food_id generalFoodItem specialFoodItem price amount
12	**Notification:** code mobileNumber emailAddress
13	**Profit_loss_graph:** date, expenses, earned money, profit, number_of_product

5.3 DATA OBJECT RELATIONS



5.4 Entity Relationship Diagram

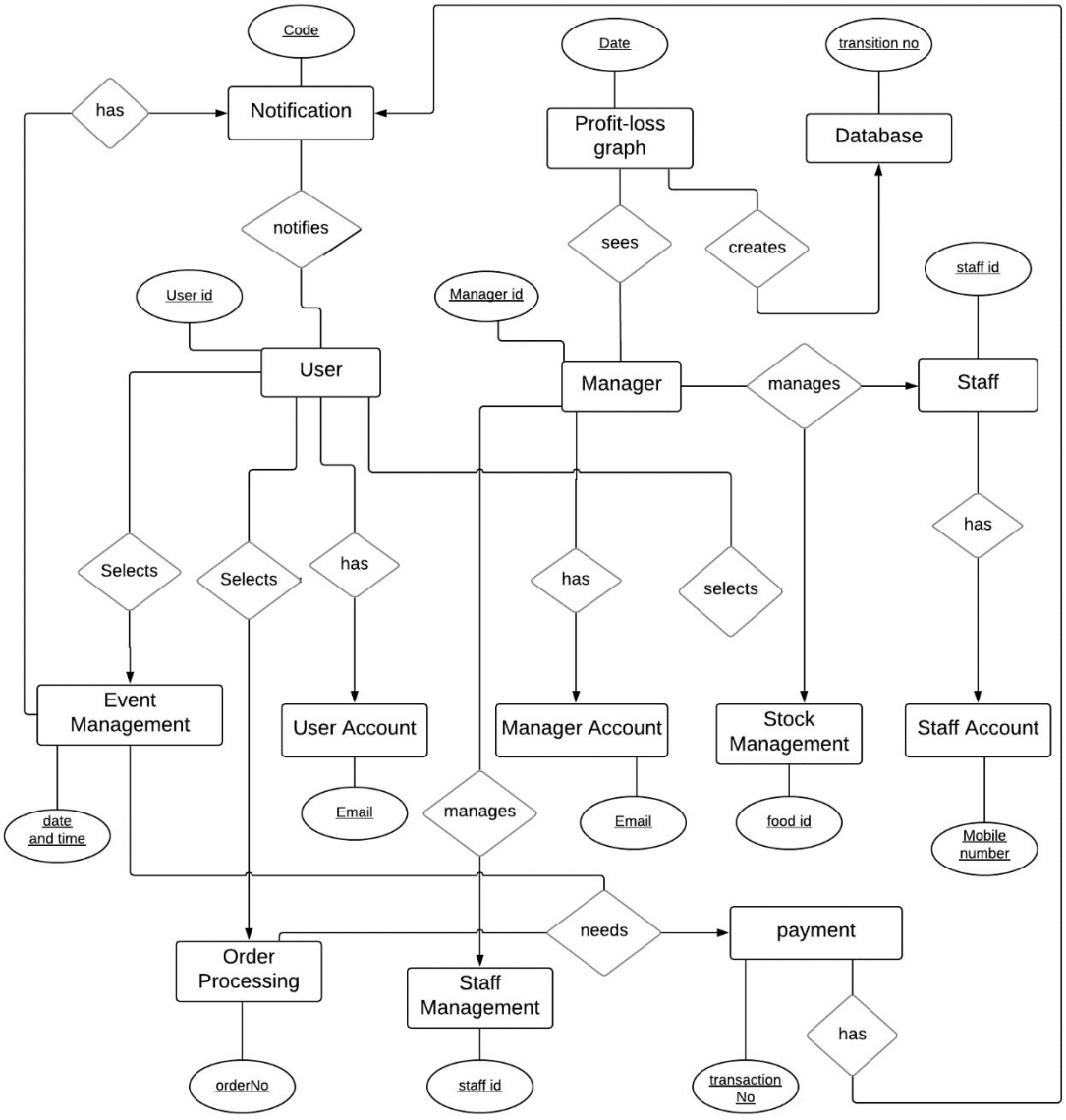


Figure 80:ER Diagram

5.5 SCHEMA DIAGRAM

USER		
Attributes	**Type**	**Size**
User id	VARCHAR	40
Full name	VARCHAR	80
Password	VARCHAR	8-20
Email	VARCHAR	30
Present Address	VARCHAR	200
Mobile Number	VARCHAR	15

MANAGER		
Attributes	**Type**	**Size**
Manager_id	VARCHAR	40
fullname	VARCHAR	40
password	VARCHAR	8-20
Email Address	VARCHAR	50
shift	VARCHAR	10

USER ACCOUNT		
Attributes	**Type**	**Size**
fullName	VARCHAR	40
User name	VARCHAR	40
User_ID	VARCHAR	20
password	VARCHAR	8-20
Email address	VARCHAR	30
Mobile number	VARCHAR	20
Room number	Number	
Department	VARCHAR	20

MANAGER ACCOUNT		
Attributes	**Type**	**Size**
Manager_id	VARCHAR	40
Username	VARCHAR	40
JoiningDate	DATE	

Salary	NUMBER	

STAFF		
Attributes	**Type**	**Size**
staffName	VARCHAR	80
staff_id	DATE	
Manager_id	VARCHAR	40
ContactNumber	VARCHAR	15
Address	VARCHAR	200
Shift	VARCHAR	50

STAFF ACCOUNT		
Attributes	**Type**	**Size**
National_ID	VARCHAR	80
Manager_id	VARCHAR	40
Post	Varchar	20
Permanent Address	Memo	

	Payment	
Attributes	**Type**	**Size**
transaction id	VARCHAR	40
User id	VARCHAR	40
code	Varchar	20

EVENT MANAGEMENT		
Attributes	**Type**	**Size**
User id	VARCHAR	40
Event_Name	VARCHAR	80
Event_Date	DATE	
Event_time	Time	

ORDER PROCESSING		
Attributes	**Type**	**Size**
User id	VARCHAR	40
Order_no	VARCHAR	20
Table	VARCHAR	20
Table Reservation	DATE	
Memo	NUMBER	

STOCK MANAGEMENT		
Attributes	**Type**	**Size**
User id	VARCHAR	40
Manager id	VARCHAR	40
food_id	VARCHAR	80
special_food_item	VARCHAR	80
general_food_item	VARCHAR	20
price	Number	

NOTIFICATION		
Attributes	**Type**	**Size**
User id	VARCHAR	40
Code	VARCHAR	20
Mobile_Number	VARCHAR	80
Email Address	VARCHAR	20

PROFIT_LOSS GRAPH		
Attributes	**Type**	**Size**
Manager_id	VARCHAR	40
Date	Date	
Time	Time	
Profit	Number	

Chapter 6

CLASS BASED MODELLING

This chapter describes the Class Based Model for the Cafeteria Management System.

6.1 Class based modelling concept

Class-based modeling is a stage of requirements modelling. It uses common concepts of object-oriented programming to craft an impression of an application that can be understood by nontechnical stakeholders. Class-based modelling represents: -

- The objects the system will manipulate
- The operation that will be applied for effective manipulation
- The relationships between the objects
- The collaborations that occur between the classes.

6.2 Identifying analysis classes

Classes are identified by underlining each noun or noun phrase and plotting it into a simple table. If the class (noun) is required to implement a solution, then it becomes a part of the solution space. Otherwise if the noun is used only to describe a solution, it is regarded as a part of the problem space. Once all the nouns have been isolated, General classification and Selection is done.

6.3 General classification

Nouns belonging to the solution space should exhibit any of the following criteria to be considered as a class. The 7 general characteristics are stated below:

No	Noun	General classification
1	Teacher	4 , 5 , 7
2	Officer	4 , 5 ,7
3	Food delivery	3
4	Order processing	3
5	Event management	3
6	Account management	3
7	User	4 , 5 , 7
8	Account	2 , 5 , 7
9	Account creation	3
10	Username	
11	Password	
12	User account	
13	Manager account	2 , 5 , 7
14	Manager	2 , 5 , 7
15	Staff account	4 , 5 , 7
16	Staff	2 , 5 , 7
17	Cashier	4 , 5 , 7
18	Full name	
19	Email address	

20	Mobile number	
21	Department	
22	Room number	
23	Post	4
24	Link	1
25	Warning	2
26	Code	1
27	Notification	2
28	Authentication	
29	Dhaka University Database	1 , 2
30	Dhaka University Authority	1
31	Request message	2
32	Confirmation	2
33	Login	3
34	Error dialogue box	
35	Trial	7
36	Reset password	3
37	Stock management	2
38	Menu	2 , 7
39	Price	2 , 7

40	Food	2
41	General food item	2
42	Special food item	2
43	Record	2, 7
44	Daily purchase	3
45	Amount	2
46	Ordered food	2, 3
47	Product	2, 5, 7
48	Food service	3
49	Memo	2, 5, 7
50	Table reservation	3
51	Table	2
52	Dine-in	3
53	Order-code	2
54	SMS	1
55	Booking time	2
56	Pre-order	3
57	Add-notes	3
58	Payment	3
59	BKash	1
60	Rocket	1

61	Card	2, 7
62	EDC machine	1
63	Cash-payment	3
64	Bill	2
65	Money	2, 5, 7
66	Pay from salary	3
67	Salary	2
68	Event calendar	2
69	Date	3
70	Party menu	2, 7
71	Event time	3
72	Advance money	2, 3
73	Cancellation	5
74	Service charge	2
75	Staff management	4, 5, 7
76	Joining date	5
77	Attendance	2, 3
78	Feedback	3
79	Rating	3, 7
80	Comment Section	2, 6
81	Graphical Representation	2, 7

82	Weekly sale	3
83	Monthly sale	3
84	Graphical Representation of Profit	2, 7
85	Profit	2, 7
86	Earned money	2
87	Expenses	2
88	Database	2

1. ***External entities:*** Other systems, devices, people that produce or consume information to be used by a computer-based system

2. ***Things*****:** Reports, displays, letters, signals that are a part of the information domain for the problem.

3. ***Events*****:** Actions or transfers (a property transfer or the completion of a series of robot movements) that occur within the context of system operation.

4. ***Roles*****:** Responsibilities played by the people who interact with the system.

5. ***Organizational units:*** Divisions, groups, teams that are relevant to an application.

6**.** ***Places*****:** Platform that establishes the context of the problem and overall function of the system.

7**.** ***Structures*****:** Something that defines a class of objects or related classes of objects.

6.4 Selection criteria

Classes that fulfilled at least 3 characteristics of general classification are again reconsidered by six Selection Criteria. The six characteristics for the selection criteria are:

1. **Retained information**: The potential class will be useful during analysis only if information about it must be remembered so that the system can function.
2. **Needed services:** The potential class must have a set of identifiable operations that can change the value of its attributes in some way.
3. **Multiple attributes:** During requirement analysis, the focus should be on "major" information; a class with a single attribute may, in fact, be useful during design, but is probably better represented as an attribute of another class during the analysis activity.
4. **Common attributes:** A set of attributes can be defined for the potential class and these attributes apply to all instances of the class.
5. **Common operations:** A set of operations can be defined for the potential class and these operations apply to all instances of the class.
6. **Essential requirements:** External entities that appear in the problem space and produce or consume information essential to the operation of any solution for the system will almost always be defined as classes in the requirements model.

To be considered a legitimate class for inclusion in the requirements model, a potential object should satisfy all (or almost all) of these characteristics. The decision for inclusion of potential classes in the analysis model is somewhat subjective, and later evaluation may cause an object to be discarded or reinstated.

No	Potential classes	Accepted Criteria
1	Teacher	1, 2, 3, 4, 5, 6
2	Officer	1, 2, 3, 4, 5, 6
3	Order processing	6
4	Event management	6

5	User	1, 2, 3, 4, 5, 6
6	Account	1, 2, 3, 4, 5, 6
7	User account	1, 2, 3, 4, 5
8	Manager account	1, 2, 3, 4
9	Manager	1, 2, 3, 4, 5, 6
10	Staff account	1, 2, 3, 4, 5
11	Staff	1, 2, 3, 4, 5, 6
12	Cashier	1, 2, 3, 4, 5, 6
13	Notification	1, 2, 3, 6
14	Authentication	3, 6
15	Stock management	1, 2, 3, 4, 5, 6
16	Menu	1, 3, 4, 5, 6
17	Payment	1, 6
18	Staff management	1, 2, 6
19	Graph	3, 6
20	Database	1, 2, 6

6.5 Attribute selection

After identifying the classes, we have specified their attributes:

No	Name	Attributes
1	Teacher	-fullName -userName -password -emailAddress -mobileNumber -department

		-roomNumber
2	Officer	-fullName -userName -password -emailAddress -mobileNumber -department -roomNumber
3	User	-fullName -userName -password -emailAddress -mobileNumber -department -roomNumber
4	Manager	-fullName -userName -emailAddress
5	Staff	-fullName -userName -post -mobileNumber
6	Cashier	-fullName -username -password -emailAddress -post

		-mobileNumber
7	Account	-fullName -userName -emailAddress
8	User account	-fullName -userName -password -emailAddress -mobileNumber -department -roomNumber
9	Manager account	fullName -userName -emailAddress
10	Staff's account	-fullName -username -password -emailAddress -post -mobileNumber
11	Notification	-code -mobileNumber -emailAddress
12	Stock Management	-generalFoodItem -specialFoodItem -price -amount

13	Authentication	-user -universityDatabase
14	Menu	-generalFoodItem -specialFoodItem -price
15	Graph	-date -money
16	Payment	-money -paymentMethod
17	StaffManagement	-staffName -staffPost -joiningDate -attendance
18	Database	
19	Event management	-eventCalender -eventDate -partyMenu -eventTime -advanceMoney -serviceCharge -rating -feedback
20	Order Processing	-table -orderCode -bookingTime -notes

6.6 Method identification

No	Name	Method
1	Teacher	+ signUp() + signIn() + searchUser() + validateUser() + editProfile() + orderFood() + manageEvent() + payment() + reserveTable()
2	Officer	+ signUp() + signIn() + searchUser() + validateUser() + editProfile() + orderFood() + manageEvent() + payment() + reserveTable()
3	User	+ signUp() + signIn() + searchUser() + validateUser() + editProfile() + orderFood() + manageEvent() + payment() + reserveTable()
4	Manager	+ createManager()

		+ createStaff() + modifyManager() + signIn() + showItem() + showGraph() + seeActivity() + manageEvent() + manageStock()
5	Staff	+ sign in () + confirmPayment() + printMemo()
6	Cashier	+ signIn() + confirmPayment() + printMemo() + storeInfo()
7	Account	+ signUp() + signIn() + searchUser() + validateUser() + editProfile() + orderFood() + manageEvent() + payment() + reserveTable()
8	User account	+ signUp() + signIn() + searchUser() + validateUser() + editProfile() + orderFood() + manageEvent() + payment() + reserveTable()
9	Manager account	+ createManager() + createStaff() + modifyManager() + signIn() + showItem() + showGraph() + seeActivity() + manageEvent()

		manageStock()
10	Staff account	+ signIn() + confirmPayment() + printMemo() + storeInfo()
11	Notification	+ accountCreate() + unsuccessfulTrial() + successfulPayment() + eventBooking() + lowFoodQuantity()
12	Stock Management	+ deductFood() + feedBack() + storeInfo()
13	Authentication	+ setUser() + validateInput() + findUser() + getUser()
14	Menu	+ showMenu() + addItems() + delete() + suggest() + edit()
15	Graph	+ earn() + expanses() + addOption() + showGraph() + timeRange()
16	Payment	+ payBkash() + payDBBL() + payRocket() + netBill() + cash()
17	StaffManagement	+ getSalary() + setSalary() + getAvailableStaff() + setBusy() + showInfo()

18	Database	+ Accounts() + foodList() + purchaseRecord() + orders() + events()
19	Event management	+ selectDate() + eventInfo() + paymentProcedure() + cancelation() + AdvancePayment() + StoreInfo()
20	Order Processing	+ tableReserve() + foodDelivery() + payment() + reservation()

6.7 Finalizing classes

To identify the final classes, we need to check if there can be any hierarchies or merges. These identifications are given below:

- After finishing G.C and S.C, we have come up with 20 potential classes.
- Teacher, officer and user class have same attributes and methods. So instead of making three different classes, we will only make a class for user.
- Staff and cashier have the same attributes and methods, so we will make a class for staff. As all cashier's information and actions can be represented by the staff class.
- There are four classes like account, user account, manager account and staff account. The attributes and functions of these classes are similar to the user, manager and staff classes. So there is no necessity of these classes when all the information are available in user, manager and staff classes.
- We create a new class here named" University" for checking email address from the university authority database. It will be also used for salary deduction also.

So, our final class list is given below: -

1. User.
2. Manager.
3. Staff.
4. Notification.
5. Stock management.
6. Authentication.
7. Menu.
8. Graph.
9. Payment.
10. Staff management.
11. Database.
12. Event management.
13. Order processing.
14. University.

We will create class for Authentication, Notification, Stock Management, Staff, Manager, Menu, Graph, Payment, Database, Event Management, University, Order Processing and HR Management.

6.8 Class cards

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

User	
Attribute	**Method**
− fullName − userName − password − emailAddress − mobileNumber − department − roomNumber	+ signUp() + signIn() + searchUser() + validateUser() + editProfile() + orderFood() + manageEvent() + payment() + reserveTable()

Responsibilities	**Collaborator**
Create account Log in Order food Pay amount Add friend	Manager Database Authentication Stock management Menu Event Management

Manager	
Attribute	**Method**
− fullName − userName − emailAddress	+ createManager() + createStaff() + modifyManager() + signIn() + showItem() + showGraph() + seeActivity() + manageEvent() + manageStock()
Responsibilities	**Collaborator**
Create account Create staff account Log in	Graph Staff management User

Manage stock Show graph Store staff info Store staff availability	Database Stock management Menu Event Management

Staff	
Attribute	**Method**
− fullName − userName − post − mobileNumber − joiningDate − attendance	+ signIn() + confirmPayment() + printMemo() + getSalary() + setSalary() + getAvailableStaff() + setBusy() + showInfo()
Responsibilities	**Collaborator**
Print memo Confirm order	Manager Database

Notification	
Attribute	**Method**
− code − mobileNumber	+ accountCreate() + unsuccessfulTrial() + successfulPayment()

− emailAddress	+ eventBooking() + lowFoodQuantity()
Responsibilities	**Collaborator**
Notify user	Payment Authentication Order Processing User

Stock management	
Attribute	**Method**
− generalFoodItem − specialFoodItem − price − amount	+ generalFoodItem () + specialFoodItem () + price () + amount ()
Responsibilities	**Collaborator**
Deduct food item Receive feedback	User Manager

Authentication	
Attribute	**Method**

− user − universityDatabase	+ setUser() + validateInput() + findUser() + getUser()
Responsibilities	**Collaborator**
Verify email address	Database User

Menu	
Attribute	**Method**
− generalFoodItem − specialFoodItem − price	+ showMenu() + addItems() + delete() + suggest() + edit()
Responsibilities	**Collaborator**
Show menu Add items Delete items Receives suggestion Edit food item	User Manager Stock management Database Order processing

Graph

Attribute	**Method**
− date − money	+ earn() + expanses() + addOption() + showGraph() + timeRange()
Responsibilities	**Collaborator**
Receive earning data Receive expenses Add info Show graph	Manager Database

Payment	
Attribute	**Method**
− Money − paymentMethod	+ payBkash() + payDBBL() + payRocket() + netBill() + cash()
Responsibilities	**Collaborator**
Connect with gateway Create bill	University Notification

	Order processing Event Management

Staff management	
Attribute	**Method**
− staffName − staffPost − joiningDate − attendance	
Responsibilities	**Collaborator**
Store staff info Store staff availability	Database Manager

Database	
Attribute	**Method**
	+ Accounts() + foodList() + purchaseRecord() + orders() + events()

Responsibilities	**Collaborator**
Provide food info Provide expense info Provide earning info	User Manager Staff Staff management Graph Authentication Payment

Event management	
Attribute	**Method**
− eventCalender − eventDate − partyMenu − eventTime − advanceMoney − serviceCharge − rating − feedback	+ selectDate() + eventInfo() + paymentProcedure() + cancelation() + AdvancePayment() + StoreInfo()
Responsibilities	**Collaborator**
Select date Cancel event Receive event info	Payment Manager User

Order processing	
Attribute	**Method**
− table − orderCode − bookingTime − notes	+ tableReserve() + foodDelivery() + payment() + reservation()
Responsibilities	**Collaborator**
Reserve table Deliver food	User Payment Notification Graph

University	
Attribute	**Method**
-emailAddress	+checkEmail () +salaryDeduction()
Responsibilities	**Collaborator**

Match email address Deduct salary	Payment

6.9 CRC Diagram

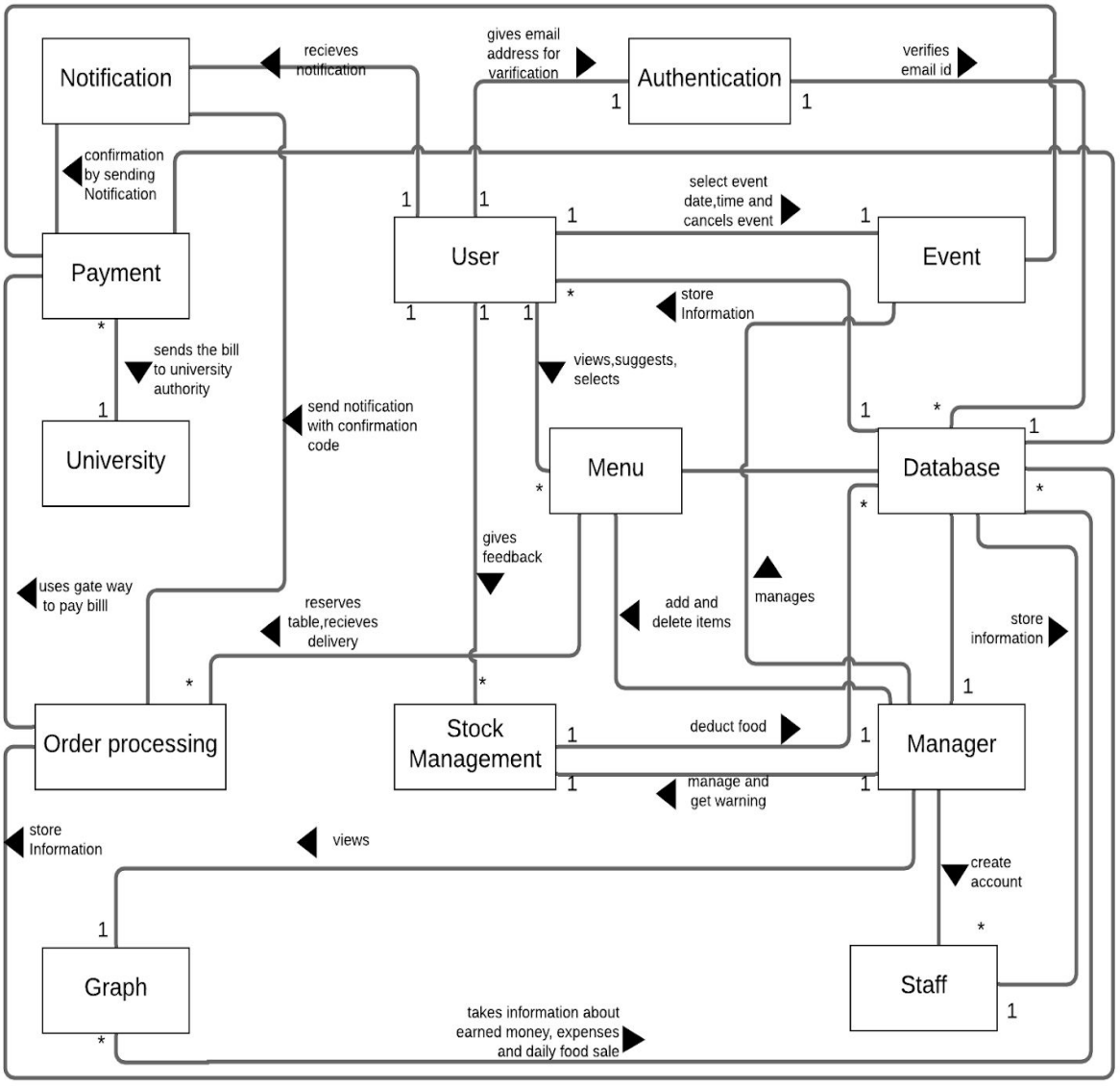


Figure 81: CRC diagram

Chapter 7

DATA FLOW MODELING

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

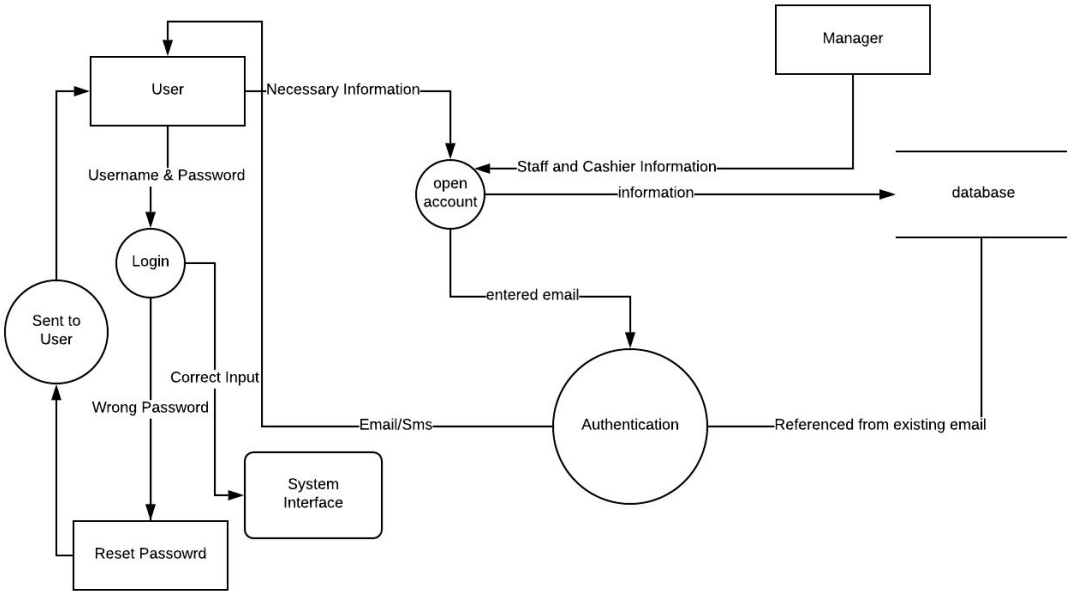


Figure 82

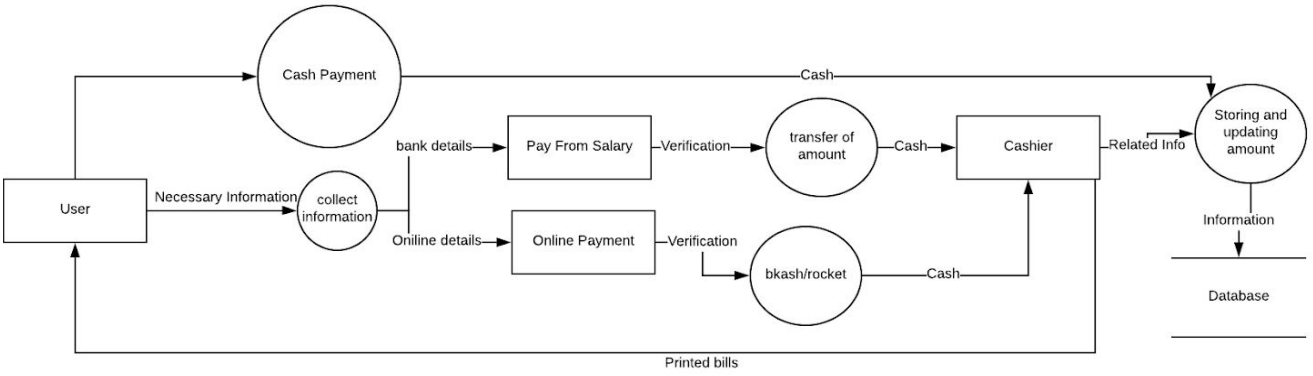


Figure 83

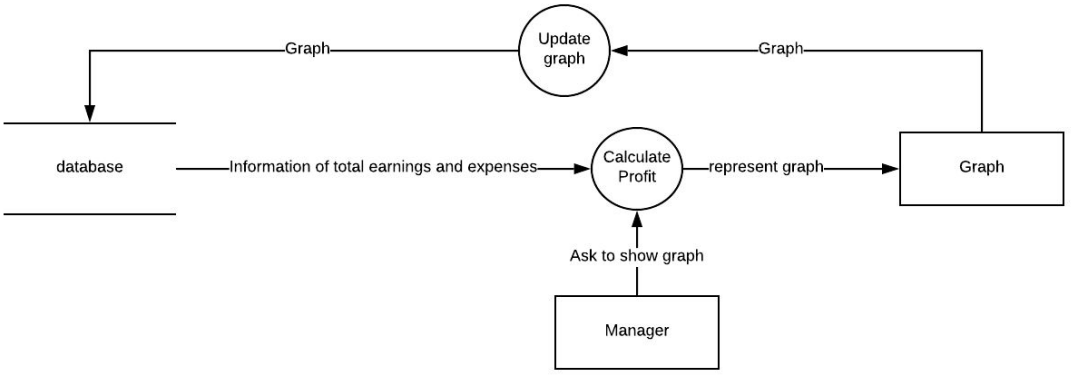


Figure 84

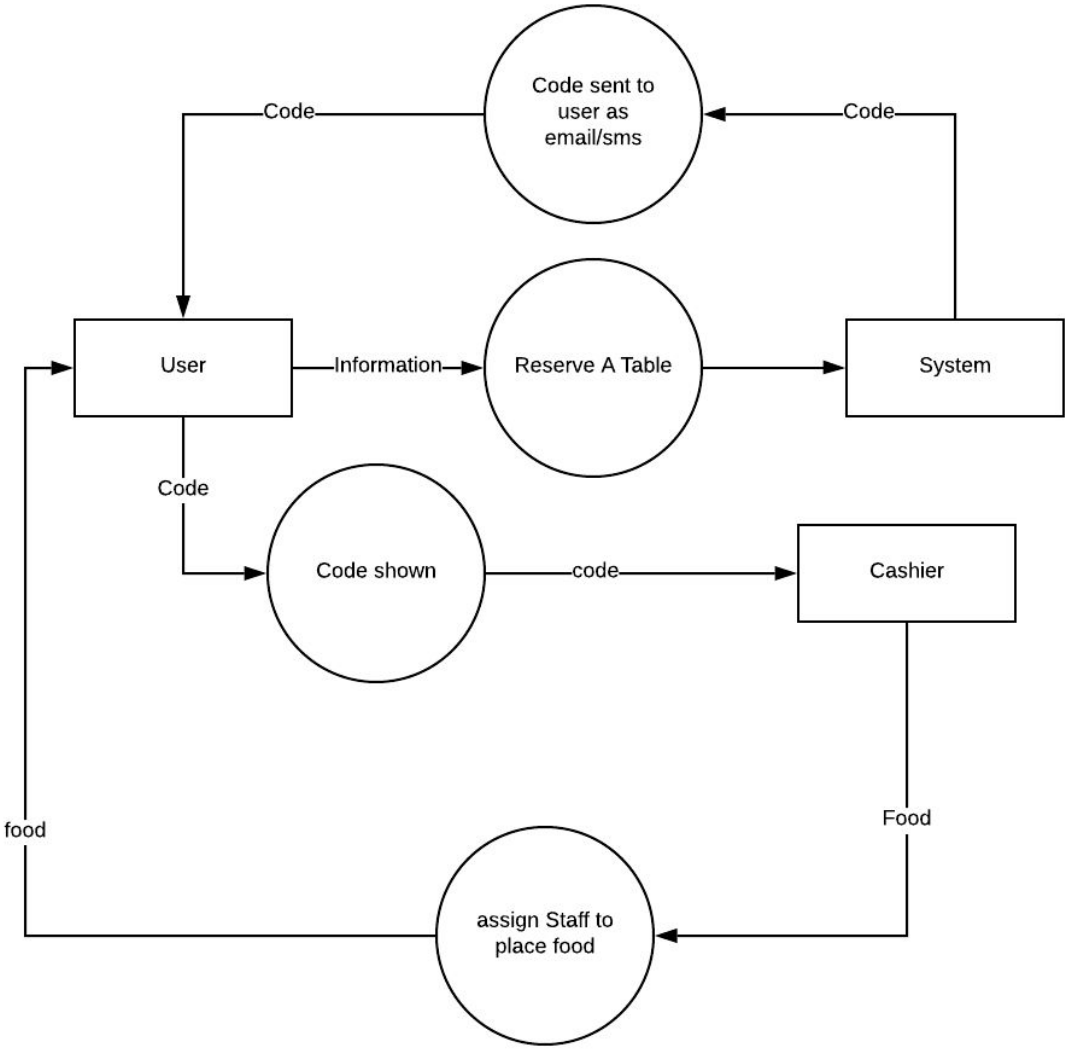

Code sent to user as email/sms
Code
Code
User
Information
Reserve A Table
System
Code
Code shown
code
Cashier
food
Food
assign Staff to place food

Figure 85

Chapter 8

BEHAVIORAL MODELING

8.1 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.

Identifying Events

Serial No.	Event	State Name	Initiator	collaborator
1	Create an account	Create account	Use Manager	Database, Authentication, University, Notification Staff
2	Get notification	Notify	Notification	User Manager staff
3	Can log in	Log in	User Manager staff	Database
4	Can log out	Log out	User Manager staff	
5	Can edit account	Edit account	User Manager	Authentication Database notification
6	Can delete account	Delete Account	User Manager	Authentication Database notification
7	Can recover account	Recover Account	User	Database

			Manager Staff	Notification
8	Can add friend	Add friend	user	database
9	Can remove friend	Remove friend	user	Database Notification
10	Can add to cart	Add to cart	user	Menu database
11	Can remove food items from cart	Remove from cart	user	Menu database
12	Can suggest food to another user	Suggest food	user	Stock management notification
13	Can add items to the menu	Add item	manager	Stock management Notification database
14	Can edit items to the menu	Edit item	manager	Stock management Notification database
15	Can remove items from the menu	Remove item	manager	Stock management Notification database
16	Can manage food record	Manage food record	staff	Stock management database
17	Will give signal for shortage of item	Shortage alert	Stock management	Notification manager
18	Can choose food delivery process	Food delivery	User	Staff Order processing Payment Database notification
19	Can give order for reserving a table	Reserve table	User	Order processing Payment Database notification

20	Can pre-order food	Pre-order	User	Order processing Notification database
21	Add food preference	Food preferences	user	Order processing manager
22	Can pay online	Online Payment	user	Payment Database notification
23	Request for transaction	Request transaction	payment	
24	Can pay in cash	Cash payment	user	Staff Database Notification payment
25	Notify after cash payment	Confirm Order	staff	Database notification
26	Will deduct salary	Pay from salary	user	University Notification database
27	Store order information	Store order info	payment	Database
28	Show event calendar	Event calendar	manager	Event management
29	Give event info	Give event details	user	manager
30	Give event confirmation	Confirm event	manager	Payment Notification Database
31	Can cancel event	Cancel event	user	Event management Database Notification manager
32	Will add staff	Add staff	manager	Database Notification Staff
33	Can delete staff info	Delete staff	manager	Database

				notification
34	Give feedback	Feedbacks	user	Manager database
35	Show graph	Get graph info	Graph	Database
36	Show sale graph	Sales graph	Manager	Graph database
37	Add expense	Add expense	Manager	Database notification
38	Delete expense	Delete expense	Manager	Database Notification
39	Show profit graph	Profit graph	Manager	Graph database
40	Will withdraw money	Withdraw Money	manager	Database Notification

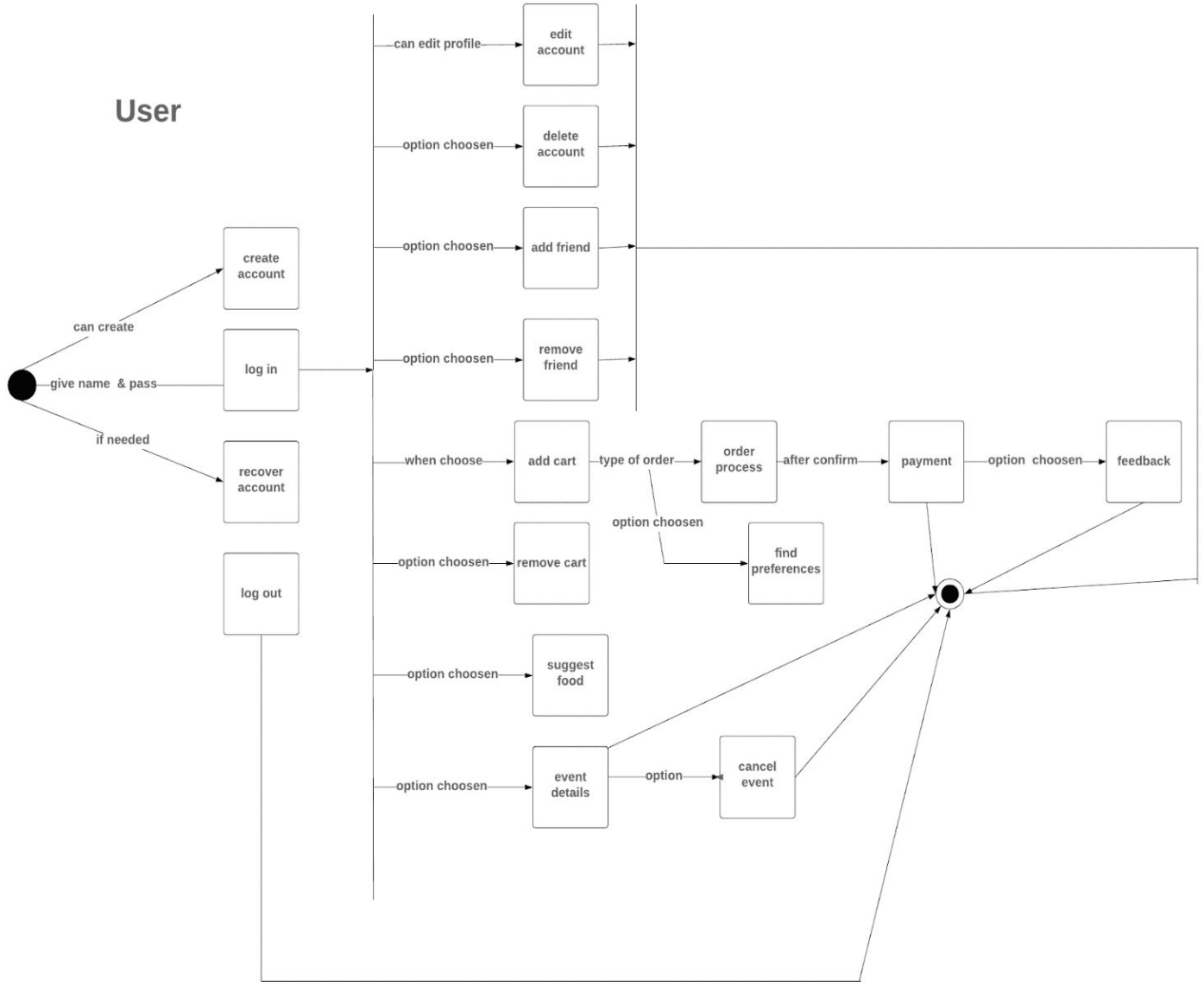


Figure 86 : State Transition Diagram-User

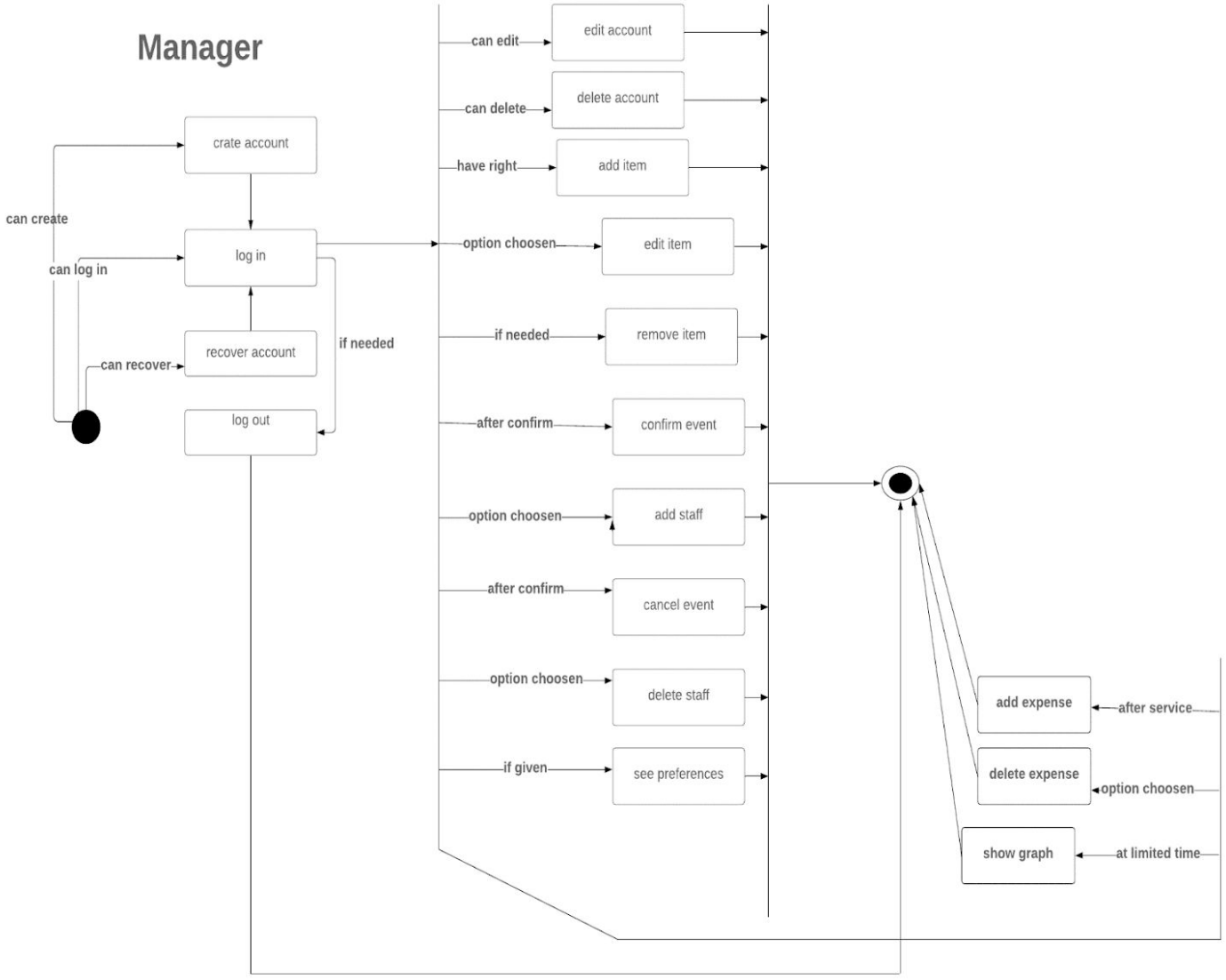


Figure 87: State Transition Diagram-Manager

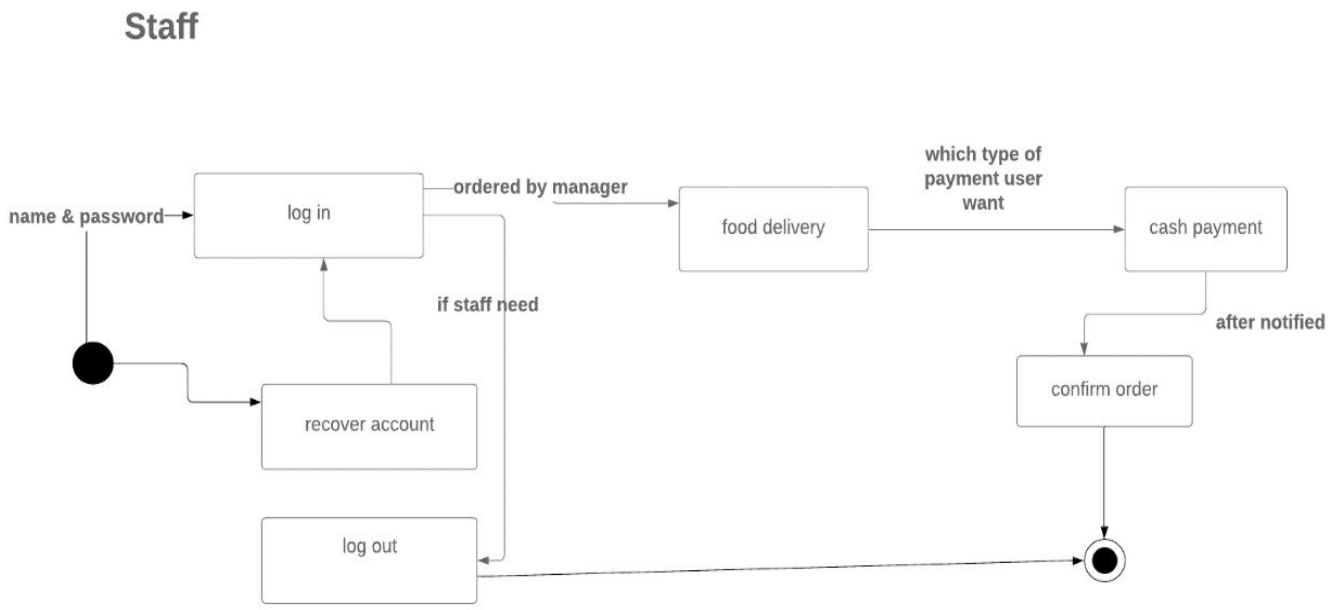


Figure 88: State Transition Diagram-Staff

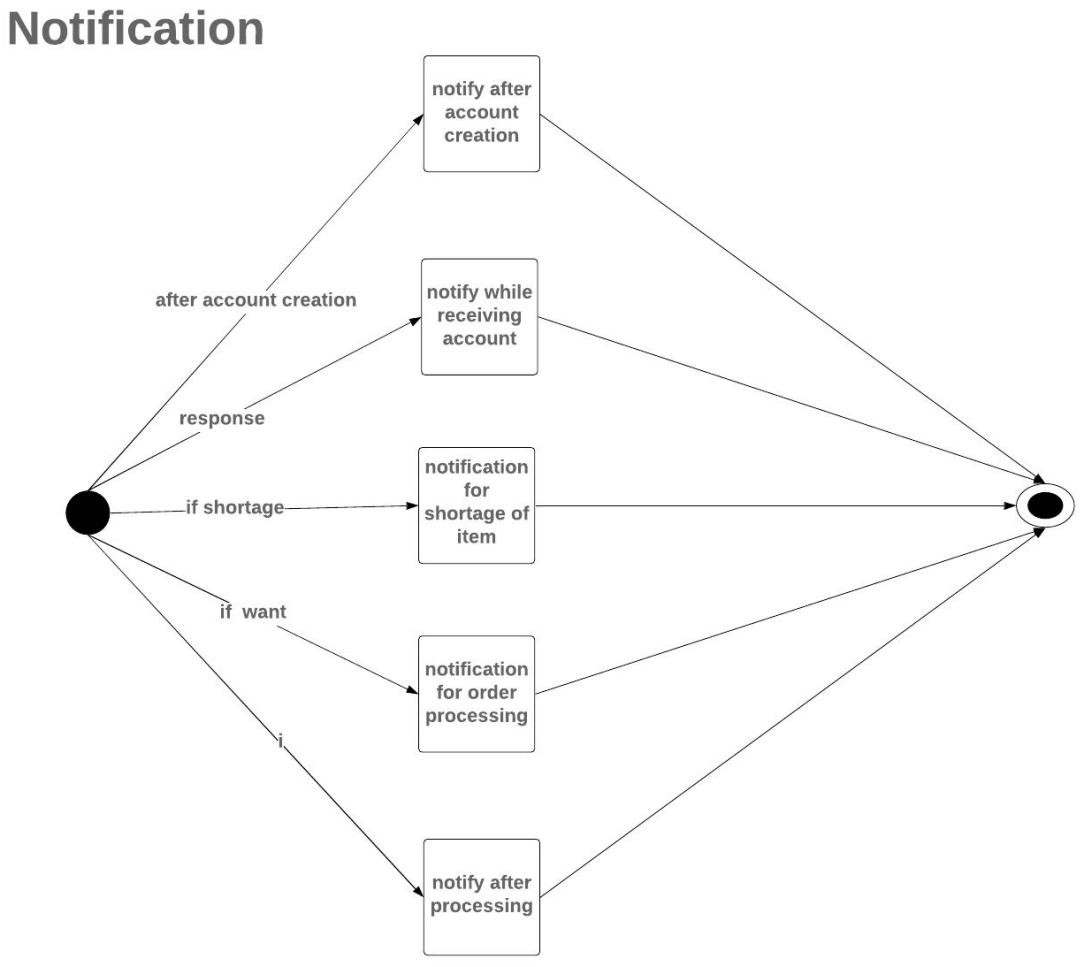


Figure 89 : State Transition Diagram-Notification

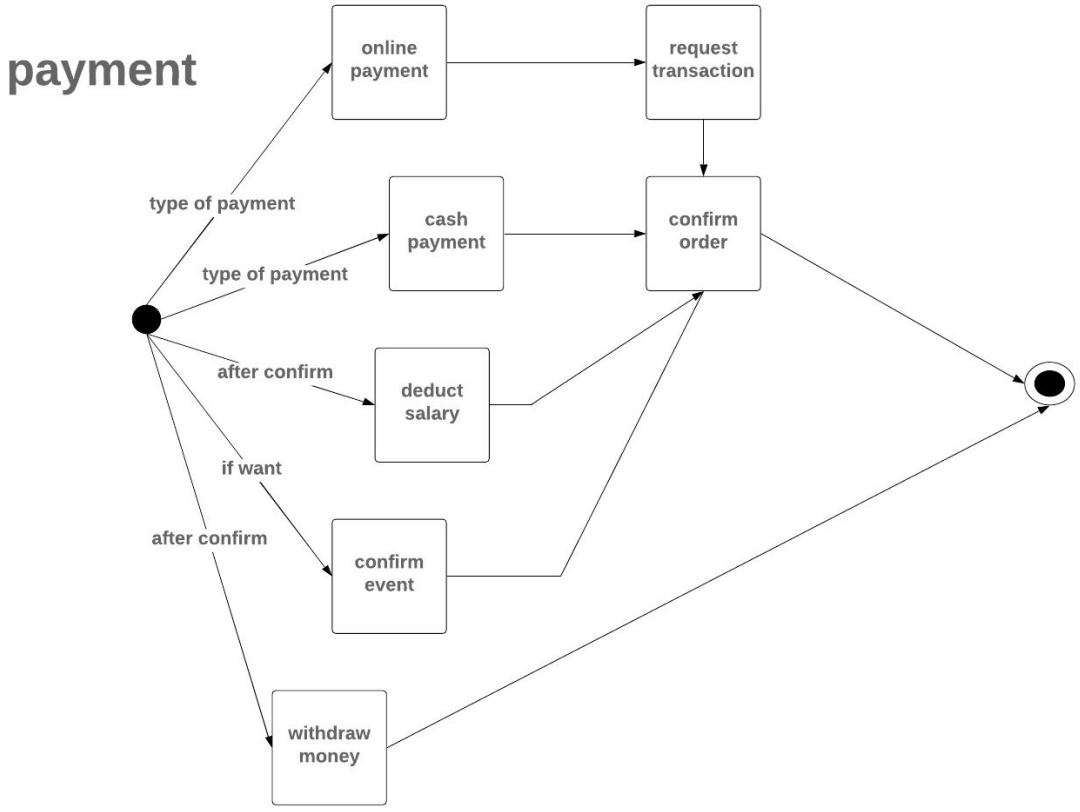


Figure 90 : State Transition Diagram-Payment

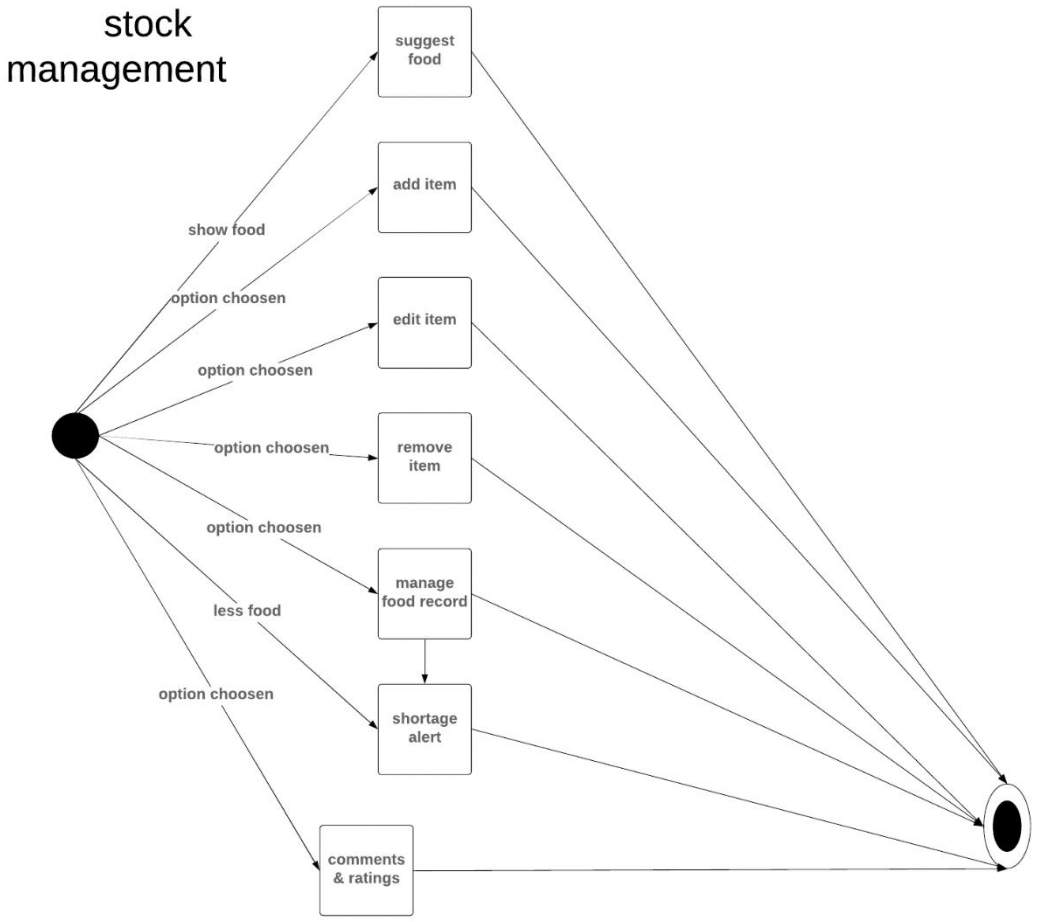


Figure 91: State Transition Diagram-Stock management

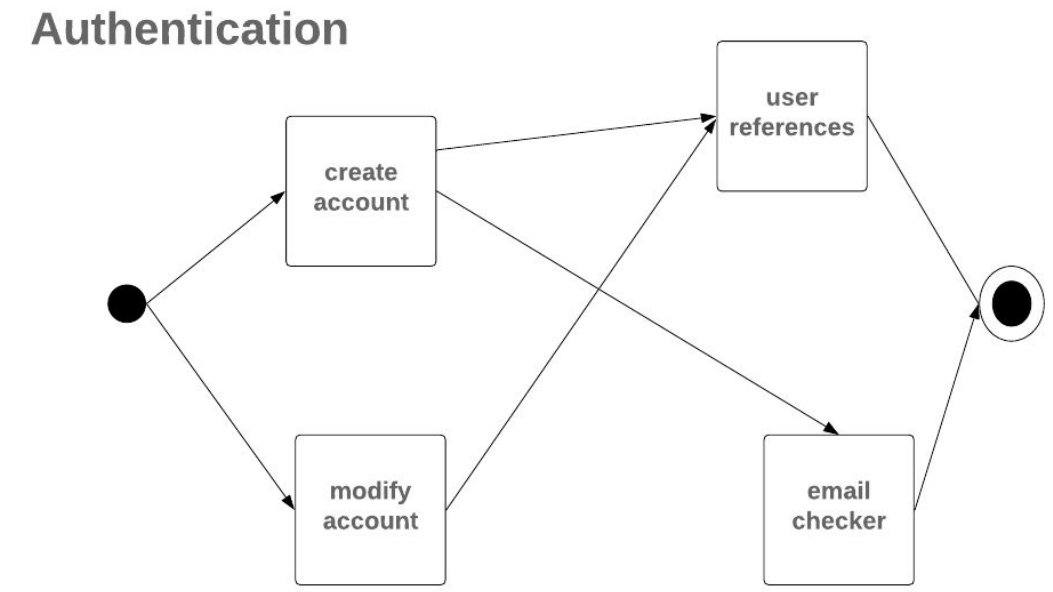


Figure 92: State Transition Diagram-Authentication

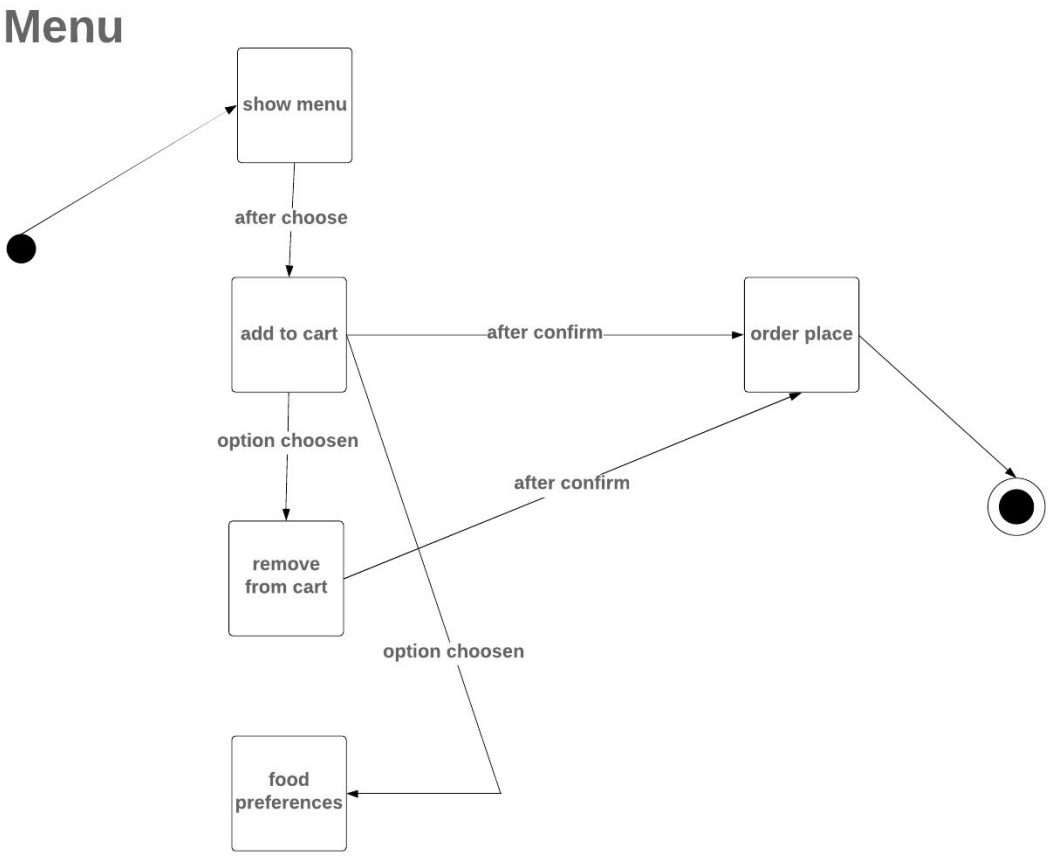


Figure 93 : State Transition Diagram-Menu

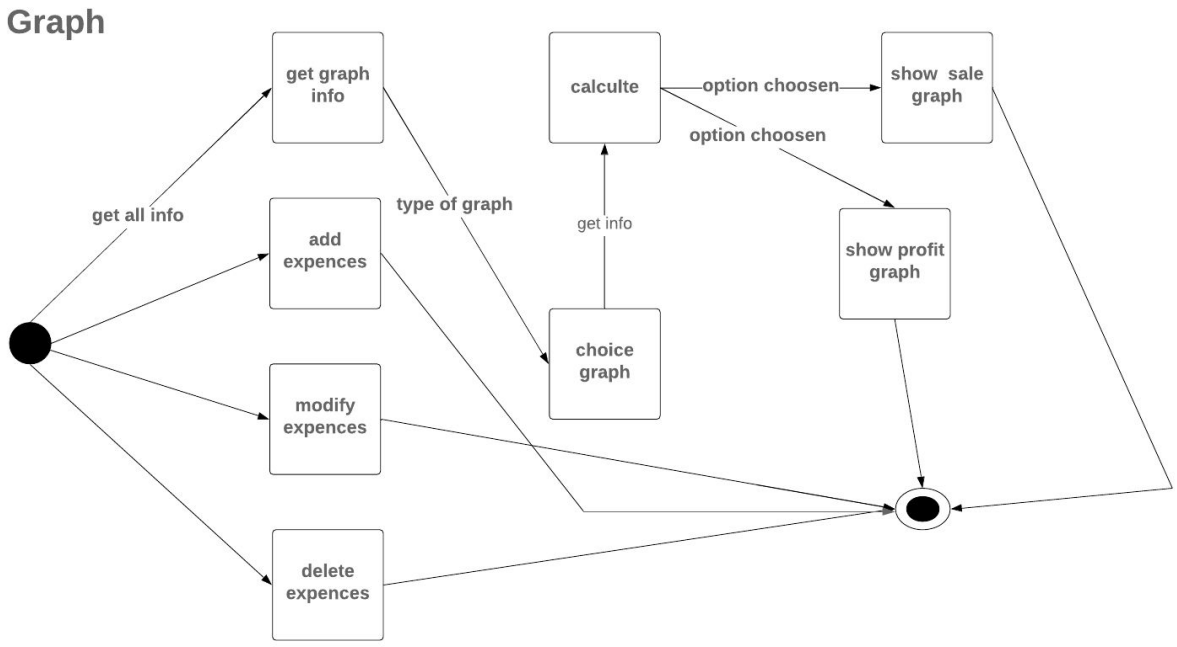


Figure 94 : State Transition Diagram-Graph

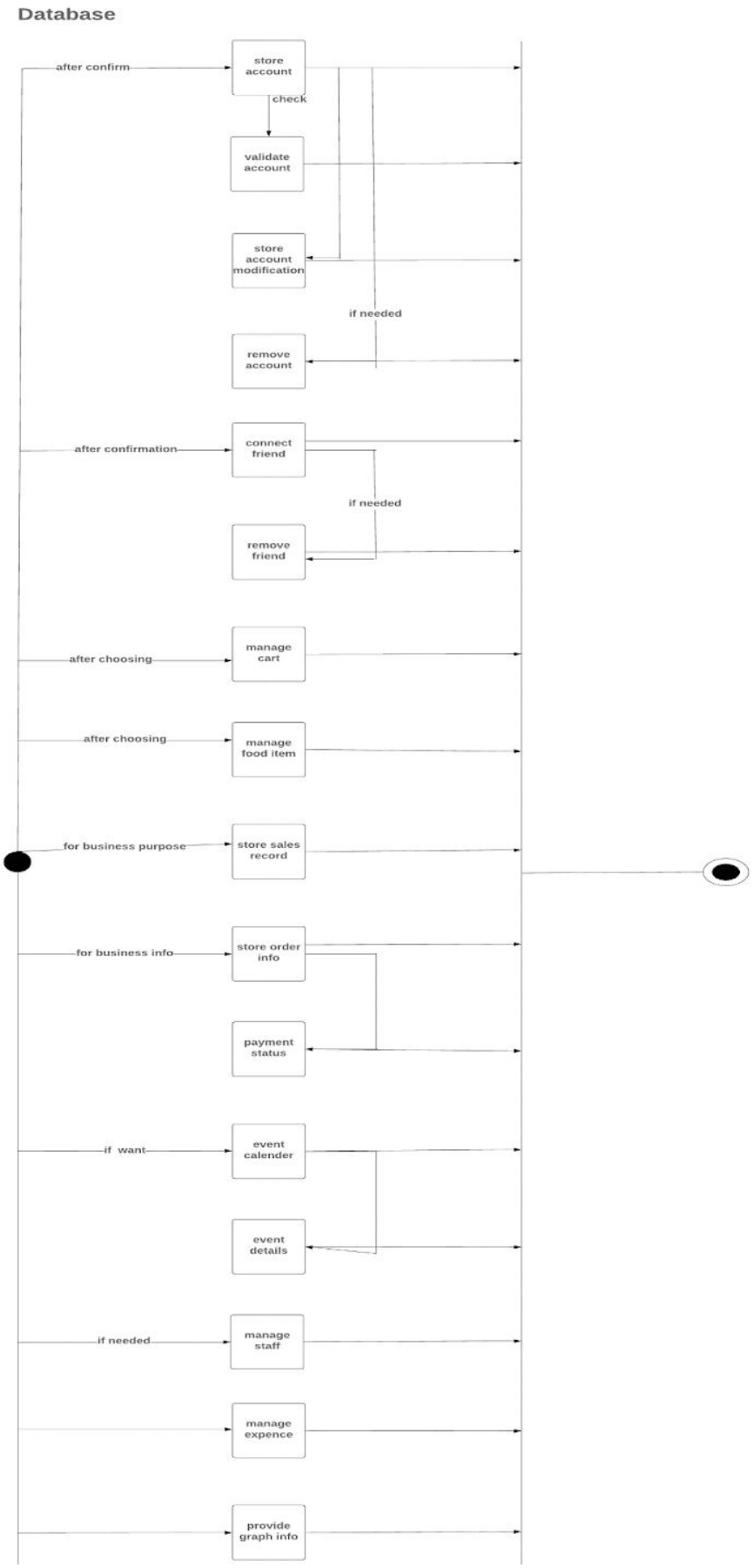

Database
after confirm
store account
check
validate account
store account modification
if needed
remove account
after confirmation
connect friend
if needed
remove friend
after choosing
manage cart
after choosing
manage food item
for business purpose
store sales record
for business info
store order info
payment status
if want
event calender
event details
if needed
manage staff
manage expence
provide graph info

Figure 95 : State Transition Diagram-Database

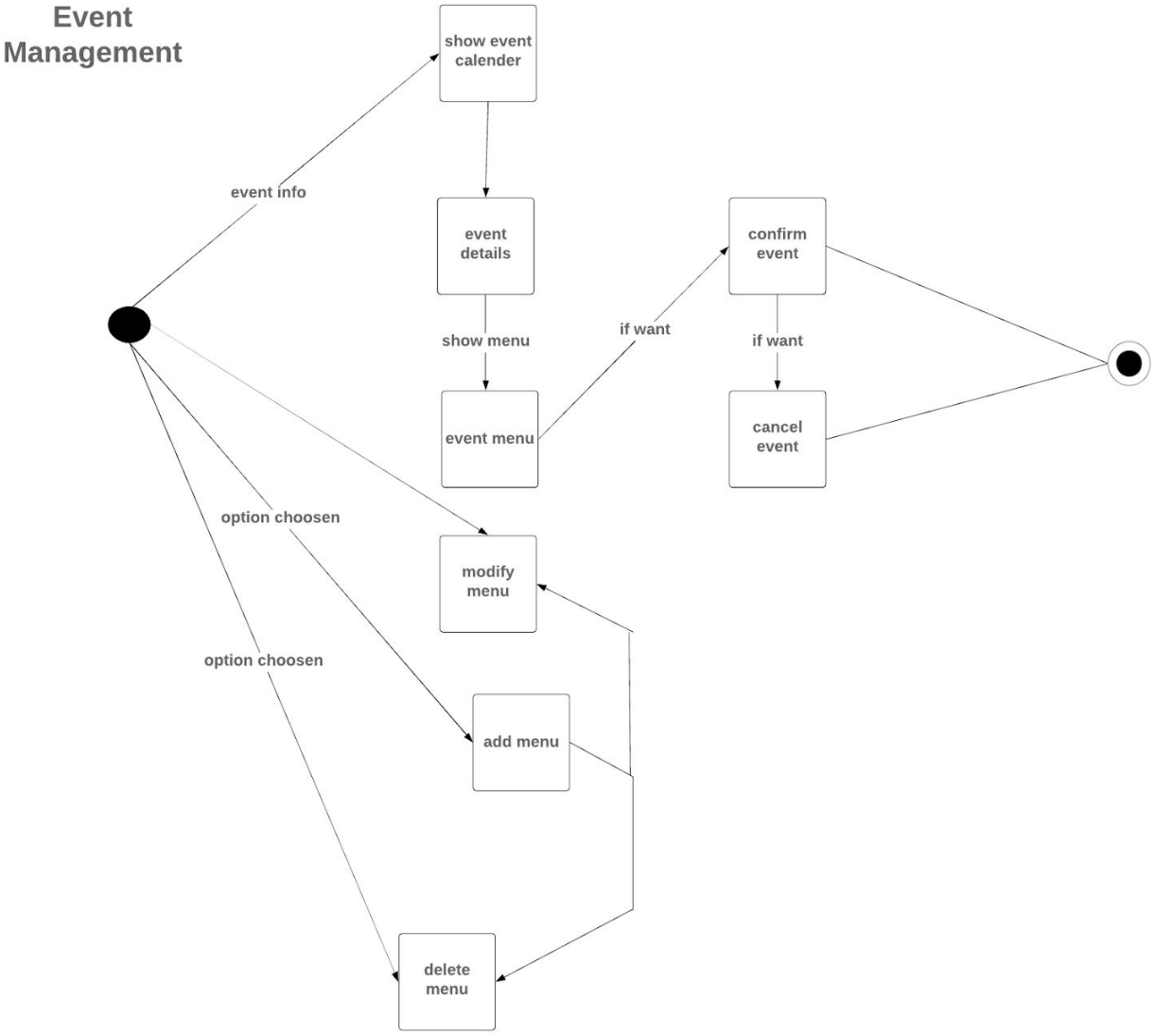


Figure 96: State Transition Diagram-Event management.

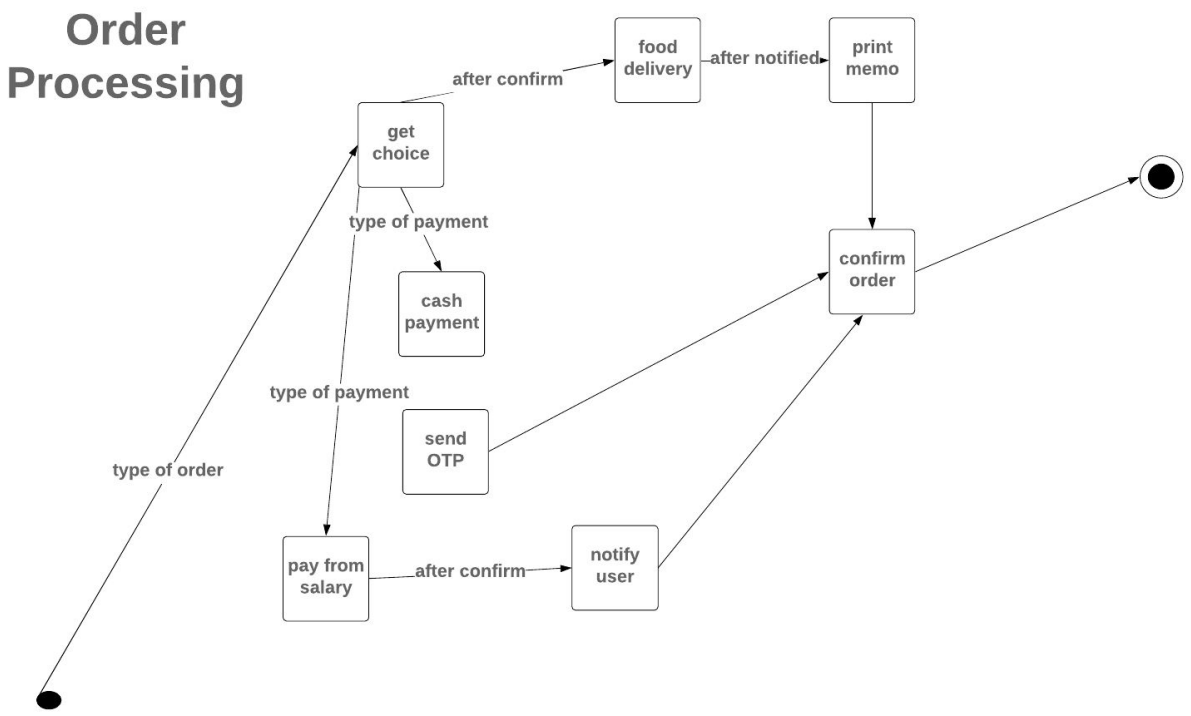


Figure 97: State Transition Diagram-Order processing.

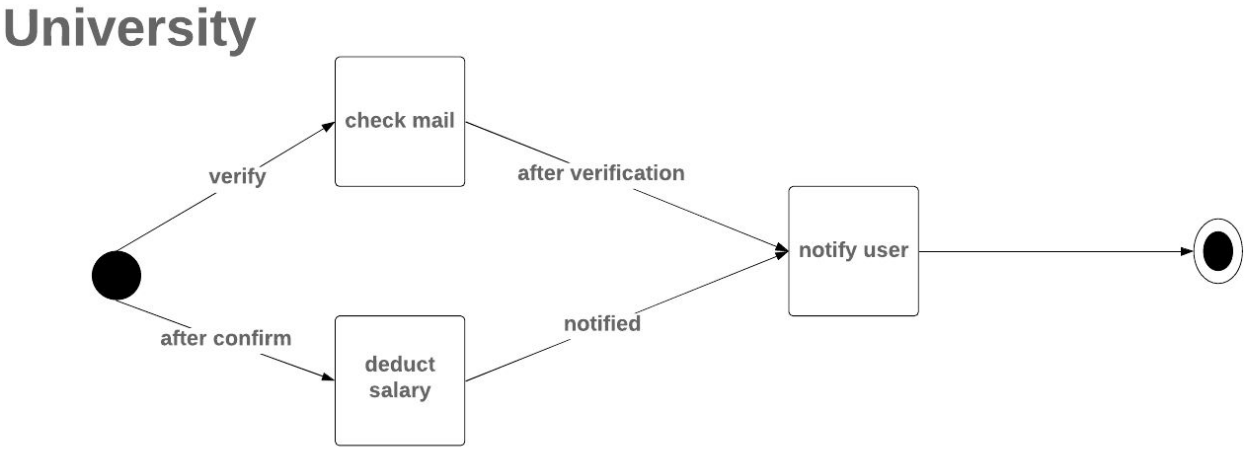
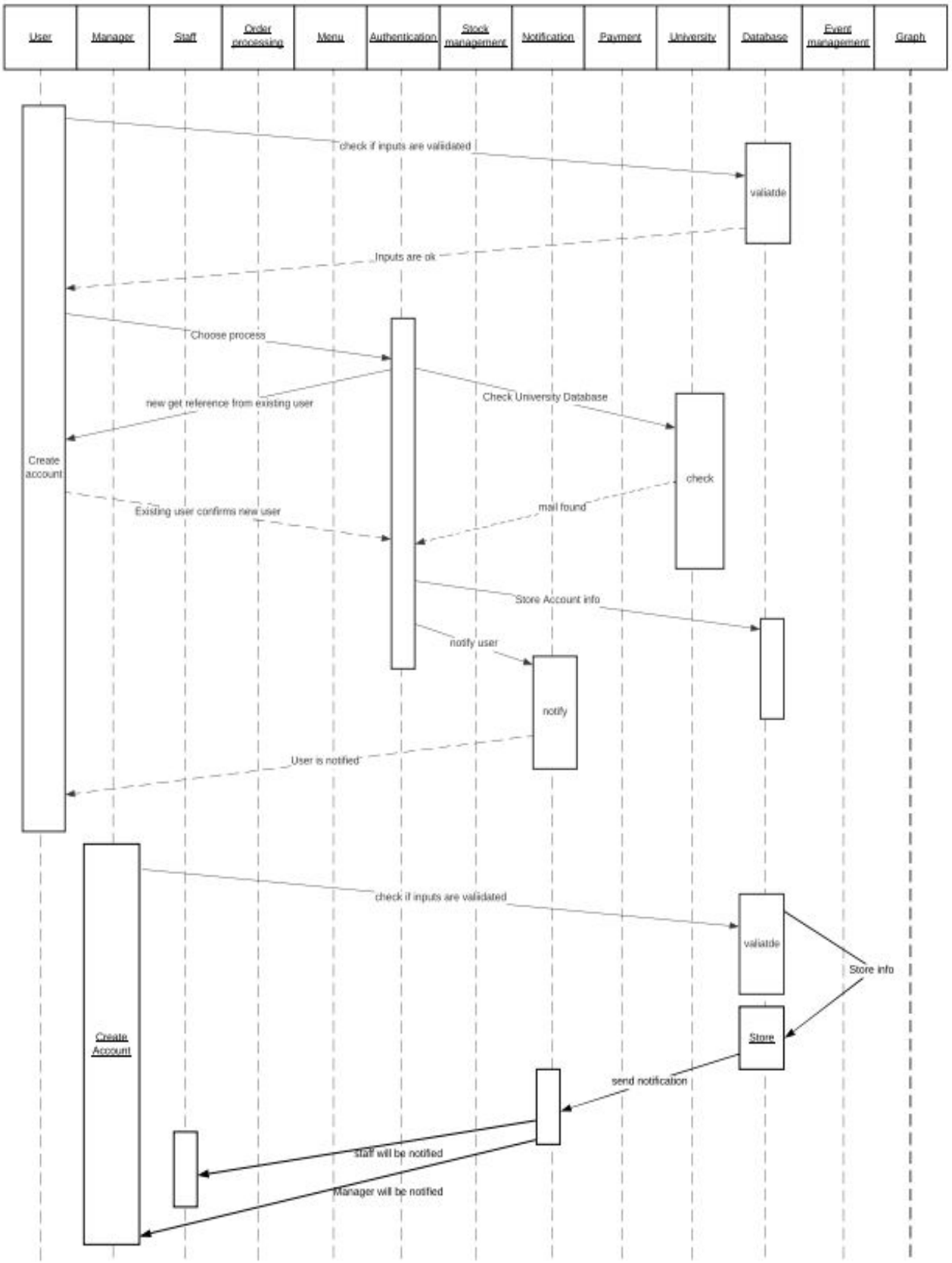
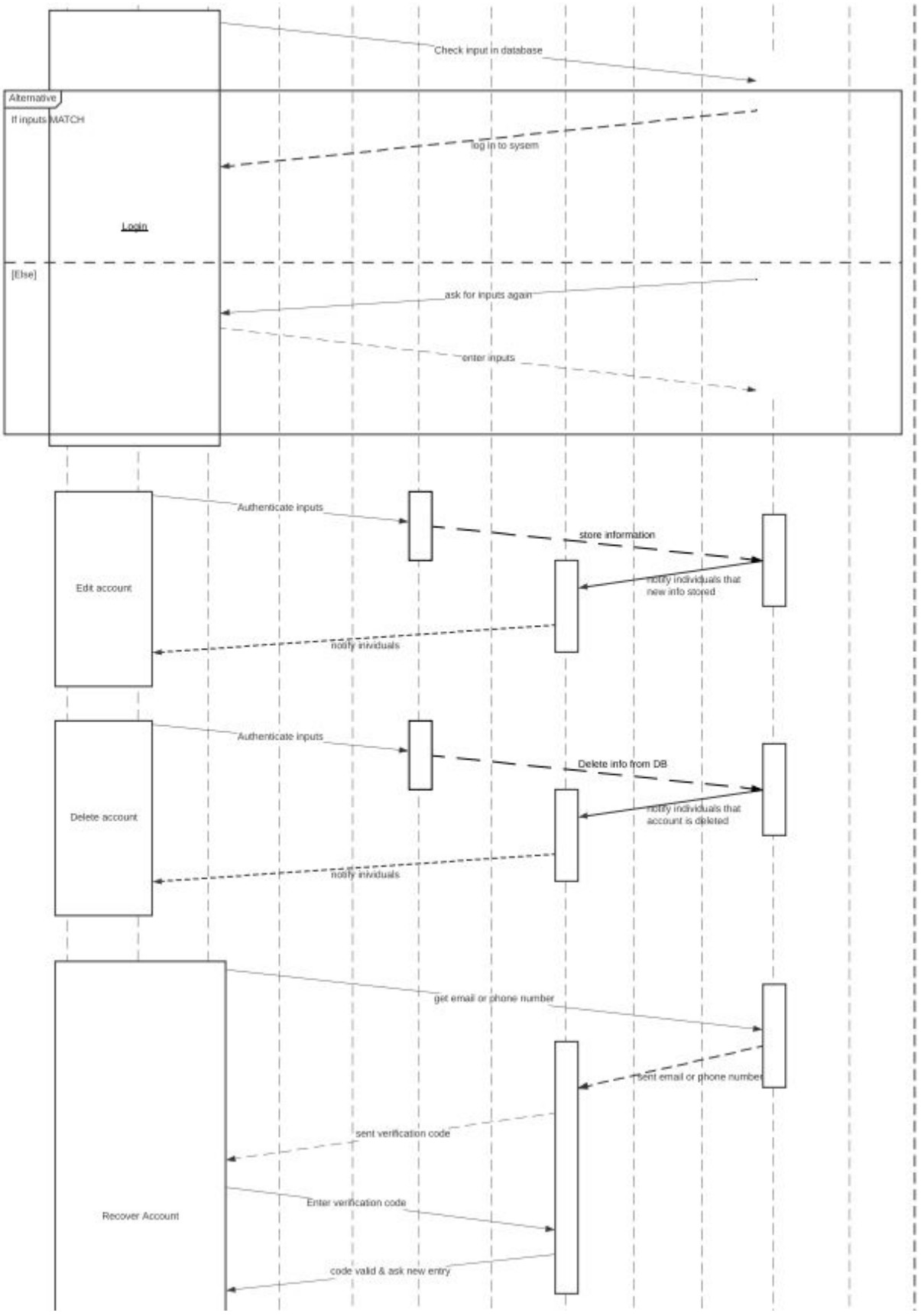
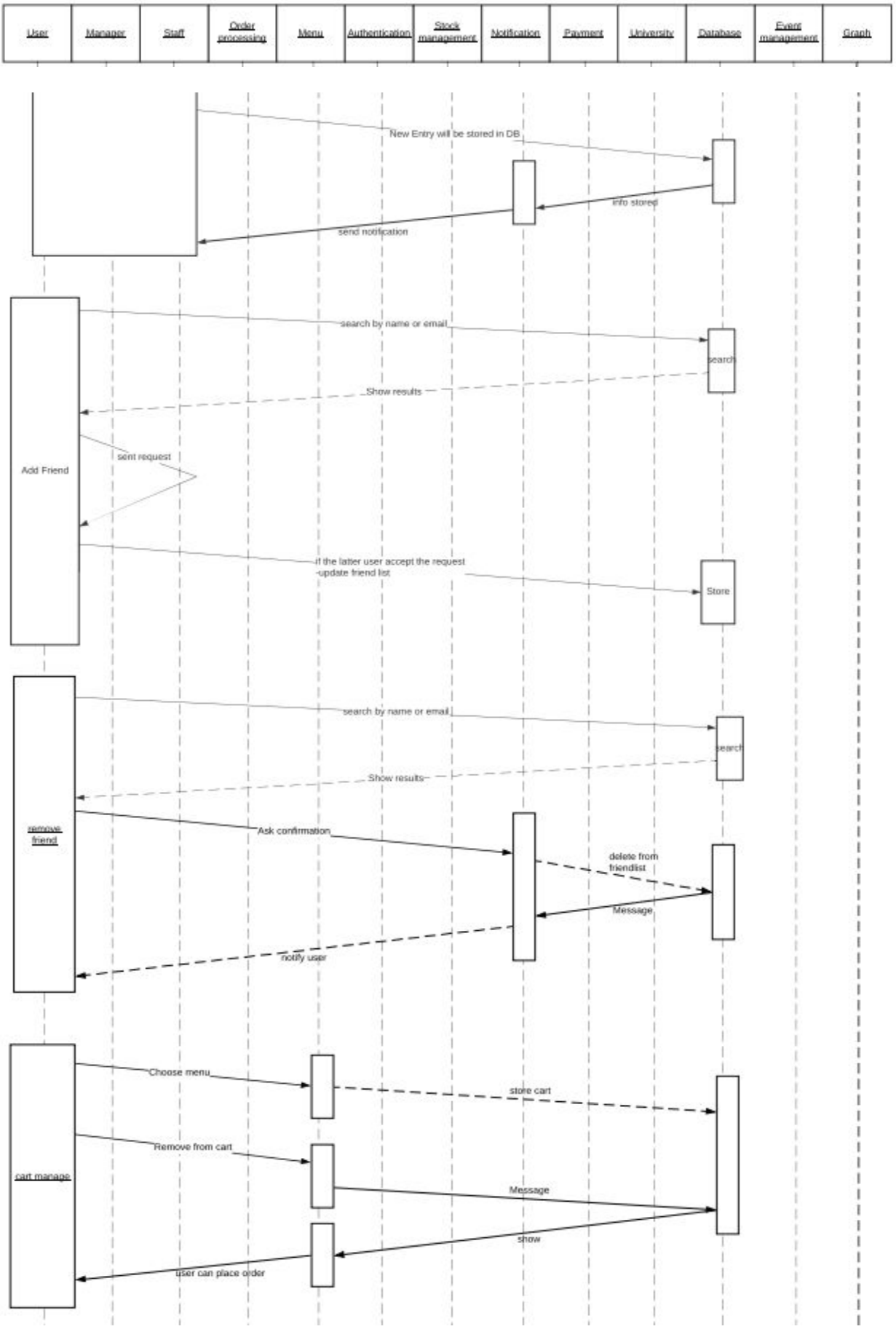


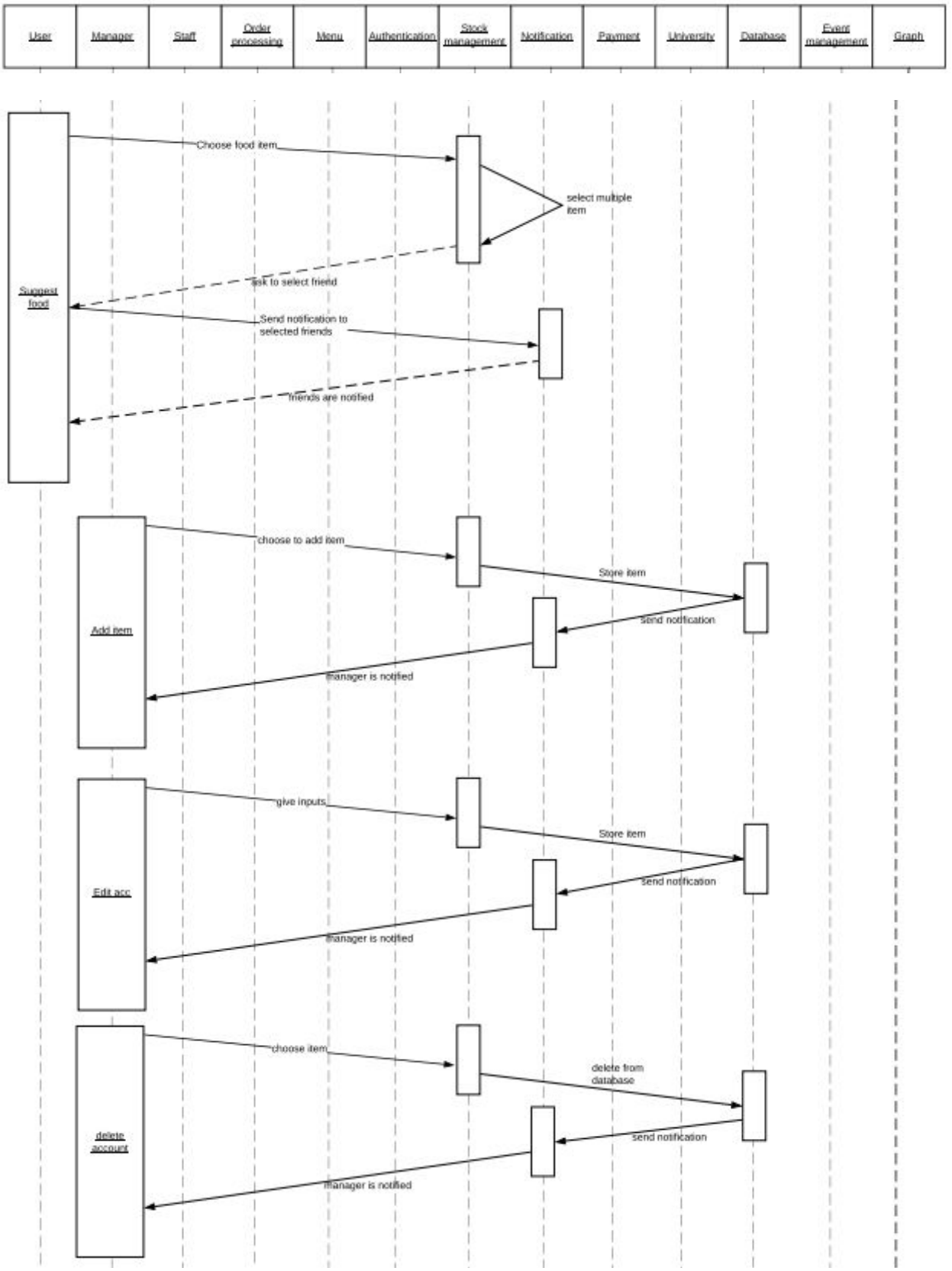
Figure 98: State Transition Diagram-University.


User
Manager
Staff
Order processing
Menu
Authentication
Stock management
Notification
Payment
University
Database
Event management
Graph
check if inputs are validated
valiatde
Inputs are ok
Choose process
new get reference from existing user
Check University Database
check
Create account
Existing user confirms new user
mail found
Store Account info
notify user
notify
User is notified
check if inputs are validated
valiatde
Store info
Store
Create Account
send notification
staff will be notified
Manager will be notified

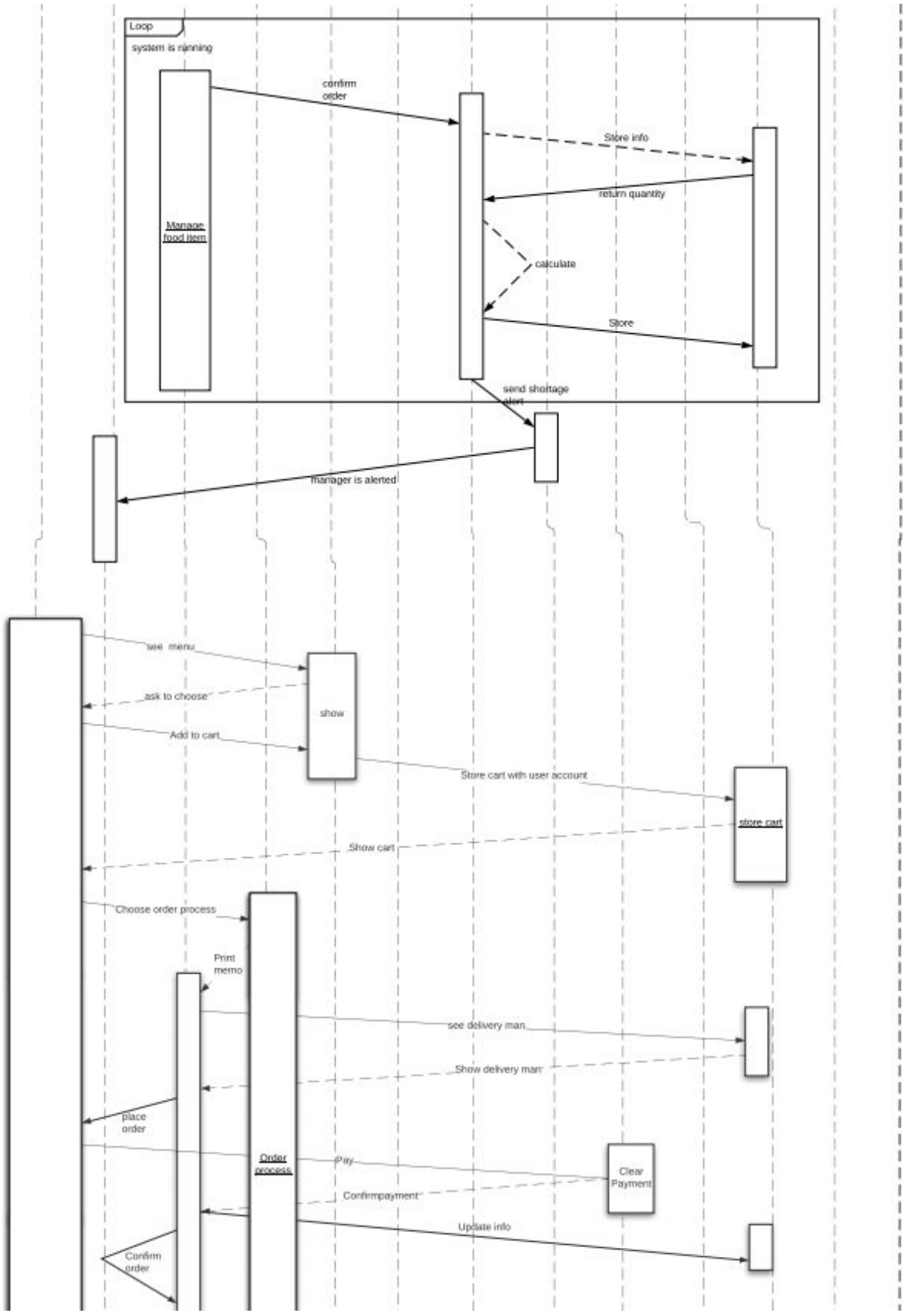

User
Manager
Staff
Order processing
Menu
Authentication
Stock management
Notification
Payment
University
Database
Event management
Graph

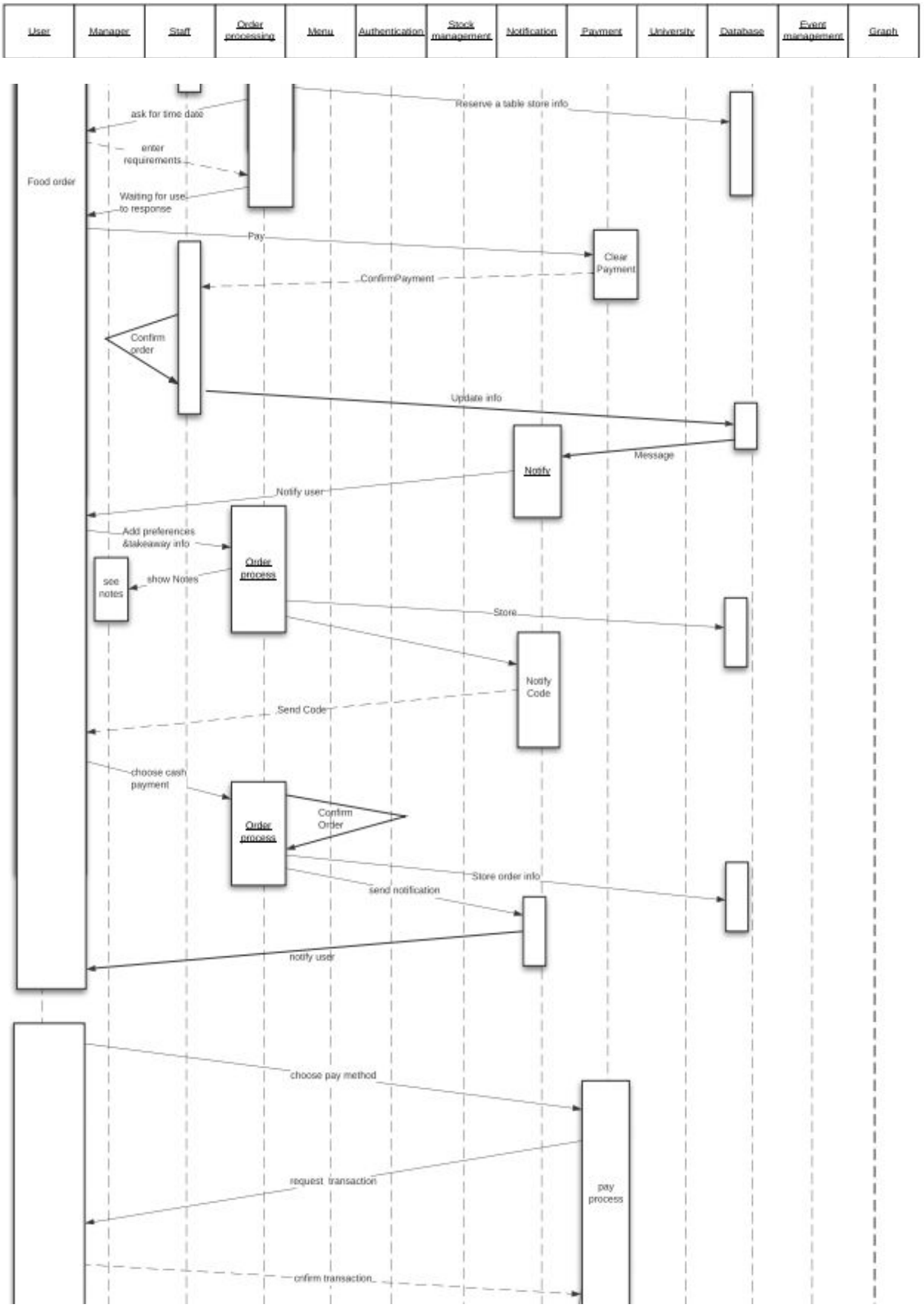

Check input in database
Alternative
If inputs MATCH
log in to sysem
Login
[Else]
ask for inputs again
enter inputs
Authenticate inputs
store information
Edit account
notify individuals that new info stored
notify individuals
Authenticate inputs
Delete info from DB
Delete account
notify individuals that account is deleted
notify individuals
get email or phone number
sent email or phone number
sent verification code
Recover Account
Enter verification code
code valid & ask new entry


User
Manager
Staff
Order processing
Menu
Authentication
Stock management
Notification
Payment
University
Database
Event management
Graph
New Entry will be stored in DB
info stored
send notification
search by name or email
search
Show results
Add Friend
sent request
if the latter user accept the request
update friend list
Store
search by name or email
search
Show results
remove friend
Ask confirmation
delete from friendlist
Message
notify user
Choose menu
store cart
Remove from cart
cart manage
Message
show
user can place order

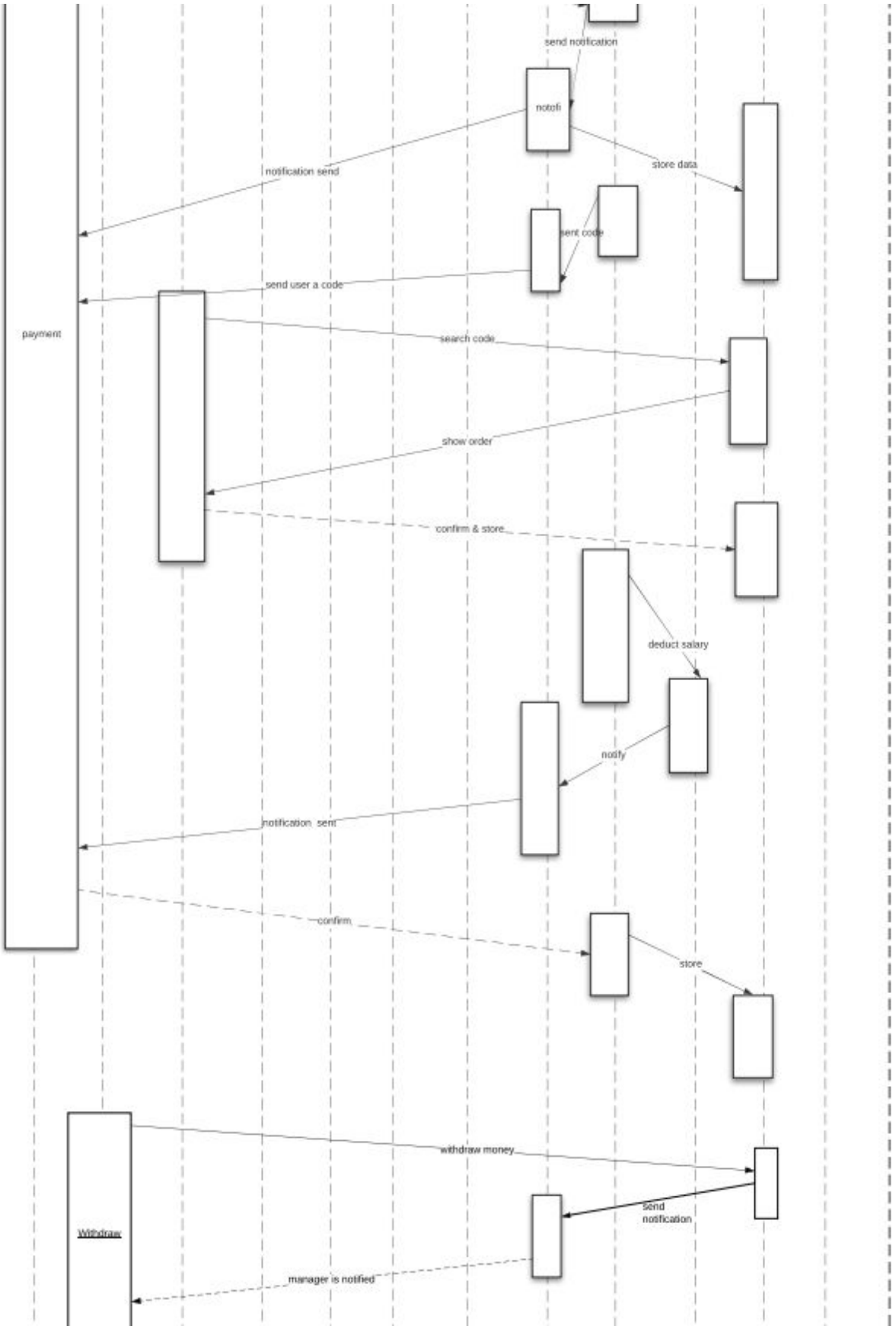

User
Manager
Staff
Order processing
Menu
Authentication
Stock management
Notification
Payment
University
Database
Event management
Graph
Choose food item
select multiple item
Suggest food
Ask to select friend
Send notification to selected friends
Friends are notified
Add item
choose to add item
Store item
send notification
Manager is notified
Edit acc
give inputs
Store item
send notification
Manager is notified
delete account
choose item
delete from database
send notification
Manager is notified


User
Manager
Staff
Order processing
Menu
Authentication
Stock management
Notification
Payment
University
Database
Event management
Graph

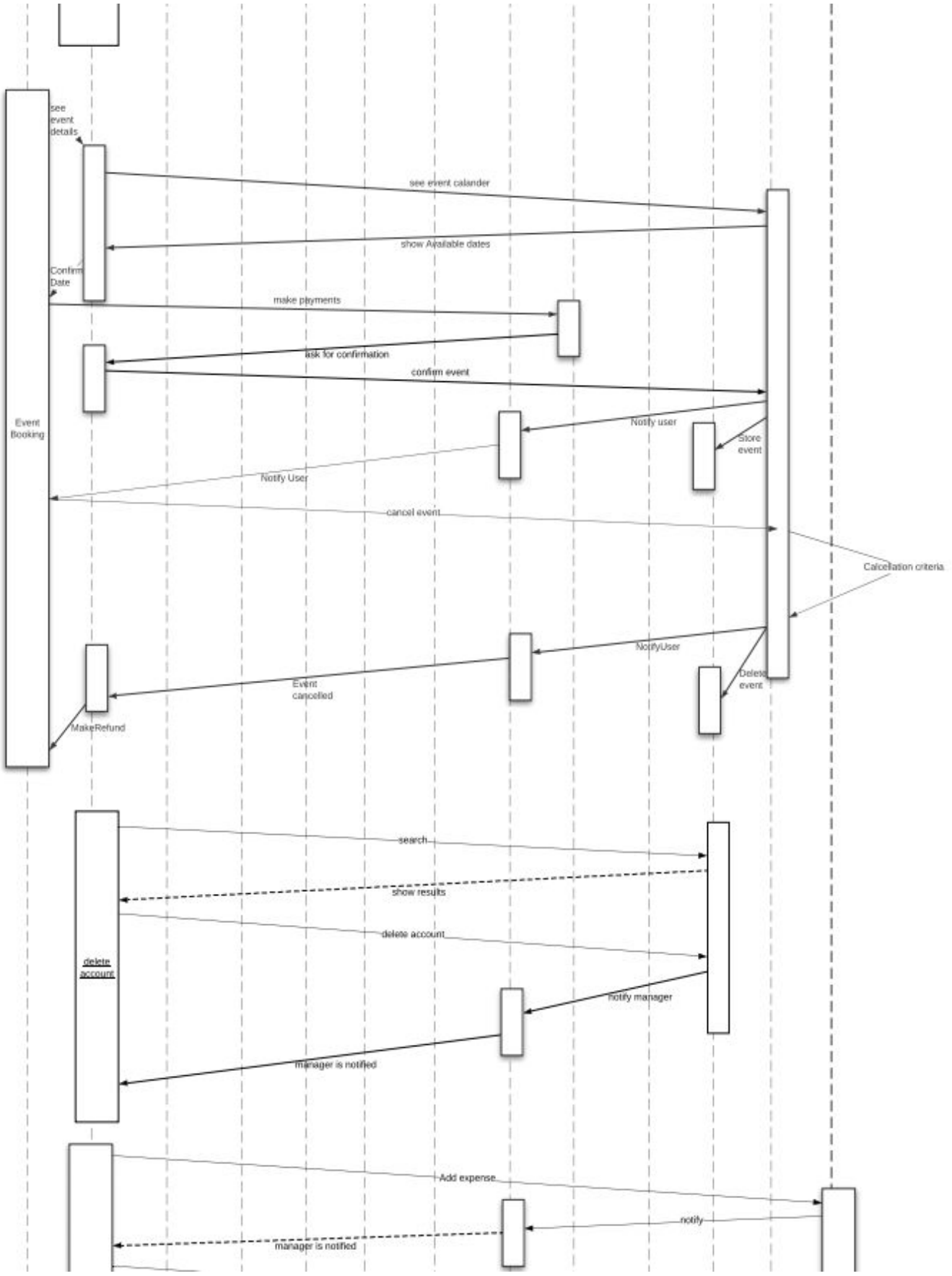

Loop
system is running
confirm order
Store info
return quantity
Manage food item
calculate
Store
send shortage alert
manager is alerted
see menu
ask to choose
Add to cart
show
Store cart with user account
store cart
Show cart
Choose order process
Print memo
see delivery man
Show delivery man
place order
Order process
Pay
Clear Payment
Confirmpayment
Update info
Confirm order

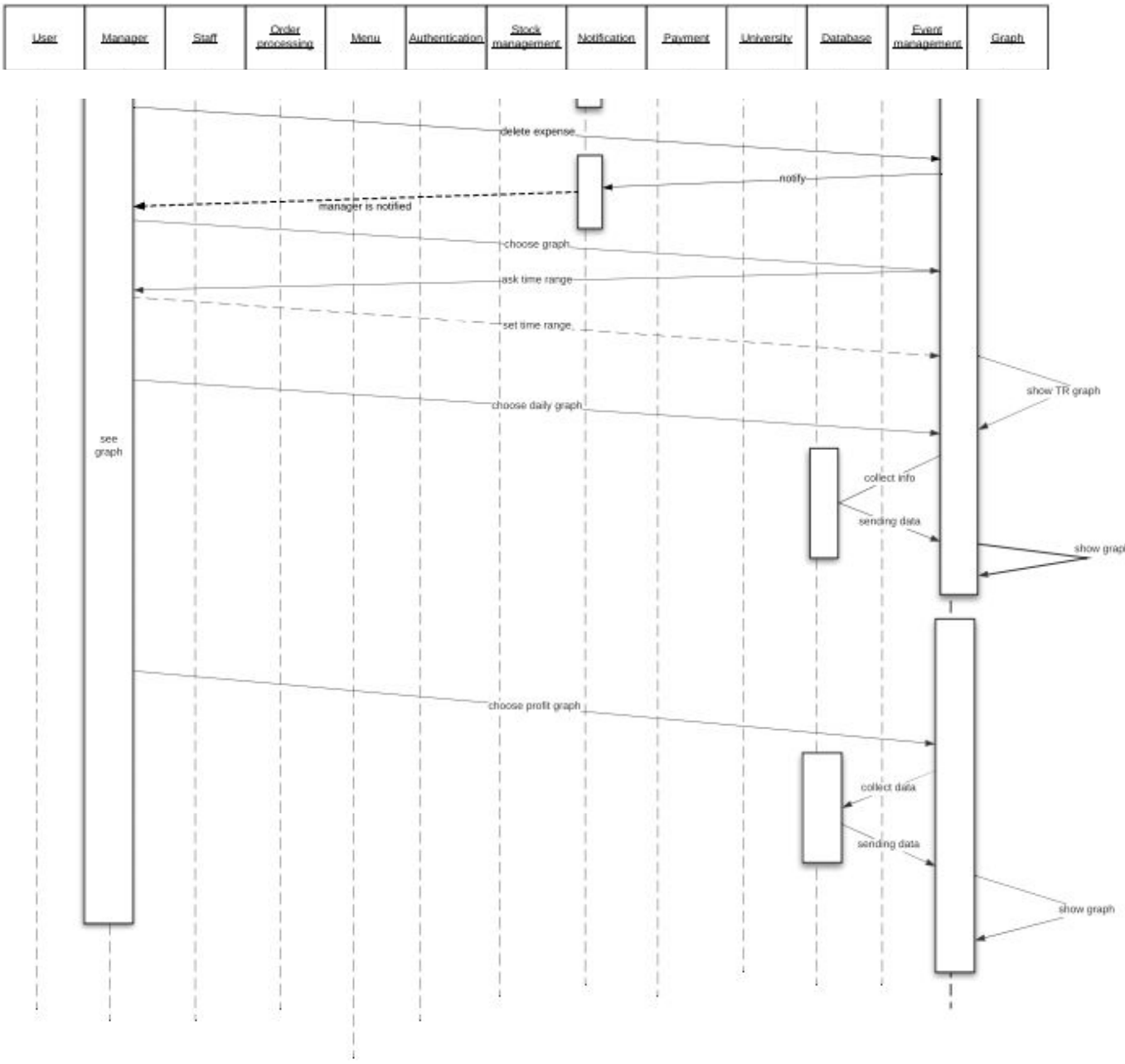

User
Manager
Staff
Order processing
Menu
Authentication
Stock management
Notification
Payment
University
Database
Event management
Graph
Food order
ask for time date
enter requirements
Waiting for user to response
Reserve a table store info
Pay
Clear Payment
ConfirmPayment
Confirm order
Update info
Message
Notify
Notify user
Add preferences &takeaway info
see notes
show Notes
Order process
Store
Notify Code
Send Code
choose cash payment
Order process
Confirm Order
Store order info
send notification
notify user
choose pay method
request transaction
pay process
cnfirm transaction

User	Manager	Staff	Order processing	Menu	Authentication	Stock management	Notification	Payment	University	Database	Event management	Graph


send notification
notofi
store data
notification send
sent code
send user a code
payment
search code
show order
confirm & store
deduct salary
notify
notification sent
confirm
store
withdraw money
Send
notification
Withdraw
manager is notified

User	Manager	Staff	Order processing	Menu	Authentication	Stock management	Notification	Payment	University	Database	Event management	Graph


see event details
see event calander
show Available dates
Confirm Date
make payments
ask for confirmation
confirm event
Event Booking
Notify user
Store event
Notify User
cancel event
Calcellation criteria
NotifyUser
Delete event
Event cancelled
MakeRefund
search
show results
delete account
delete account
notify manager
manager is notified
Add expense
notify
manager is notified


User
Manager
Staff
Order processing
Menu
Authentication
Stock management
Notification
Payment
University
Database
Event management
Graph
delete expense
notify
manager is notified
choose graph
ask time range
set time range
show TR graph
choose daily graph
see graph
collect info
sending data
show graph
choose profit graph
collect data
sending data
show graph

CHAPTER 9

CONCLUSION

We are pleased to submit the final SRS report on Cafeteria Management System. From this, the readers will get a clear and easy view of the overall system of small-scale cafeteria. This SRS document can be used effectively to maintain the software development cycle. It will be very easy to conduct the whole project using this SRS. Hopefully, this document can also help our junior BSSE batch students. We tried our best to remove all dependencies and make an effective and fully designed SRS. We believe that the reader will find it in order.

CHAPTER 10

REFERENCES

Pressman, Roger S. Software Engineering: A Practitioner's Approach (7th Edition)