

CANTEEN MANAGEMENT SYSTEM

UCS503 Software Engineering Project Report

Final Lab Evaluation

Submitted By:

101803346 SHREY SAWHNEY

101803703 VARUN AGGARWAL

101803705 TANISHQ BHALLA

BE Third Year, COE

Submitted To

Dr. V. K. Bhalla

Assistant Professor



Computer Science and Engineering Department

TIET, Patiala

November 2020

TABLE OF CONTENTS

TABLE OF CONTENTS	2
1. PROJECT OVERVIEW	3
2. PROJECT REQUIREMENTS	4
2.1. FUNCTIONAL REQUIREMENTS	4
2.2. NON-FUNCTIONAL REQUIREMENTS	5
3. STRUCTURED ANALYSIS	6
3.1. DATA FLOW DIAGRAMS	6
3.1.1. DFD LEVEL 0	6
3.1.2. DFD LEVEL 1	7
3.1.3. DFD LEVEL 2	8
4. OBJECT ORIENTED ANALYSIS	10
4.1. USE CASE DIAGRAM	10
4.1.1. CONTEXT LEVEL USE CASE DIAGRAM	10
4.1.2. LEVEL 2 USE CASE DIAGRAM	11
4.2. SEQUENCE DIAGRAM	14
4.3. CLASS DIAGRAM	19

1. PROJECT OVERVIEW

A canteen is a necessity for Offices, Factories, Call Centres, Hostels, Schools, Clubs and Hospitals for their employees and customers. In these organizations, entire mess management and billing calculations are done manually till date. It is very time consuming and increases the chances of performing calculation mistakes. Many times, customers reach the cafeteria only to find the item they wanted is unavailable. Also, at times due to unavailability of change at time of transaction, the manager often writes down the balance in a register manually. Possibility of false orders also exists in the current receipt system. Customer satisfaction is also less.

A Canteen Management System (CMS) provides a user-friendly system that facilitates quick and efficient operations to complete any canteen related task such as placing an order, payment, etc. This prevents queues and reduces cafeteria transaction time. In order to explain all aspects of the project we have listed below all that is being done by and for the involved parties in this Canteen Management System:

- **Customer** - A customer can see the live count of the menu items on a mobile app and can place orders directly through the app via e-payment. The customer can view their previous orders and the balance left in their digital wallet. They can also share feedback to the canteen management staff.
- **Mess Manager** - The mess manager can view, edit, add and remove items from the system through a desktop application. The manager can view feedback received from customers and also send announcements and/or messages/notifications back. Daily reports can also be generated. He/she can manually edit the wallets of any customer in the system.
- **Mess Workers** - They can view the list of present orders and mark them as completed, ready to be taken or preparing. The system will send a notification to the customer when their order is ready. They can also edit the global item count.
- **Admin** - The admins will receive the database of all the registered users (customers, mess managers or mess workers) along with all the item information and count. The admin can view, edit or delete any record from the system.

Hence, our Canteen Management System is a completely automated software that facilitates paperless order management, quick transactions and report generation. The customers now don't need to visit the canteen unnecessarily hence, reducing the crowd. In case of unavailability of change for cash payments, our system also provides a quick solution in the form of a digital wallet. With automatic transfer/visibility of orders in the kitchen after payment, the communication need between the cashier and kitchen staff is minimised.

2. PROJECT REQUIREMENTS

2.1. Functional requirements

1. Login -

1. Customers will login in their mobile app.
2. Mess Manager will login in their desktop application.
3. Admin login.

2. Order Placement - Orders can be placed by customers through mobile app or through the mess manager. The customer must give their details while placing the order such as customer ID, etc.

3. Payment - Payment must be done during placement of order by customer either digitally through mobile app or via cash for in-person orders.

4. Realtime Item Updates - The mess manager can update item count, add new items or remove existing items from the database and modify other item details which will be synced to the customers' mobile app. Mess workers can also update item quantities.

5. Current Kitchen Orders - The mess worker can change the current orders' status from 'Incomplete' to 'Complete' or 'Ready to be taken' via their desktop app.

6. Add/Remove/Modify Customer - Admin must be able to add/remove customers or modify already existing customer details in the database.

7. Wallet - In case of cash payments, if there is any remaining or surplus balance, it is maintained in the Customer's Wallet in the database only by the mess manager.

8. Sending Announcements and Messages – The mess manager can send announcements to every customer or messages to some students as needed.

9. **Additional notes in orders** - Displaying additional notes/information while placing orders.
10. **Saving customer's favorite Orders** - Displaying pre-saved combos to the Customer in their mobile app.
11. **Recommendations to customer** - Showing recommendations to the Customer in their mobile app.

2.2. Non-Functional Requirements

1. **Availability** - The system should be available at predetermined fixed time intervals every day.
2. **Reliability** - In case of database failure or corruption a replacement page will be shown telling them to visit the mess to order.
3. **Speed and Reliability**- Latency and delays will be minimised.
4. **Correctness** - As soon as someone presses confirm order and is taken to the payments page, the items are deducted from the database. So, even if someone had added items to their cart but are now out of stock, the person won't be able to order it now. This has been done to maintain proper records of item availability.

3. STRUCTURED ANALYSIS

3.1. Data Flow Diagrams

3.1.1. DFD Level 0

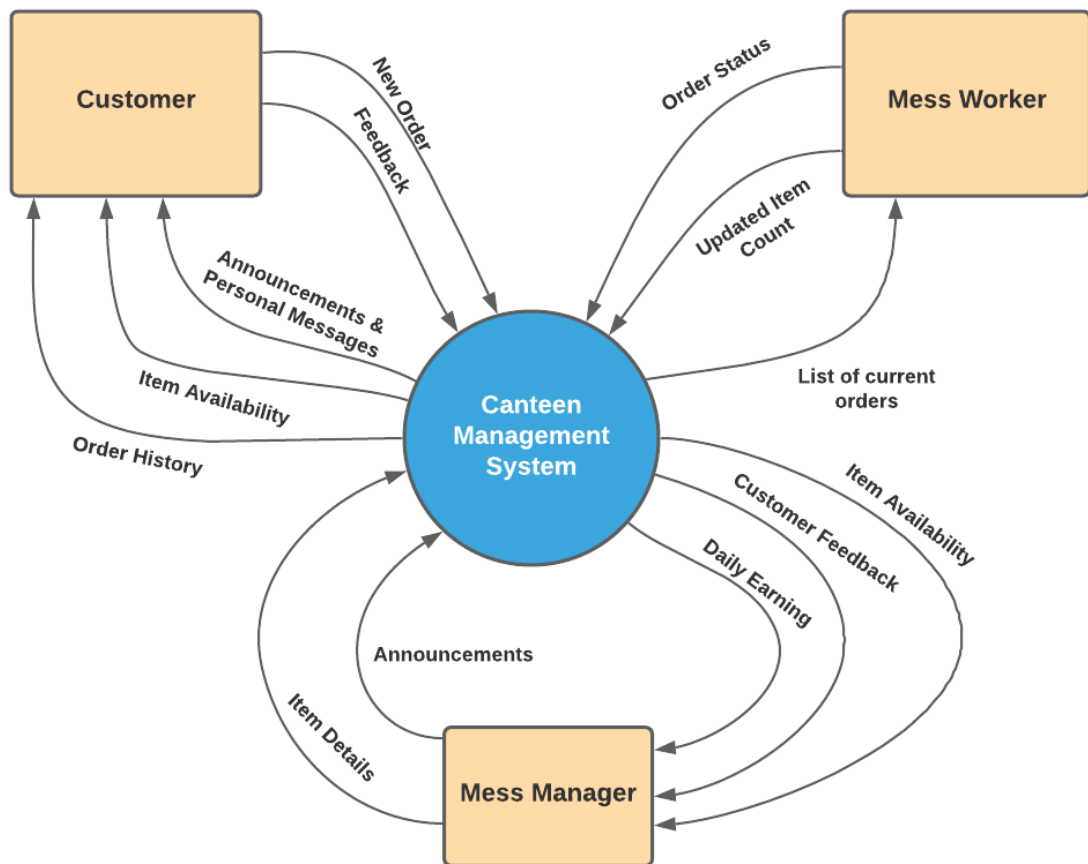


Figure 1: Context Level Diagram

3.1.2. DFD Level 1

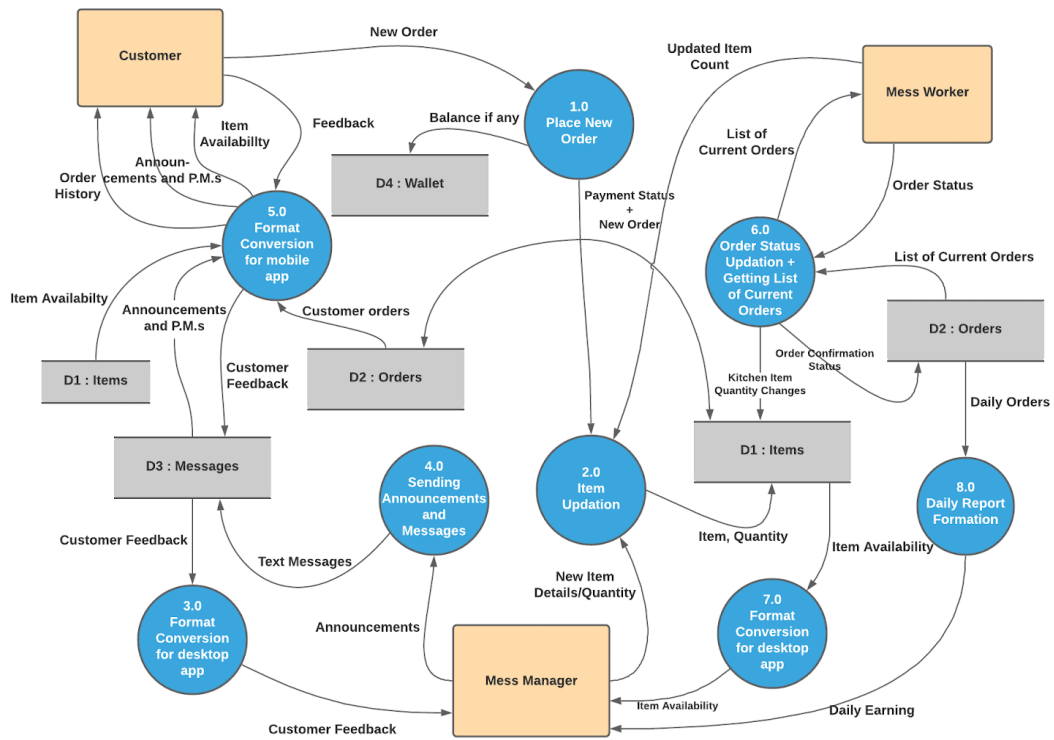
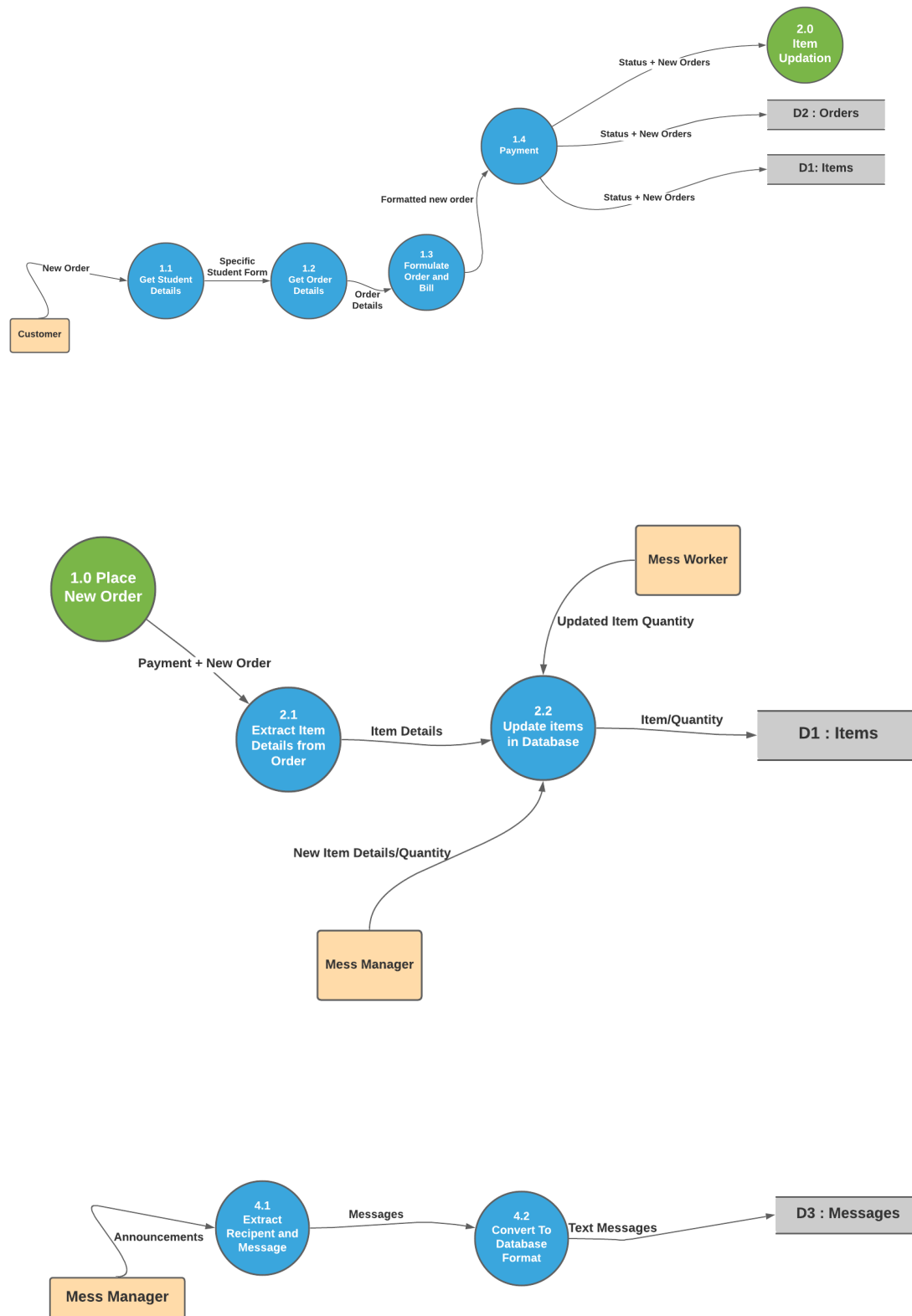


Figure 2: Level 1 DFD

3.1.3. DFD Level 2



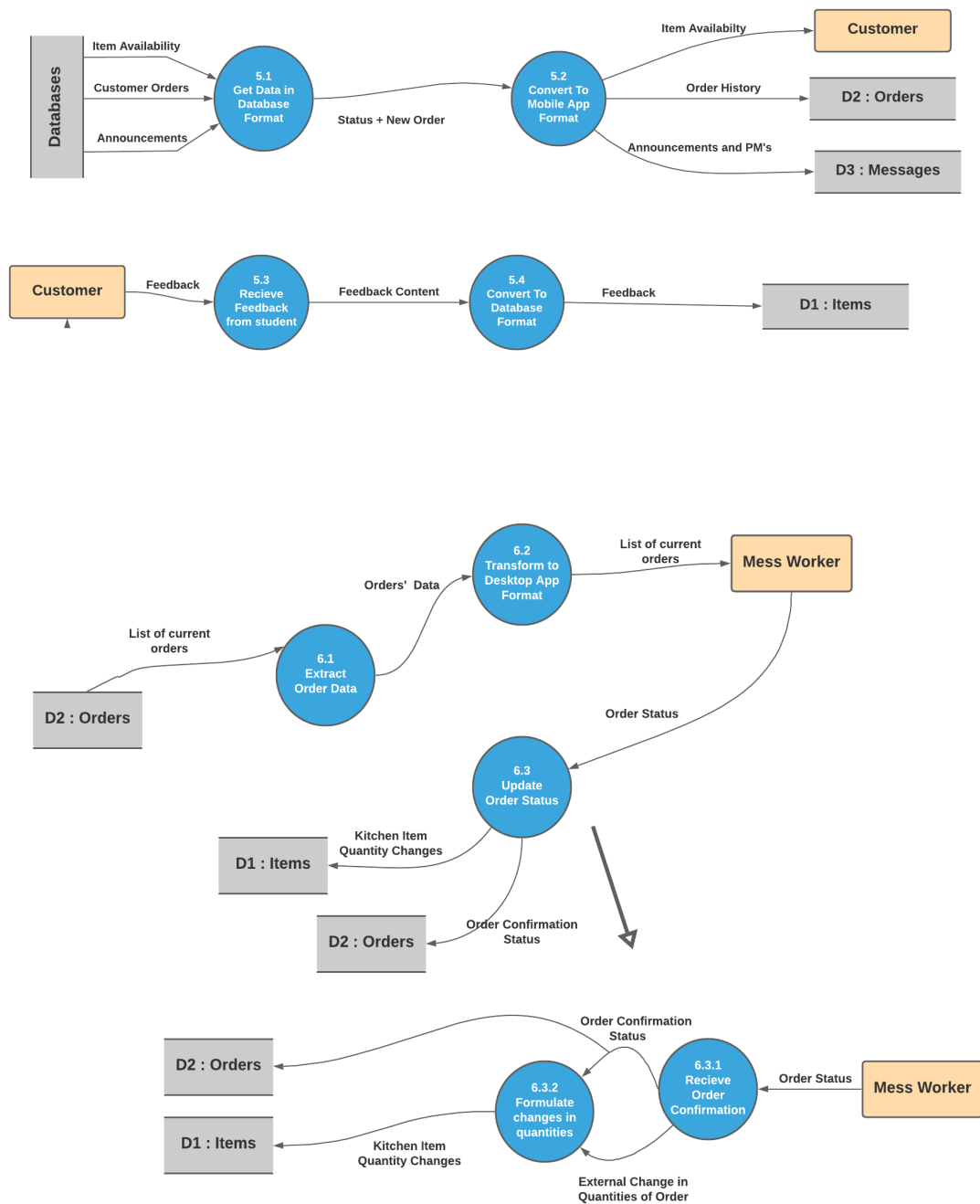


Figure 3: Level 2 DFD

4. OBJECT ORIENTED ANALYSIS

4.1. Use Case Diagram

4.1.1. Context Level Use Case Diagram

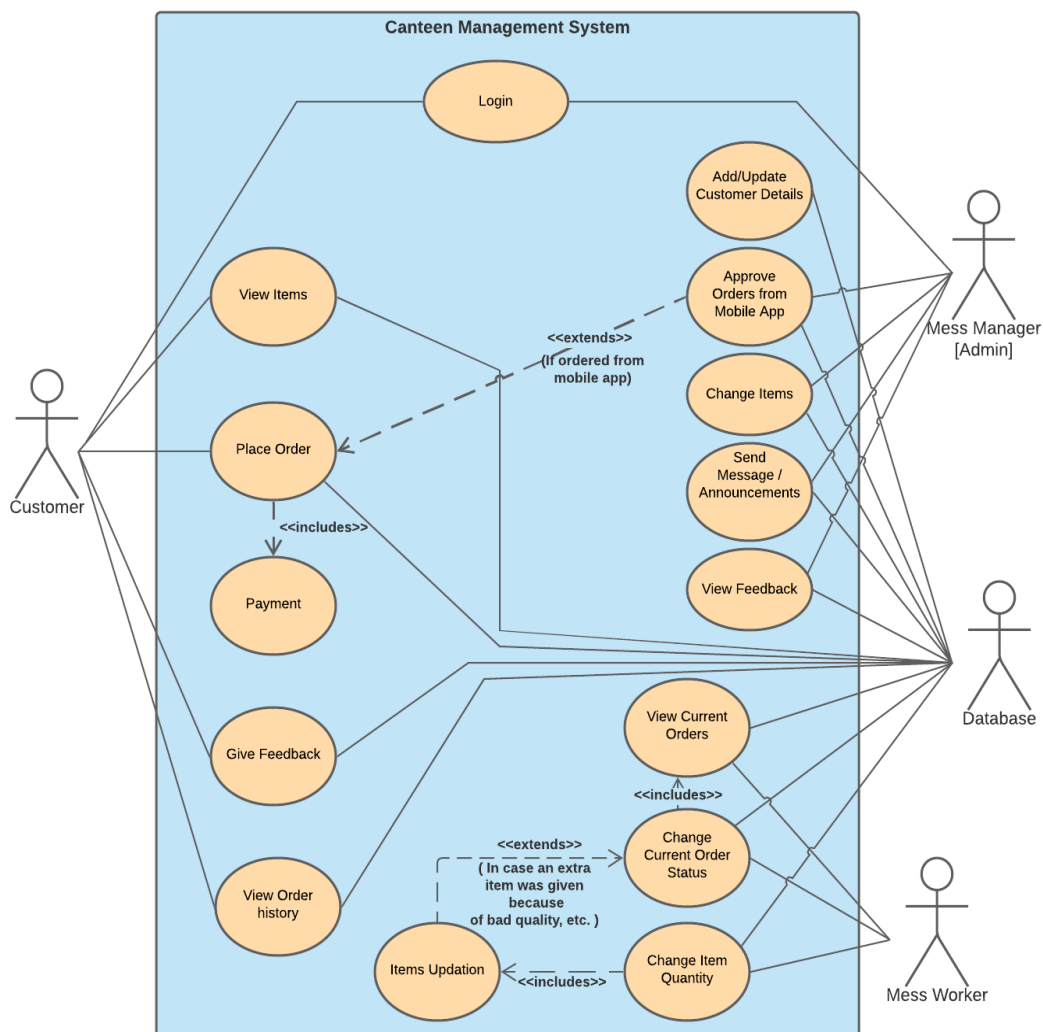
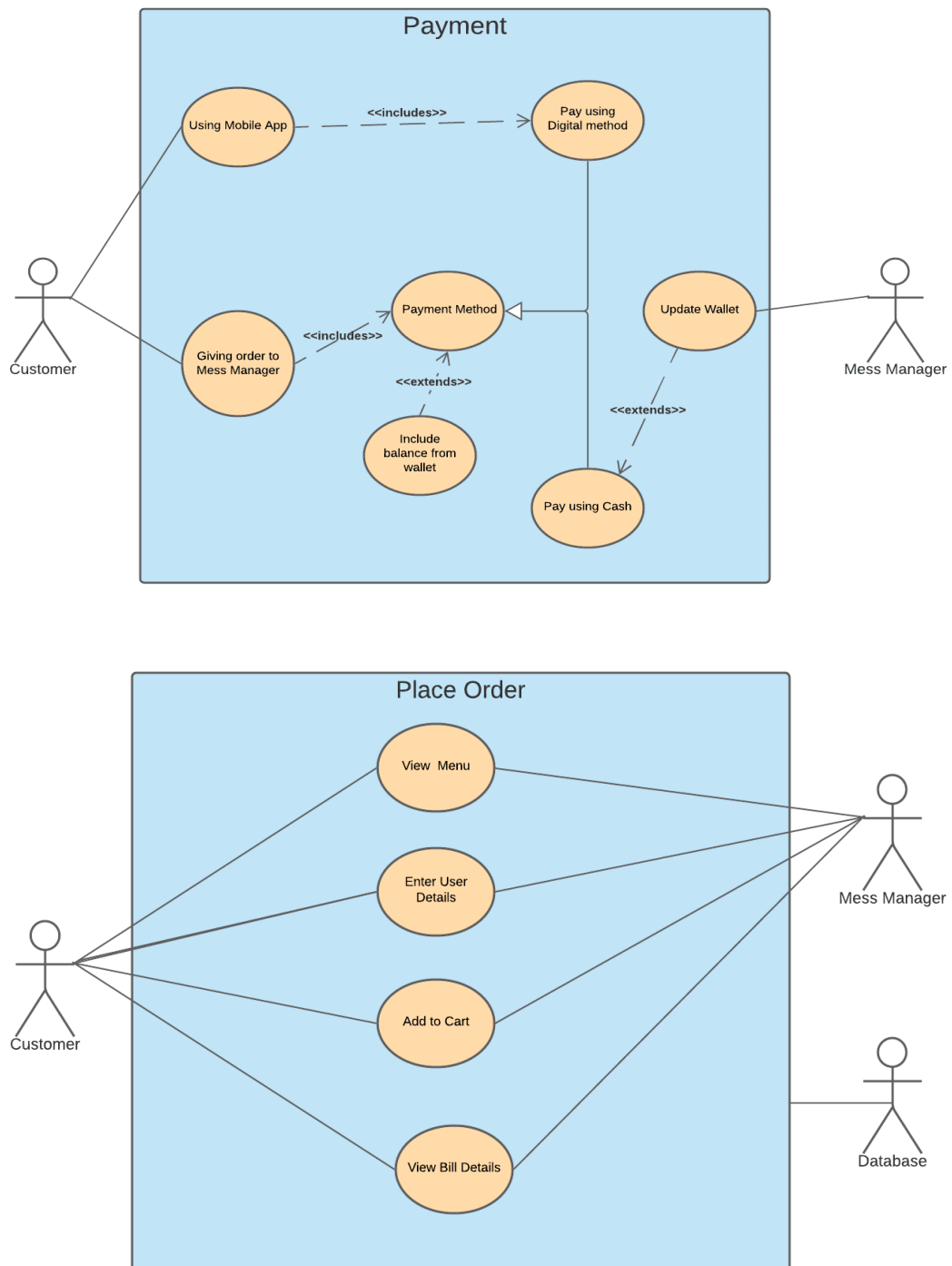
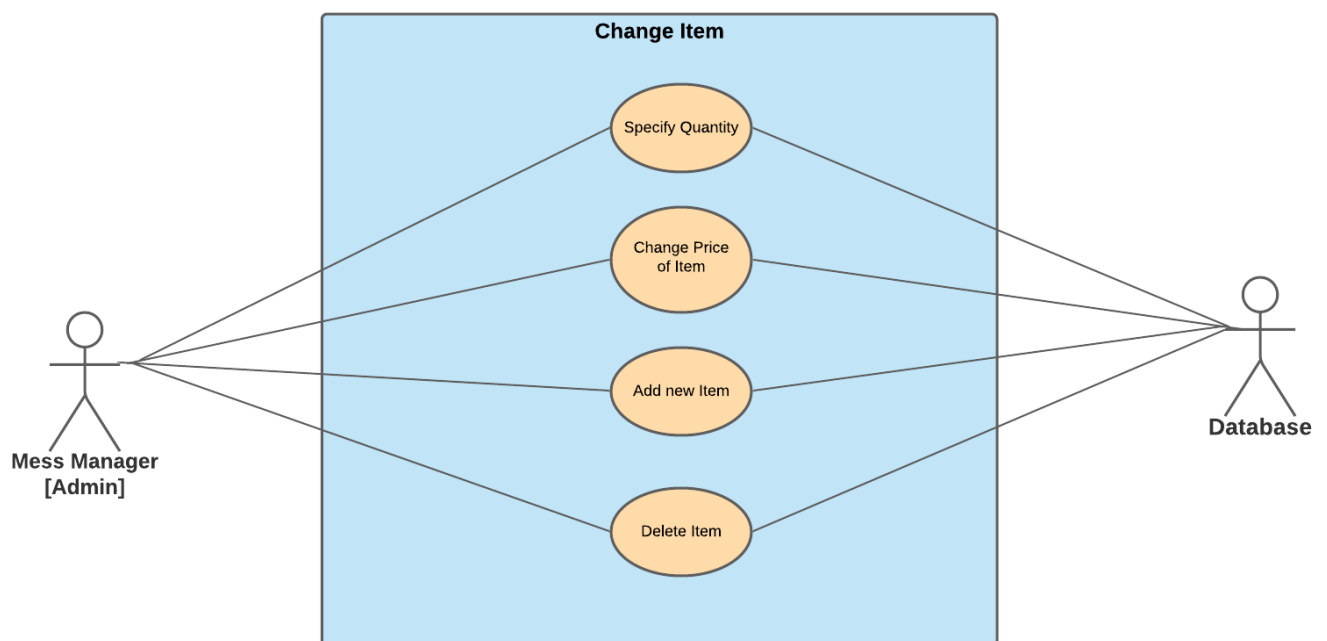
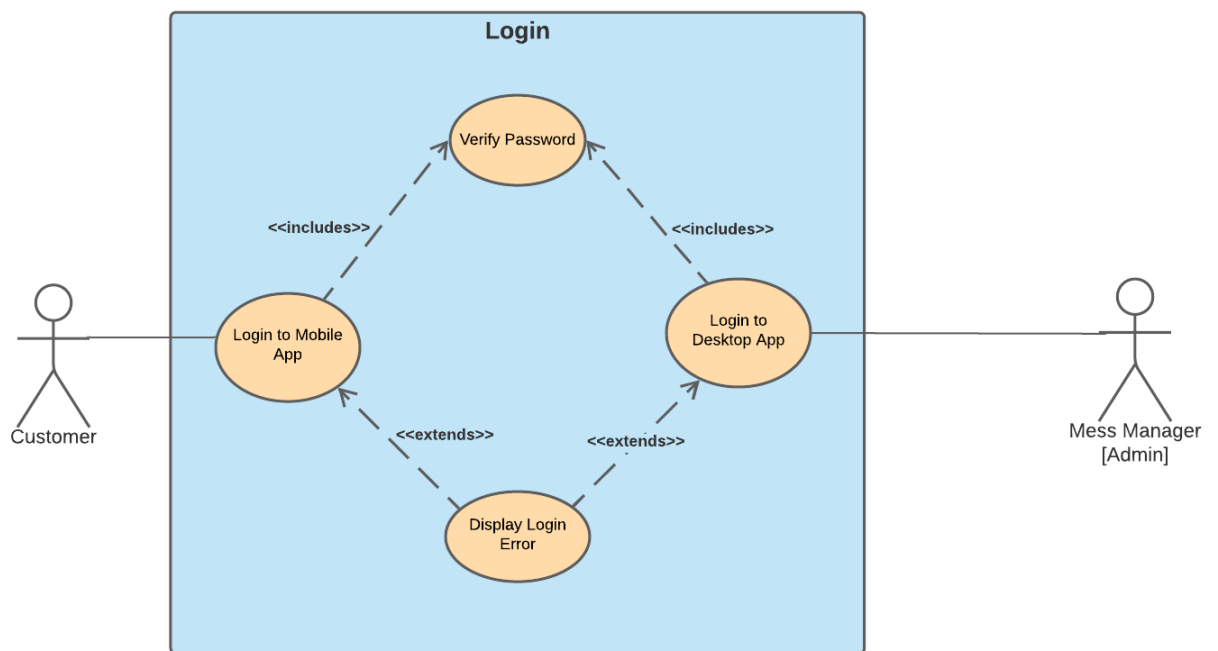


Figure 4: Context Use case diagram

4.1.2. Level 2 Use Case Diagram





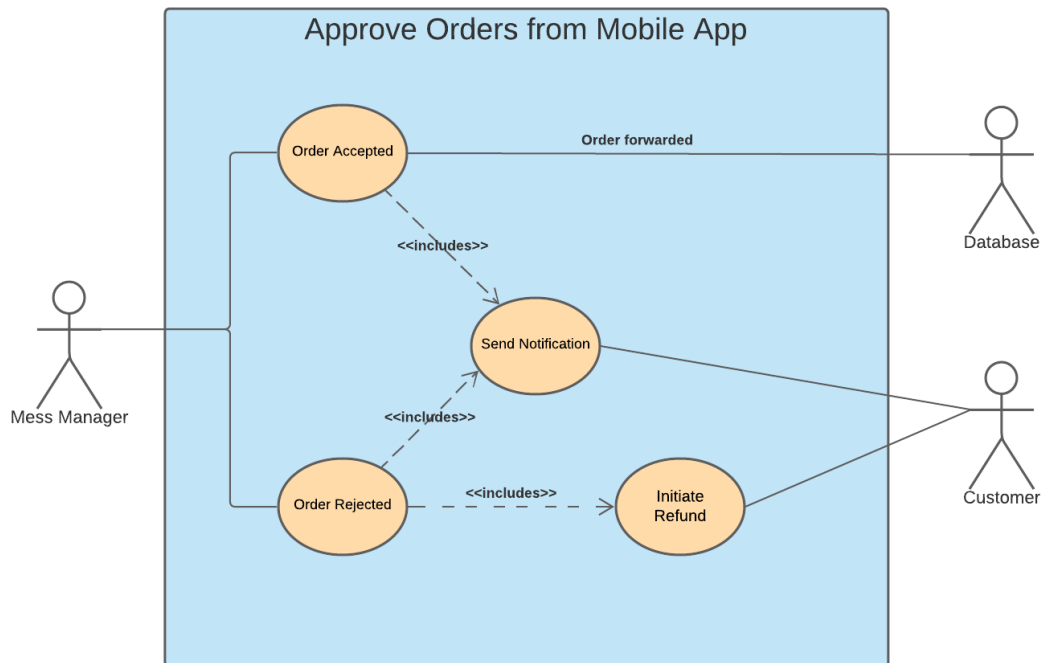
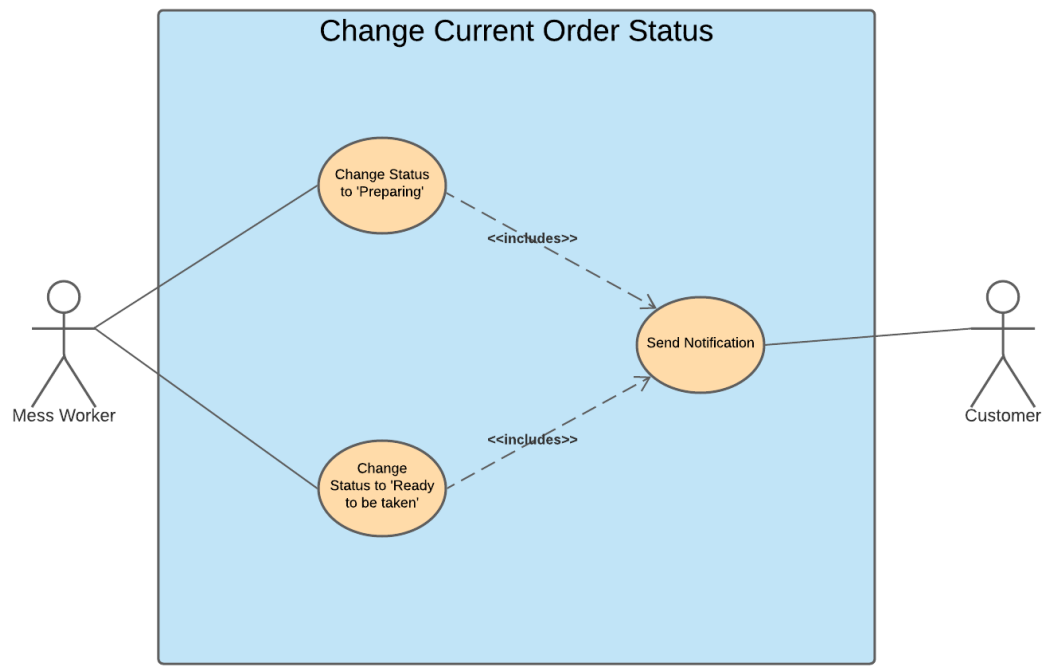
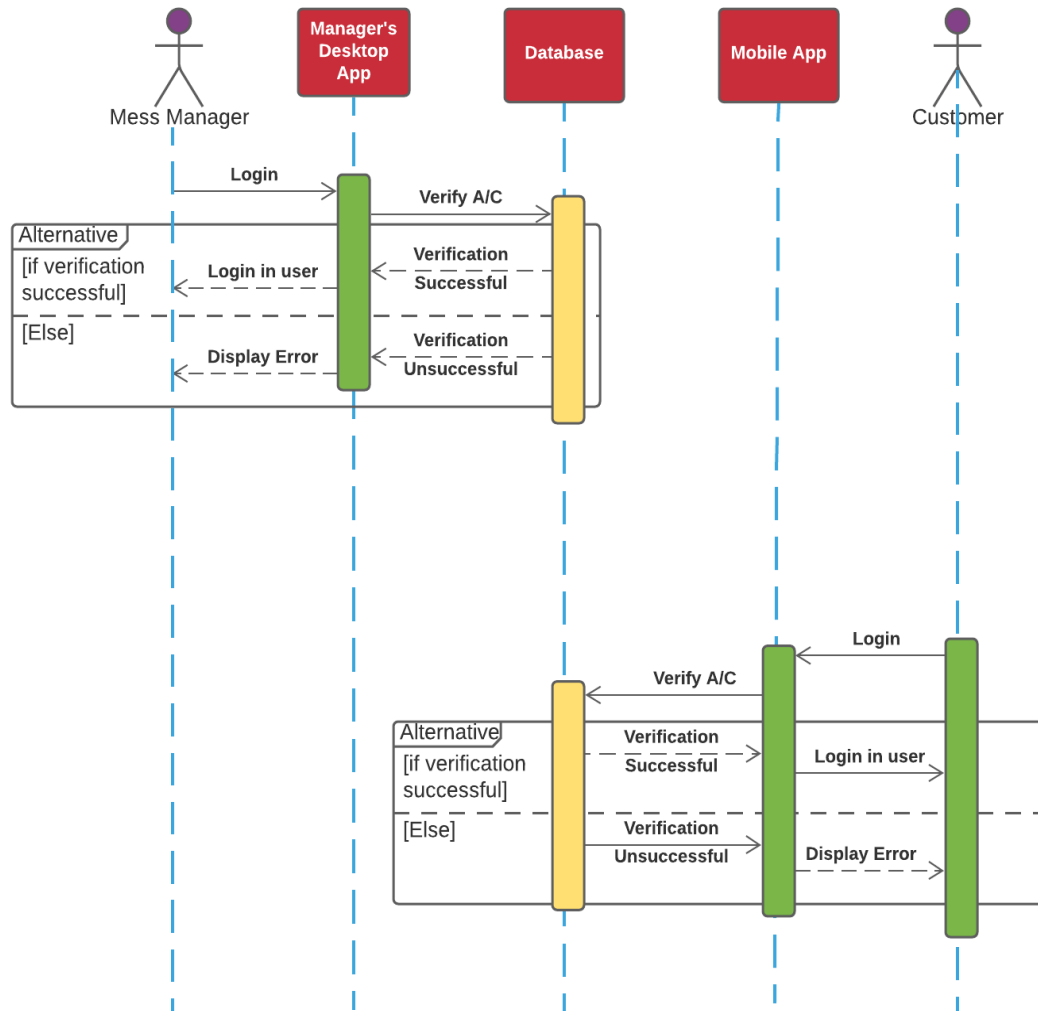


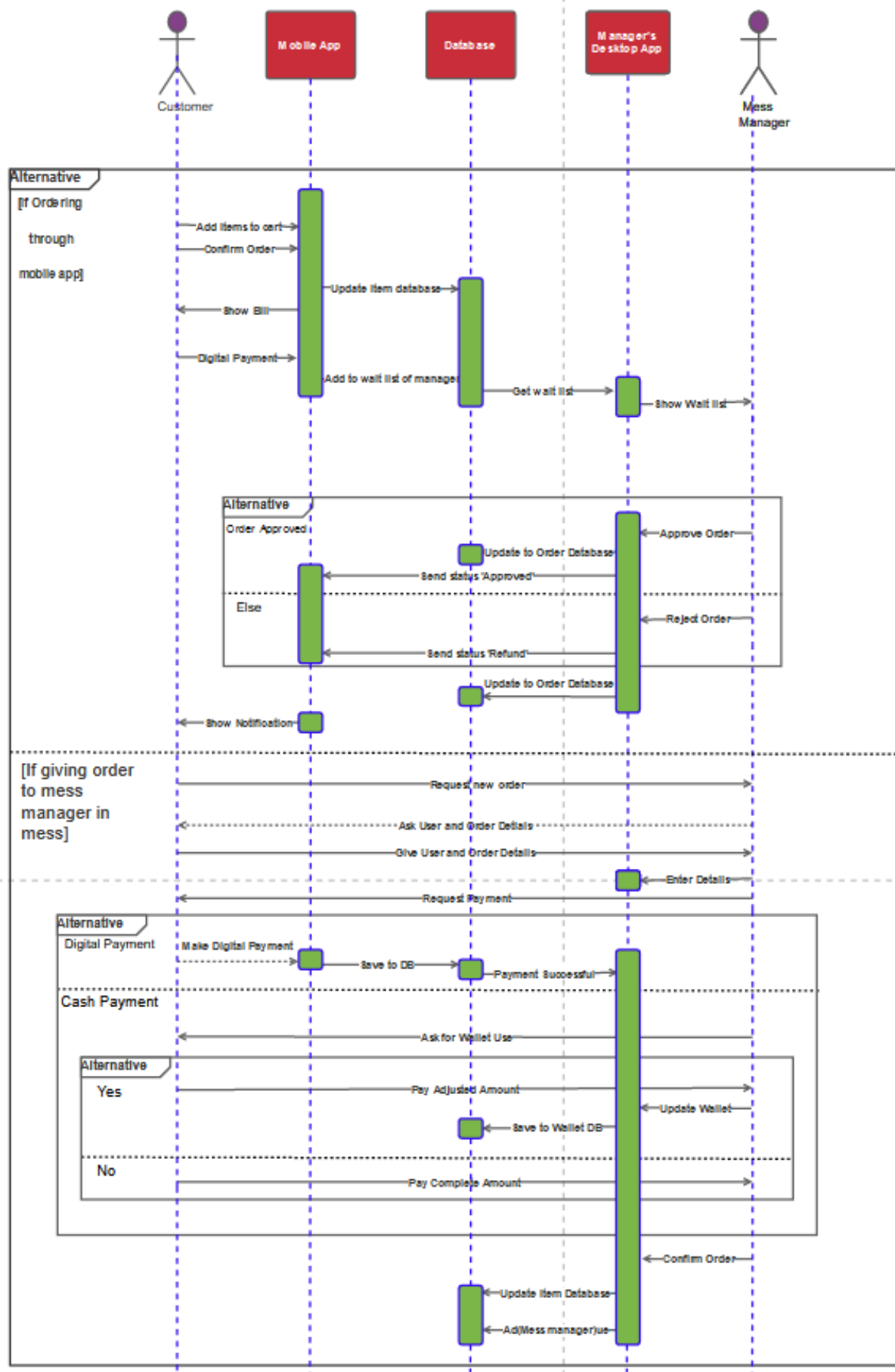
Figure 5: Level 2 Use case Diagram

4.2. Sequence Diagram

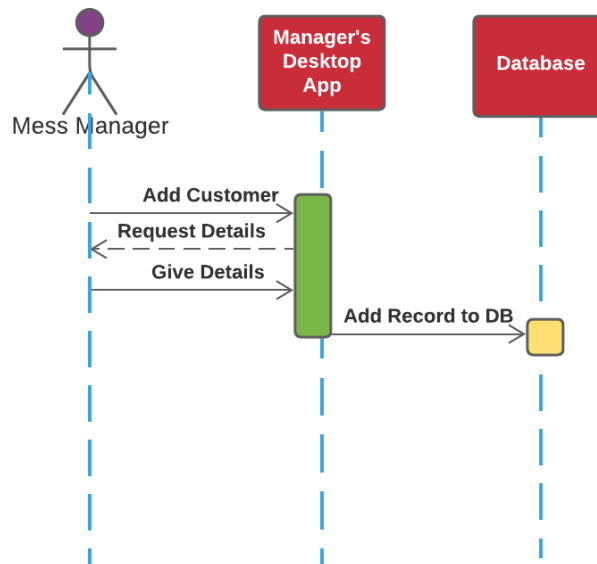
Login



Place Order

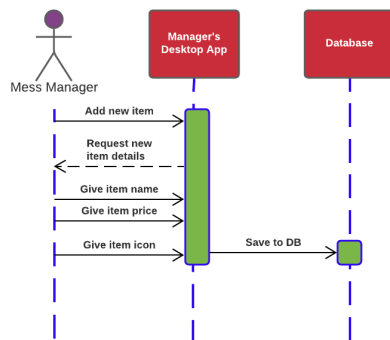


Add Customer

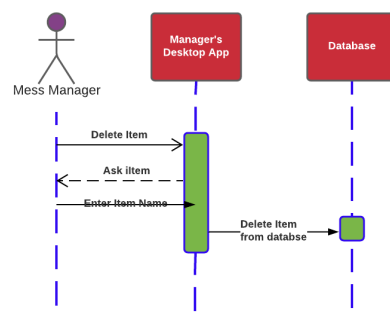


Change Item

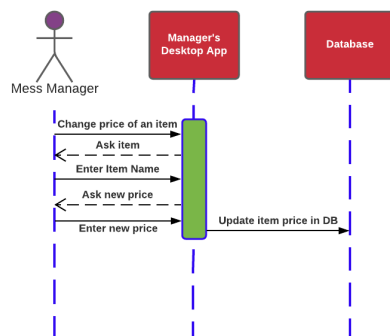
Add new item



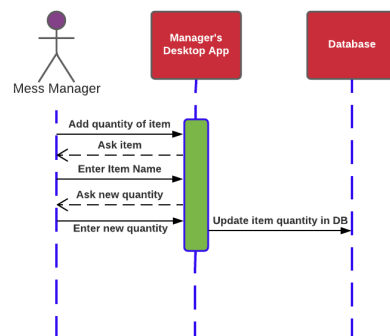
Delete Item



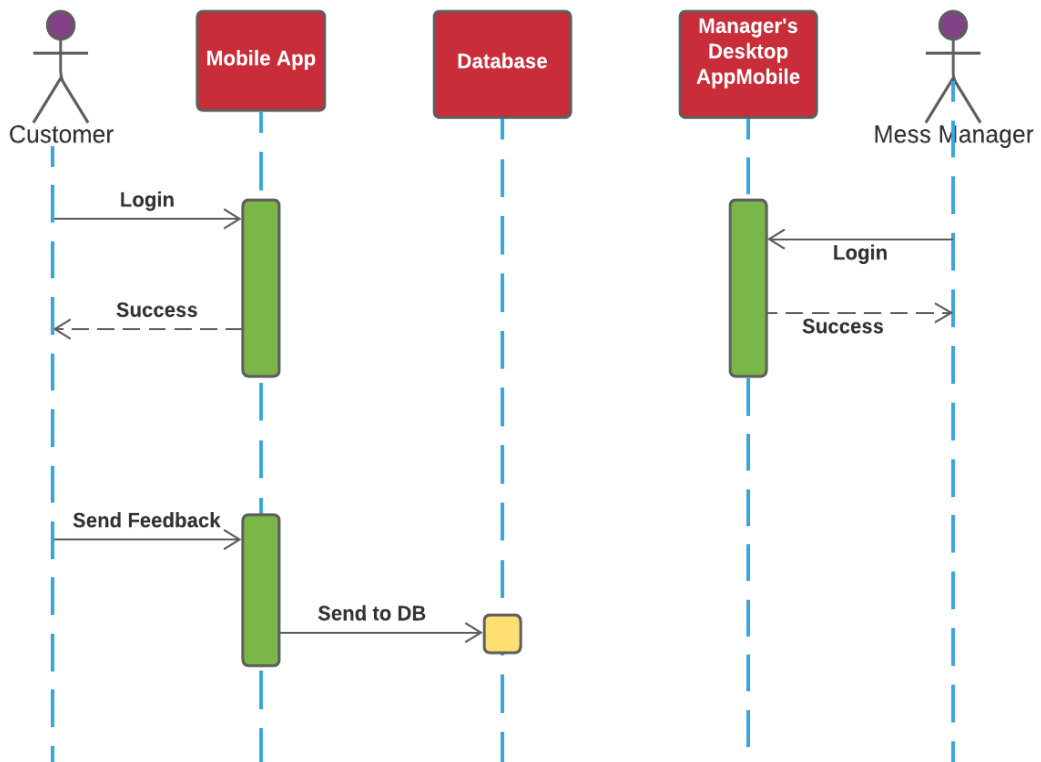
Change price of item



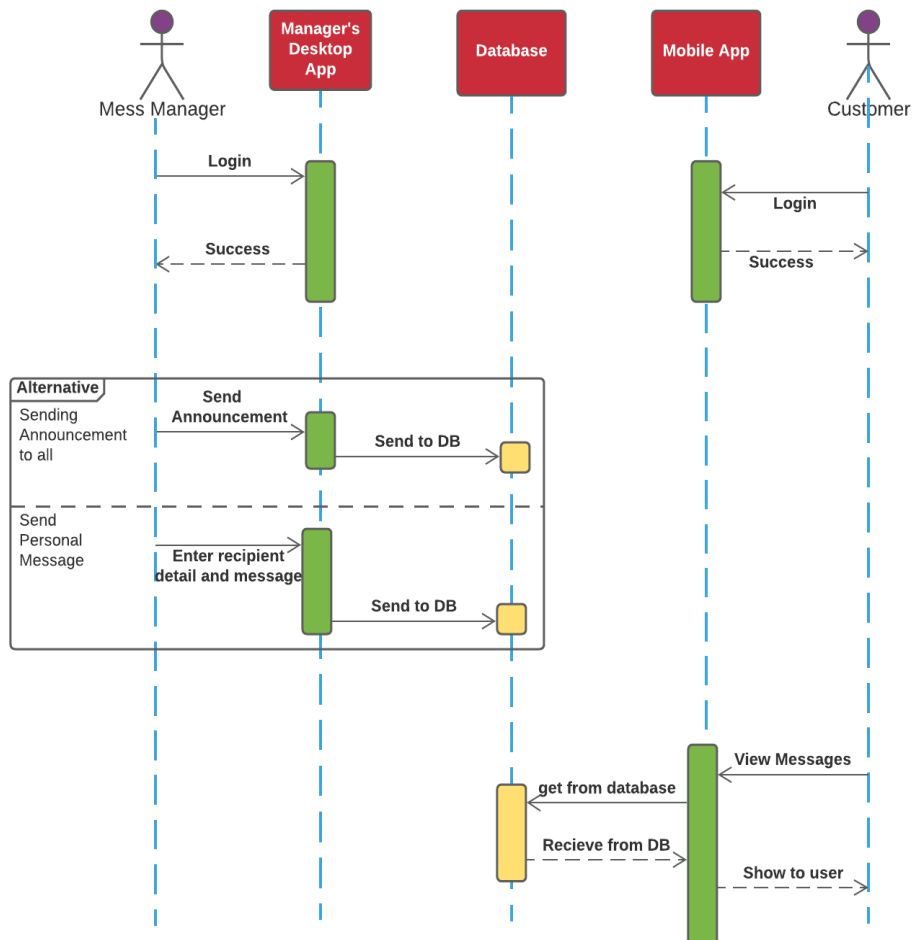
Specify quantity



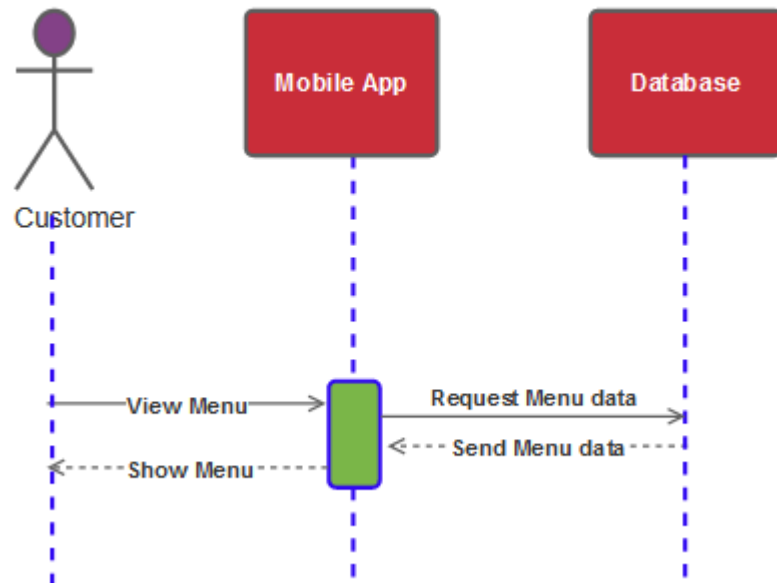
Send Feedback



Send Announcement/Message



View items



Change Order Status/Quantity

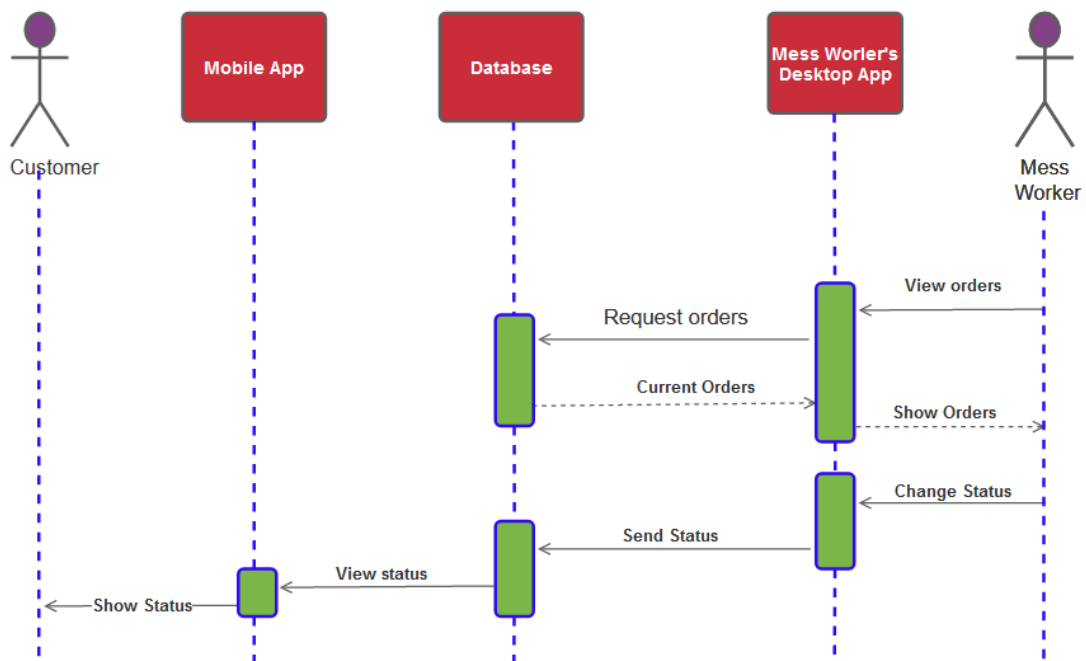


Figure 6: Sequence Diagram

4.3. Class Diagram

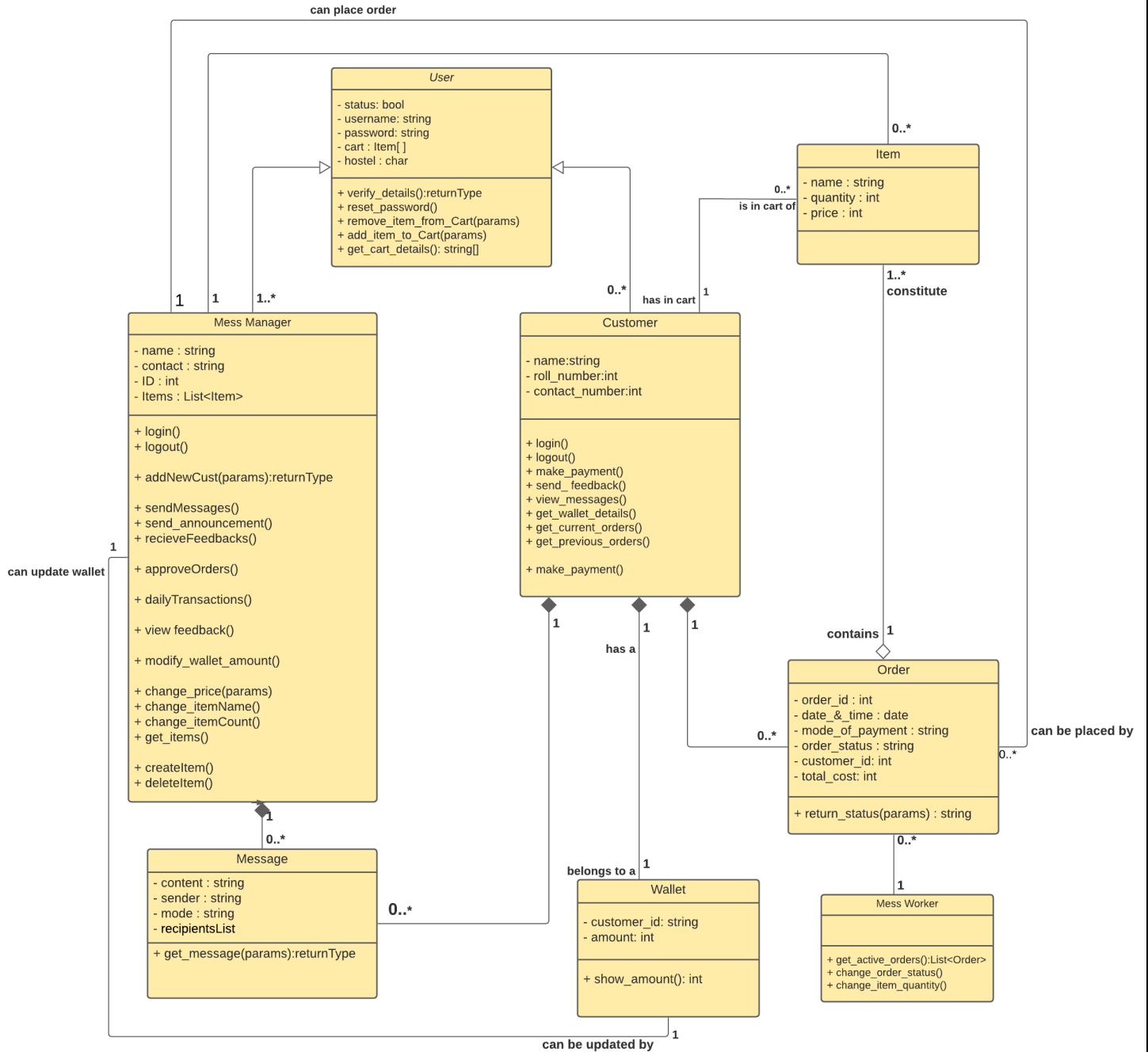


Figure 7: Class Diagram