B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

Lab Record

Software Engineering and Object-Oriented Modeling

Submitted in partial fulfillment for the 5th Semester Laboratory

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

SHASHIDHAR B M

1BM22CS257

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
Mar-June 2024

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Object-Oriented Analysis and Design(22CS6PCSEO) laboratory has been carried out by **SHASHIDHAR B M (1BM22CS257)** during the 5th Semester Oct24-Jan2025.

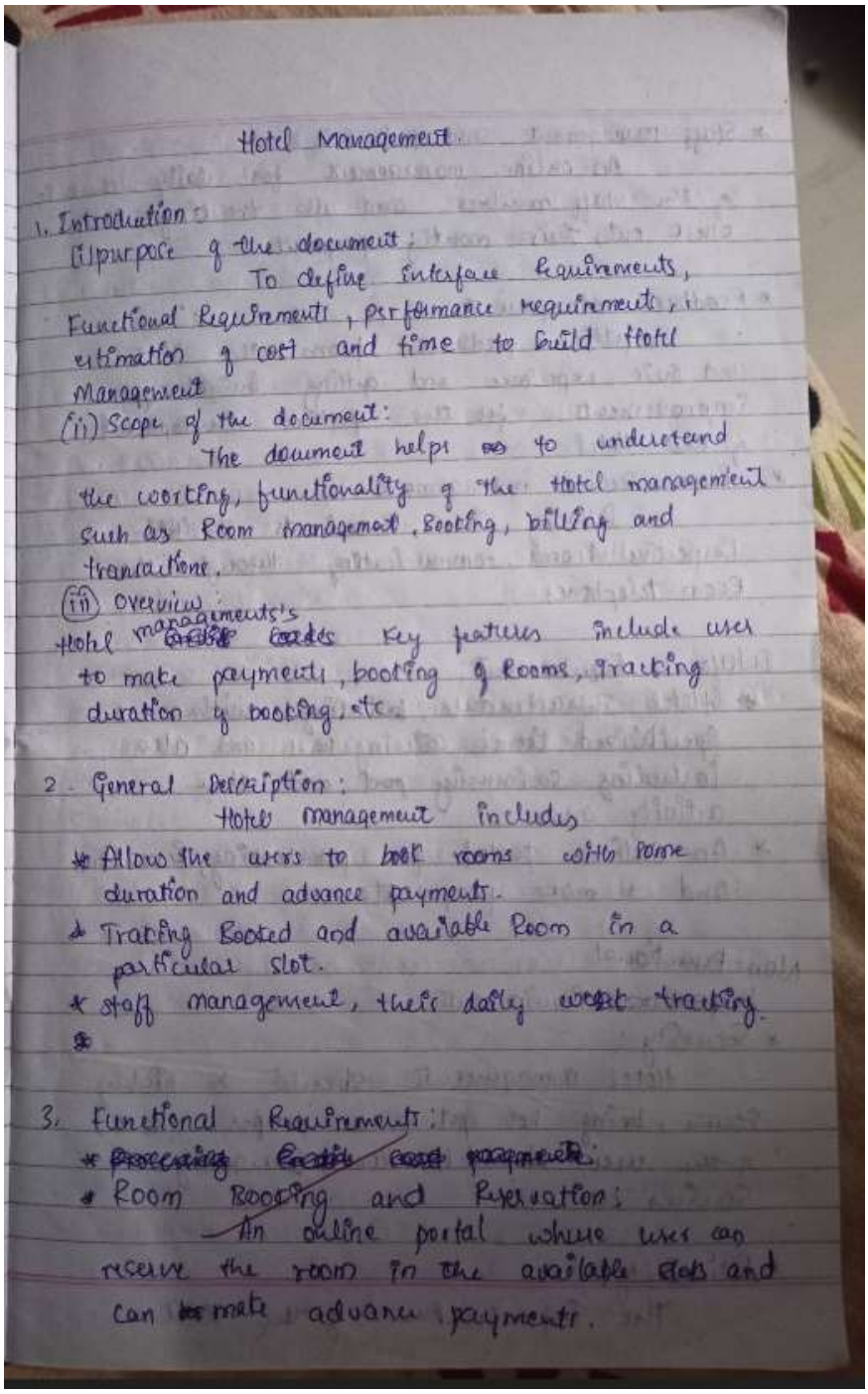
Signature of the Faculty Incharge:

Sunayana S
Assistant Professor
Department of Computer Science and Engineering
B.M.S. College of Engineering, Bangalore

Table of Contents

1. Hotel Management System

Software Requirement Specification



Hotel Management

1. Introduction:

(i) purpose of the document:

To define interface Requirements, Functional Requirements, performance requirements, estimation of cost and time to build Hotel Management

(ii) Scope of the document:

The document helps us to understand the working, functionality of the Hotel management such as Room management, Booking, billing and transactions.

(iii) Overview:

Hotel management's Key features include user to make payments, booking of Rooms, tracking duration of booking, etc.

2. General Description:

Hotel management includes

* Allow the users to book rooms with some duration and advance payments.
* Tracking Booked and available Room in a particular slot.
* staff management, their daily work tracking.

3. Functional Requirements:

* Room Booking and Reservation:

An online portal where user can receive the room in the available slots and can make advance payments.

* Staff management and their role:
An online management for daily check in of the staff members and also tracking their check out, their monthly payment.

* Feedback management:
Getting feedback from the user and their experience and getting insights & improvements for the further development of the Hotel.

* Room service management:
Delivering the Requirents to their respective Rooms, communicating through the Room telephones.

Interface Requirement:
* Hotel Infrastructure, building, with specialized Rooms, catering hall and also including swimming pool and other activity area.
* An Online portal for pre-booking Reservation, and to make payments.

Non-Functional
~~Performance~~ Requirements:
* Security:
Hotel management should be highly secure, being hot spot area proper checks of the user should be done, Emergency Services etc.

* Reliability:
The Infrastructure of the Hotel

management should be strong enough, and reliable.

* Scalability:
  It can be easily scalable for future upgradations.

performance Requirements:

* Faster and efficient payments, billing & the user and faster response to user.
* Faster Room service and catering to the users.
* clean, hygenic hotel and friendly enviroment in the hotel.

Design constraints:

* No. of Rooms in the hotel should be restricted below 50.
* Hotel staff should contain atleast 20 in strength
* Room design should be made for only 2 persons.
* Minimum advance payment for online Room booking should be 25%.

Preliminary Schedules and Budget:

project takes around 8 years to plain, design, build and cat.

Cost → 10 crore to build, 40 lakhs/month - management.

Schedule

planning	2 months.
design	6 months.
build	2.3 years.
testing and use	6 months.

Class Diagram

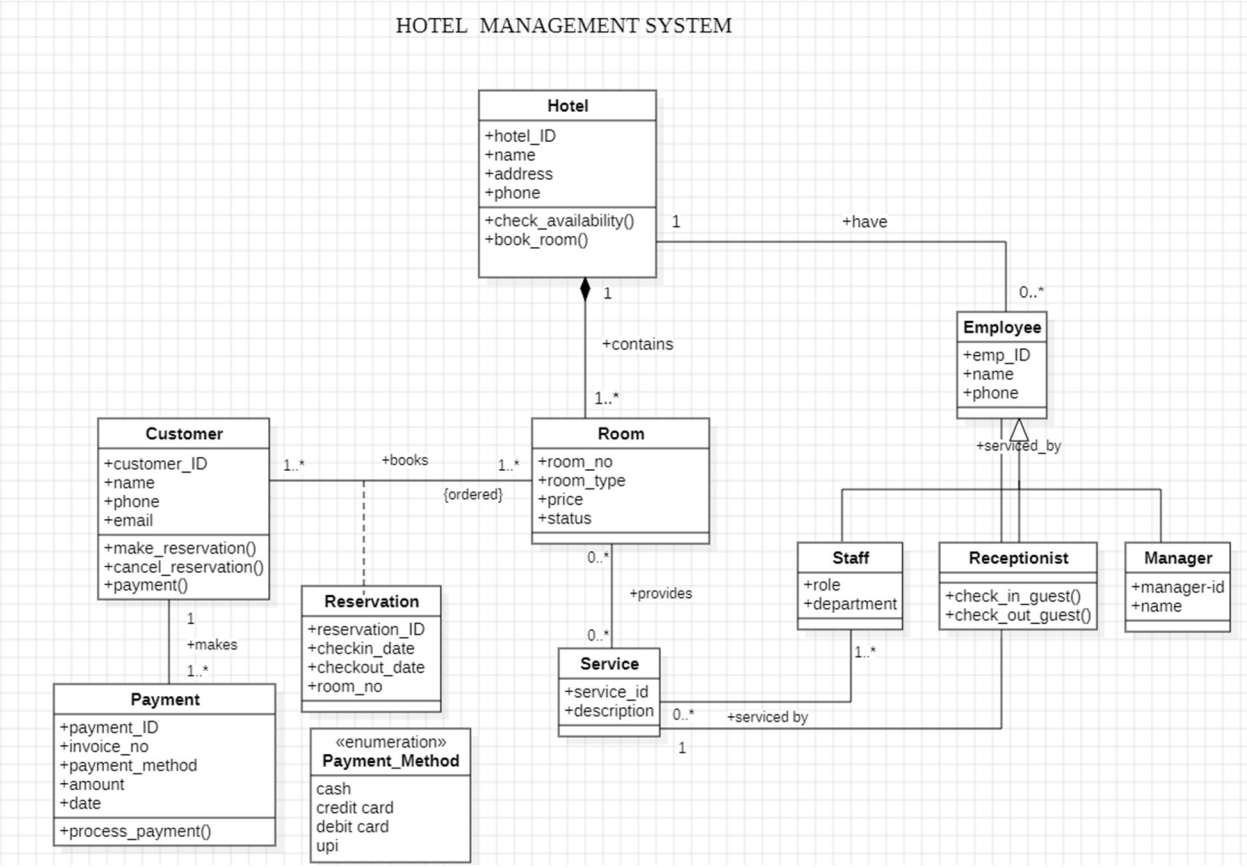


Fig1.1 Hotel Management System - Class Diagram

The diagram represents a hotel management system. It showcases the relationships between various entities such as Customer, Hotel, Room, Reservation, Payment, Service, and Staff. The diagram defines the attributes and operations associated with each entity, such as making a reservation, checking in/out guests, processing payments, etc. It also depicts the relationships between these entities, including one-to-one, one-to-many, and many-to-many relationships. For example, a customer can make multiple reservations, each reservation is associated with a specific room, and different types of staff members can be involved in various services. The diagram provides a comprehensive overview of the system's structure and interactions

State Diagram

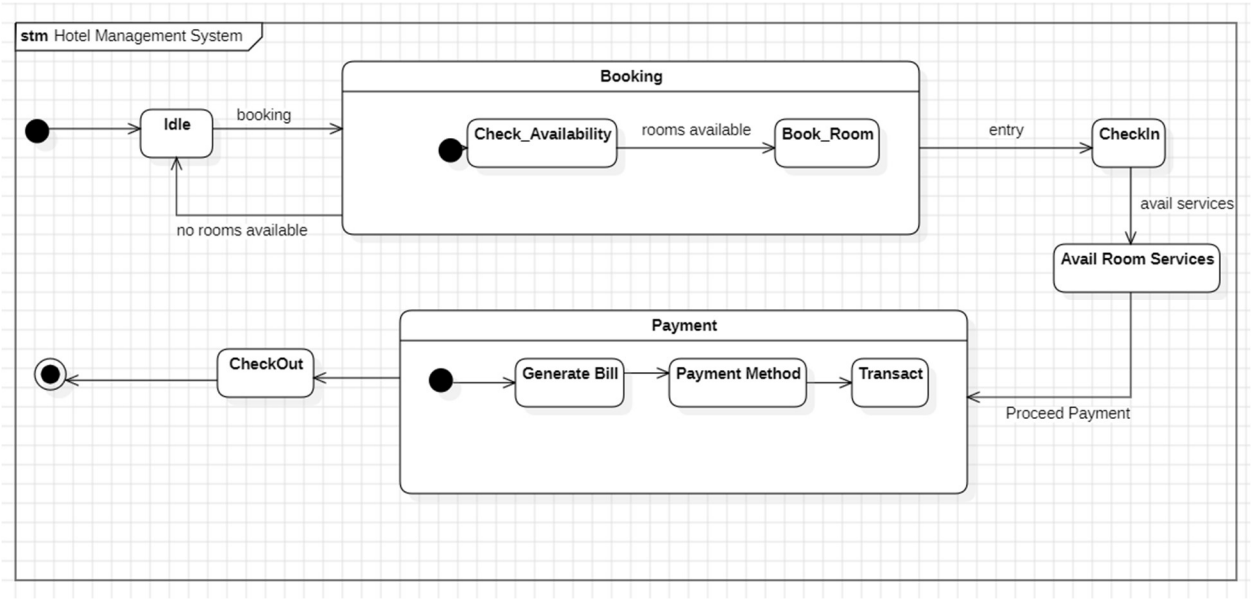


Fig1.2 Hotel Management System - State Diagram

The hotel management system state machine models the operational workflow of a hotel, transitioning through distinct states triggered by specific events. It begins in an Idle state, waiting for booking requests. Upon receiving a request, it transitions to Check_Availability to verify room availability. If rooms are available, it moves to Book_Room, confirming the booking; otherwise, it returns to Idle. Once booked, the customer proceeds to Checkin, after which they can avail services in Avail Room Services. The CheckOut state initiates upon the customer's departure, followed by Generate Bill to prepare their bill. In Payment Method, the customer selects how to pay, leading to Transact, where the payment is processed. Each state and transition ensures smooth and sequential operation of the system, ensuring efficiency and clarity in hotel management.

Use Case Diagram

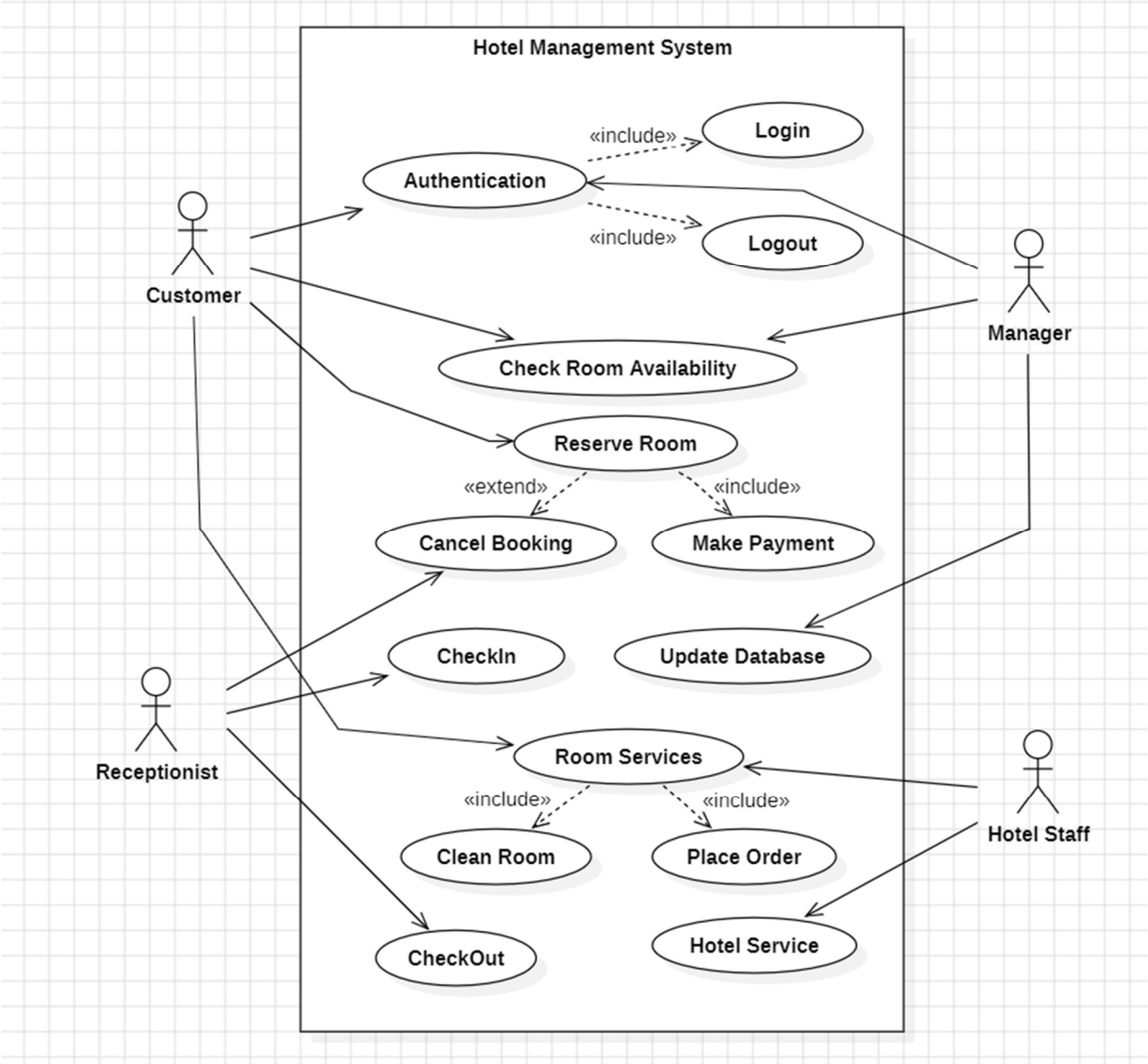


Fig1.3 Hotel Management System - Use Case Diagram

The diagram represents a Use Case Diagram for a Hotel Management System, showcasing various interactions between users (actors) and system functionalities. The primary actors include Customer, Manager, Receptionist, and Hotel Staff. Key use cases are grouped under the system, such as Authentication (which includes login and logout), Check Room Availability, Reserve Room (extended by Cancel Booking and including Make Payment), Check-In, and Room Services (further including cleaning, placing orders, and other hotel services). The diagram emphasizes the relationships and interactions among actors and system processes, demonstrating how each user contributes to the system's operations. For example, the Manager and Receptionist oversee updates and reservations, while Hotel Staff handle room services.

Sequence Diagram

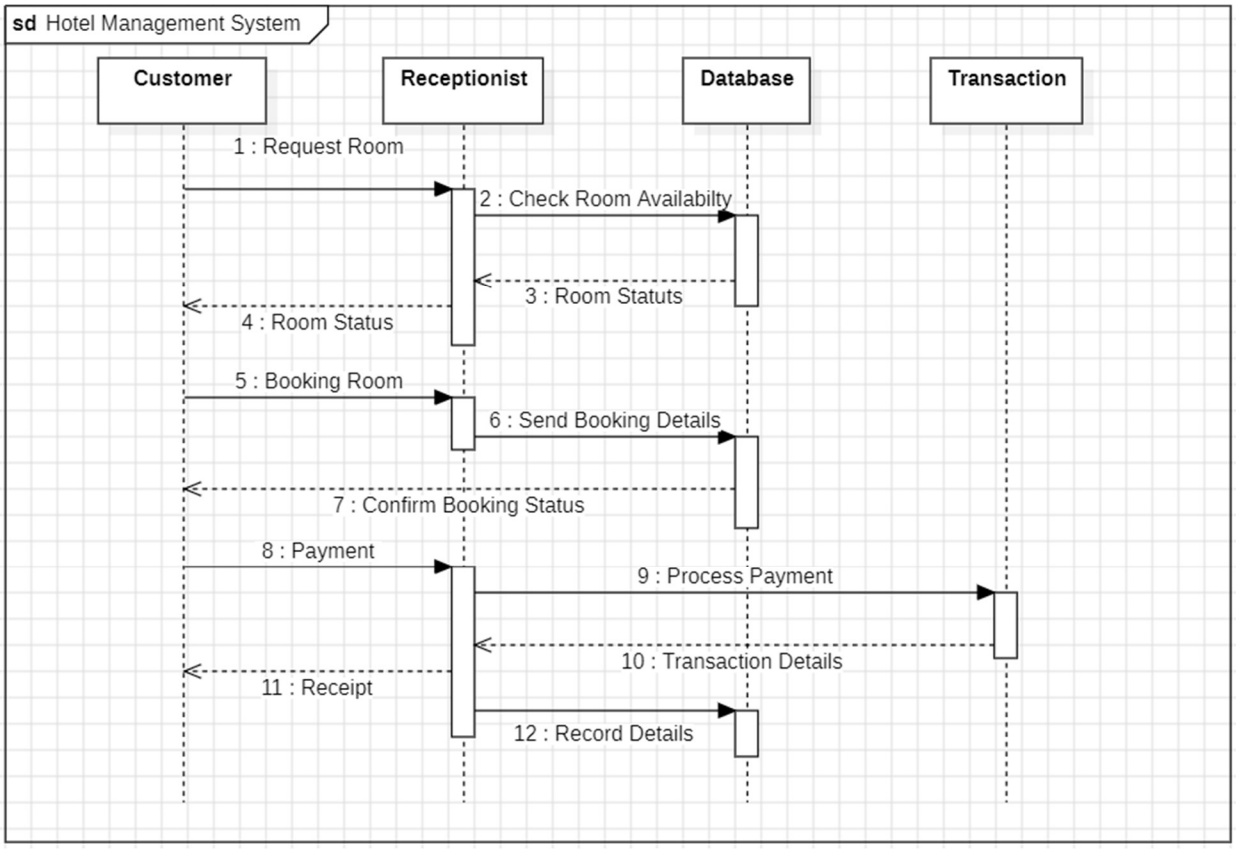


Fig1.4 Hotel Management System - Sequence Diagram

The sequence diagram illustrates the process of booking a room at a hotel. The customer initiates the process by requesting a room. The receptionist then checks the availability of the room in the database and returns the status to the customer. If the room is available, the customer can book the room. The receptionist sends the booking details to the database and confirms the booking status to the customer. The customer then makes the payment, and the transaction is processed by the database. Finally, the customer receives a receipt, and the database records the transaction details.

Activity Diagram

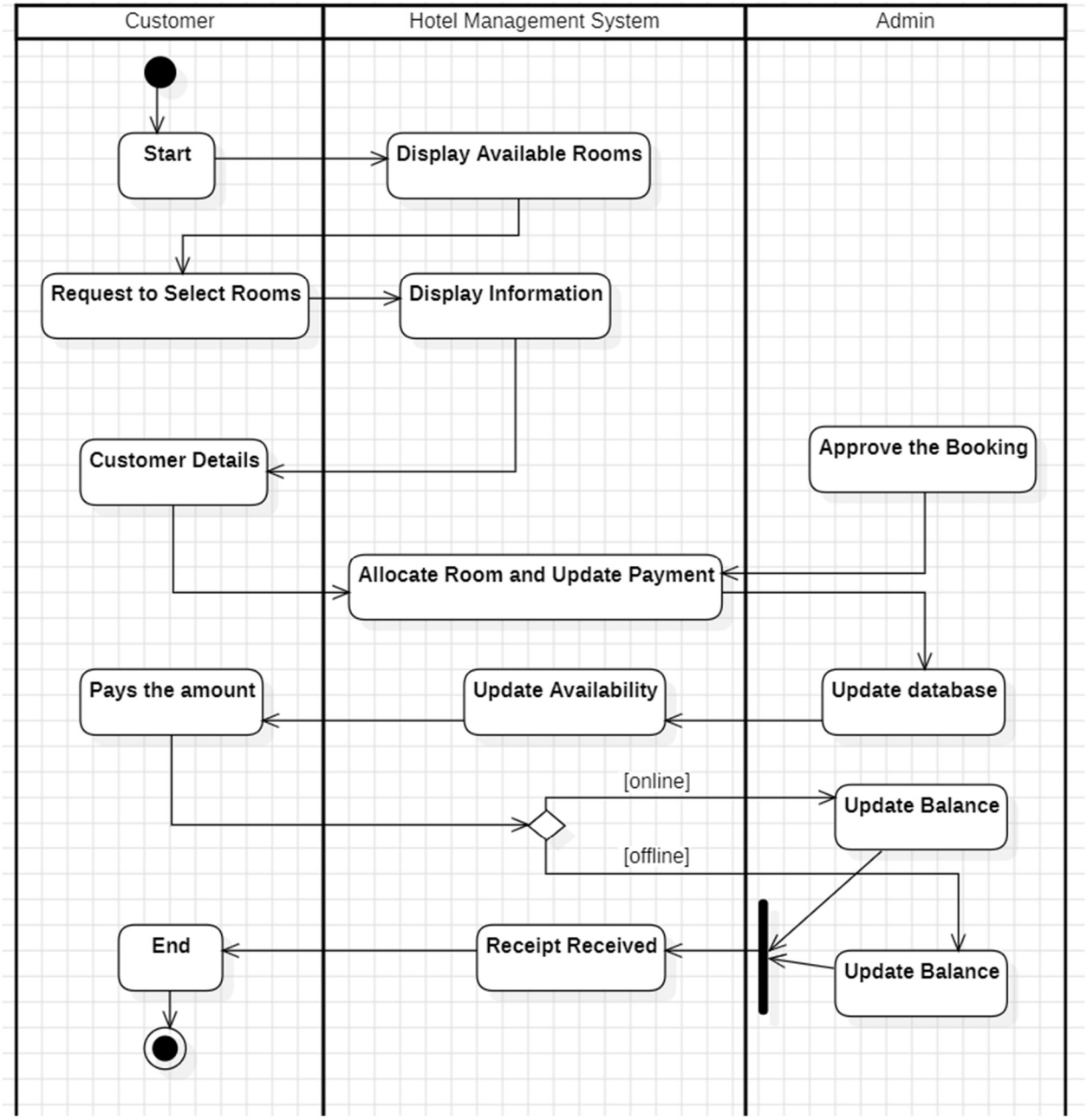
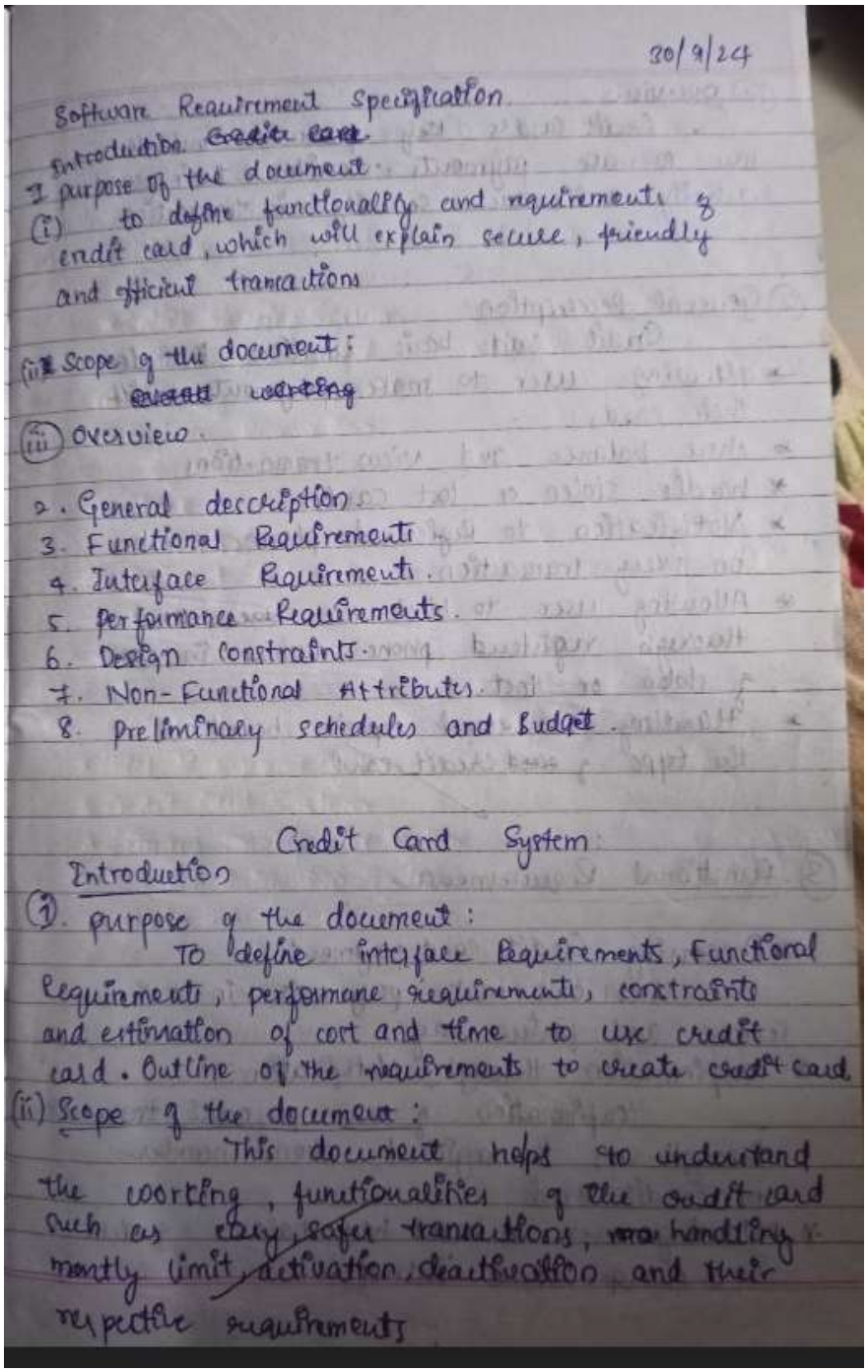


Fig1.5 Hotel Management System - Activity Diagram

The activity diagram illustrates the process of booking a room at a hotel. The customer starts by requesting to select rooms, and the system displays available rooms and their information. The customer then provides their details and selects a room. The admin approves the booking, and the system allocates the room and updates the payment. The customer pays the amount, and the system updates the room availability and balance. Finally, the customer receives a receipt, and the system updates the balance.

2.Credit Card Processing System

Software Requirement Specification



30/9/24

Software Requirement Specification

Introduction. ~~Credit card~~

I purpose of the document.

(i) to define functionality and requirements of credit card, which will explain secure, friendly and efficient transactions

(ii) Scope of the document:

~~overall working~~

(iii) Overview.

2. General description
3. Functional Requirements
4. Interface Requirements.
5. performance Requirements.
6. Design constraints.
7. Non-Functional Attributes.
8. preliminary schedules and budget.

Credit Card System

Introduction

(i). purpose of the document:

To define interface Requirements, Functional Requirements, performance requirements, constraints and estimation of cost and time to use credit card. Outline of the requirements to create credit card.

(ii) Scope of the document:

This document helps to understand the working, functionalities of the credit card such as ebay, safe transactions, ~~ma~~ handling monthly limit, activation, deactivation and their respective requirements

(1) overview.

Credit cards Key features include user to make payments, monitor their card activity and ensure security in transaction.

(2) General Description.

Credit cards basic functions include.

* allowing user to make payments with their cards
* check balance and view transactions
* handle stolen or lost cards
* Notification to Registered phone number on every transaction.
* Allowing user to block the ~~card~~ card through registed phone number in case of stolen or lost.
* Handling the anual deposit based on the type of ~~card~~ credit card.

(3) Functional Requirements

* Processing Credit card payments:
  User can make payments, in a secure and faster way
* Confirmation through Notification:
  Confirmation of the payment transaction through sms to registed phone number to credit card.
* Allowing user to ~~be~~ have balance of debt and charge some intrest

* Verification at every transaction:
  Verification of validity of card, whether the user have sufficient credits to make further payments.
* handling annual fee of credit card:
  Managing dues, and annual fee to be paid based on the type of credit card using.
* Detection and alerts to user if there any fraud activities

Interface Requirements:
* App or web application:
  where user can login with their card details and can check balance and make payments.
* Payment gateway using credit card and communication with other systems like banks, to process payments

Performance Requirement:
* Credit card payment should take minimum time nearly 1-2 seconds.
* Credit card service should be available consistently at all the time 24x7.
* Minimum error in transactions

Design Constraints:
* design of the credit card includes EMV chip, 16-digit card number, 3-digit CVV and other details
* web interface using React + firebase and flutter mobile app and algorithms to handle fraud

Non-functional Attributes:

* portability and compatability:
Credit card system can be made from any comm, using any platform that may be actual credit card, or software application

* Security:
transactions in credit card should be secure, incase of fault in the system immediate refund to the user.

* Reliability:
Credit card should be working, consistent at all times with the same quality.

* Scalability:
It can be easily scalable for future updations.

Prilimiary Schedule and Budget:

* project take around 8 months to plan, design, code or build and test.

* Schedule: planning 1 week
design 1 month
build 5 months
test 1.5 months.
final build 2 weeks
and deploy

* Cost: Build and deployment: ₹10,00,000
testing: 2,00,000.
Maintainance: 1,00,000/month

28/9/2024

Class Diagram

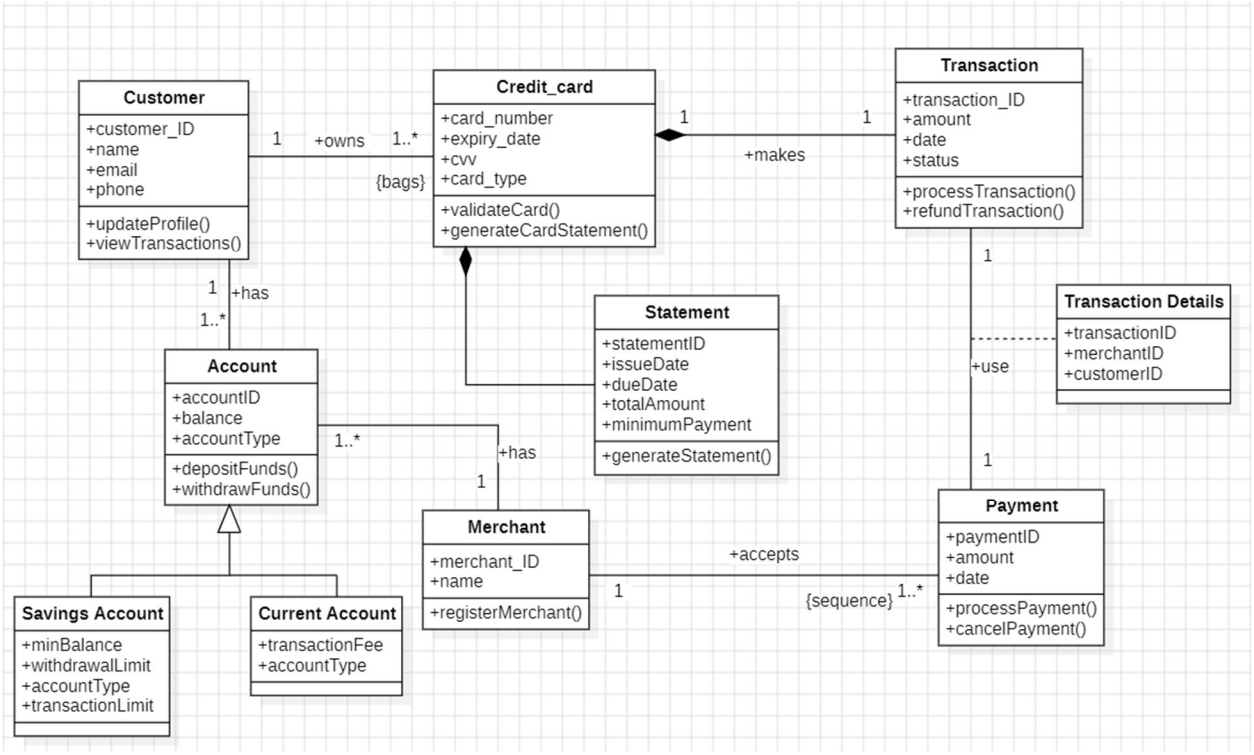


Fig 2.1 Credit Card Processing System - Class Diagram

The class diagram represents a credit card processing system. Customers own one or more Credit Cards, which are used to perform Transactions. Each credit card is validated and associated with a Statement that includes payment details like total amount and due date. Accounts (Savings or Current) store the customer's funds and enable deposits and withdrawals. Merchants register to accept payments, and payments are linked to Transaction Details, specifying the customer and merchant involved. Key functionalities include processing and refunding transactions, validating credit cards, generating statements, and updating customer profiles.

State Diagram

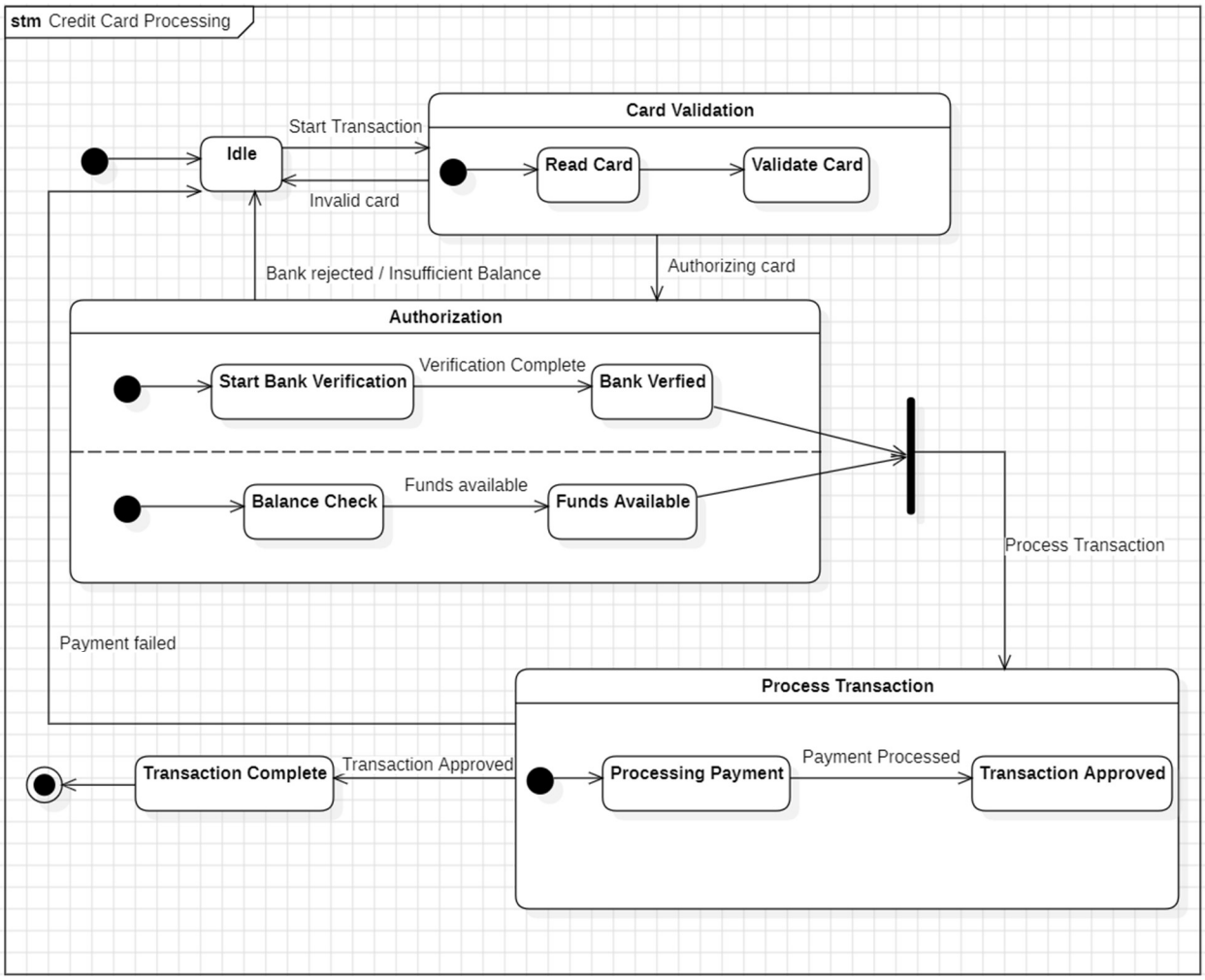


Fig 2.1 Credit Card Processing System - State Diagram

The state diagram illustrates the process of a credit card transaction. The system starts in an idle state and transitions to the "Read Card" state when a transaction is initiated. The card is then validated, and if it is invalid, the transaction is rejected. If the card is valid, the system moves to the "Authorization" state and verifies the card with the bank. If the card is verified and the funds are available, the system proceeds to the "Process Transaction" state and completes the transaction. If the card is not verified or there are insufficient funds, the transaction fails.

Use Case Diagram

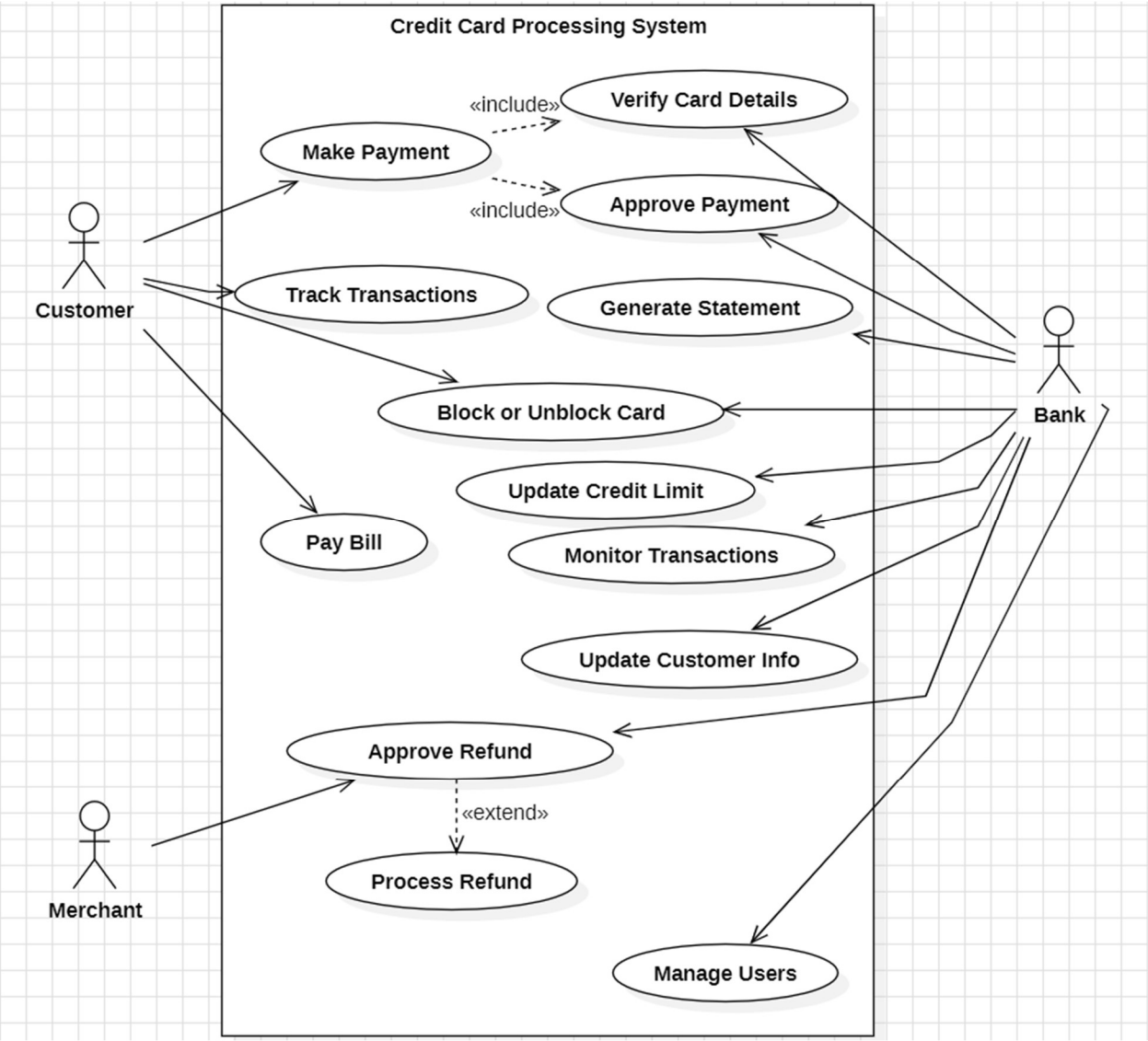


Fig 2.1 Credit Card Processing System - Use Case Diagram

The diagram depicts a Use Case Diagram for a Credit Card Processing System, highlighting the roles of the primary actors: Customer, Bank, and Merchant. The Customer interacts with the system to make payments (which includes verifying card details and approving payments), track transactions, generate statements, block or unblock cards, and pay bills. The Bank is responsible for approving payments, monitoring transactions, updating customer information, adjusting credit limits, and managing users. Additionally, merchants can request refunds, which involve approval and subsequent processing by the system. This diagram effectively illustrates the interactions and responsibilities within the credit card processing workflow.

Sequence Diagram

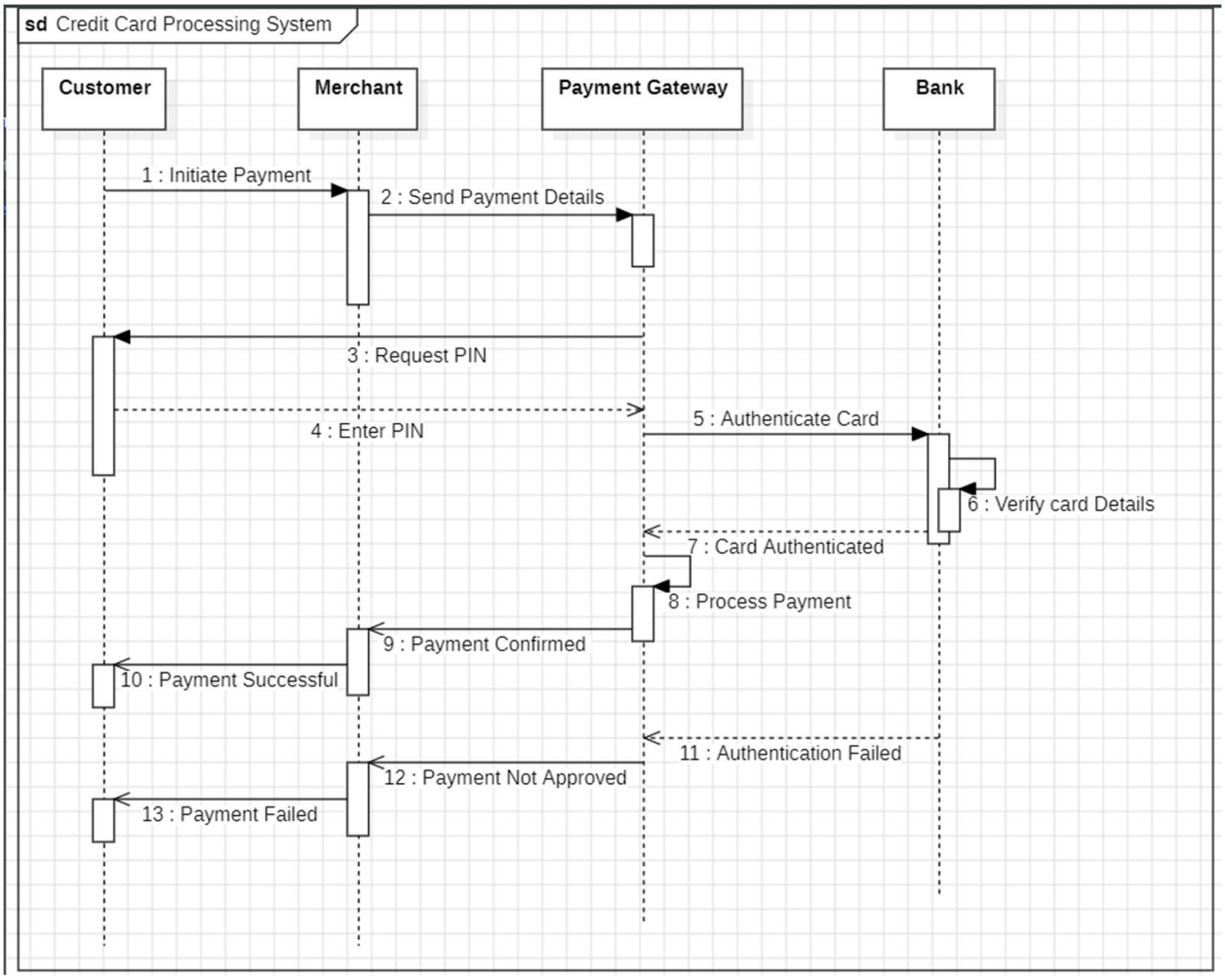


Fig 2.1 Credit Card Processing System - Sequence Diagram

The sequence diagram illustrates the process of a credit card transaction. The customer initiates the payment, and the merchant sends the payment details to the payment gateway. The payment gateway requests the customer to enter their PIN for authentication. Once the PIN is entered, the gateway authenticates the card with the bank. If the card is authenticated, the payment gateway processes the payment and confirms it to the merchant. Finally, the customer receives a notification of successful payment. If the card authentication fails, the payment is not approved

Activity Diagram

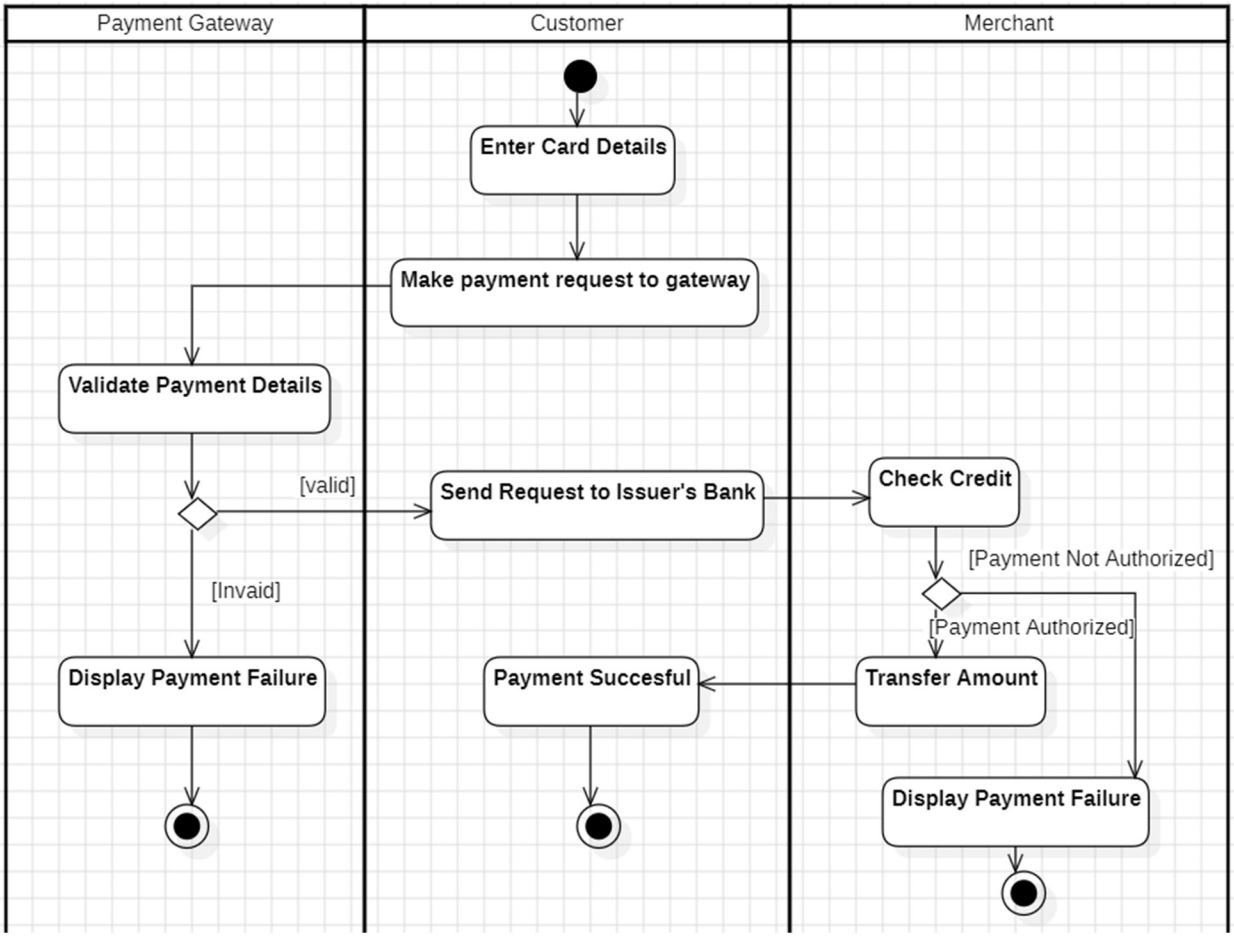
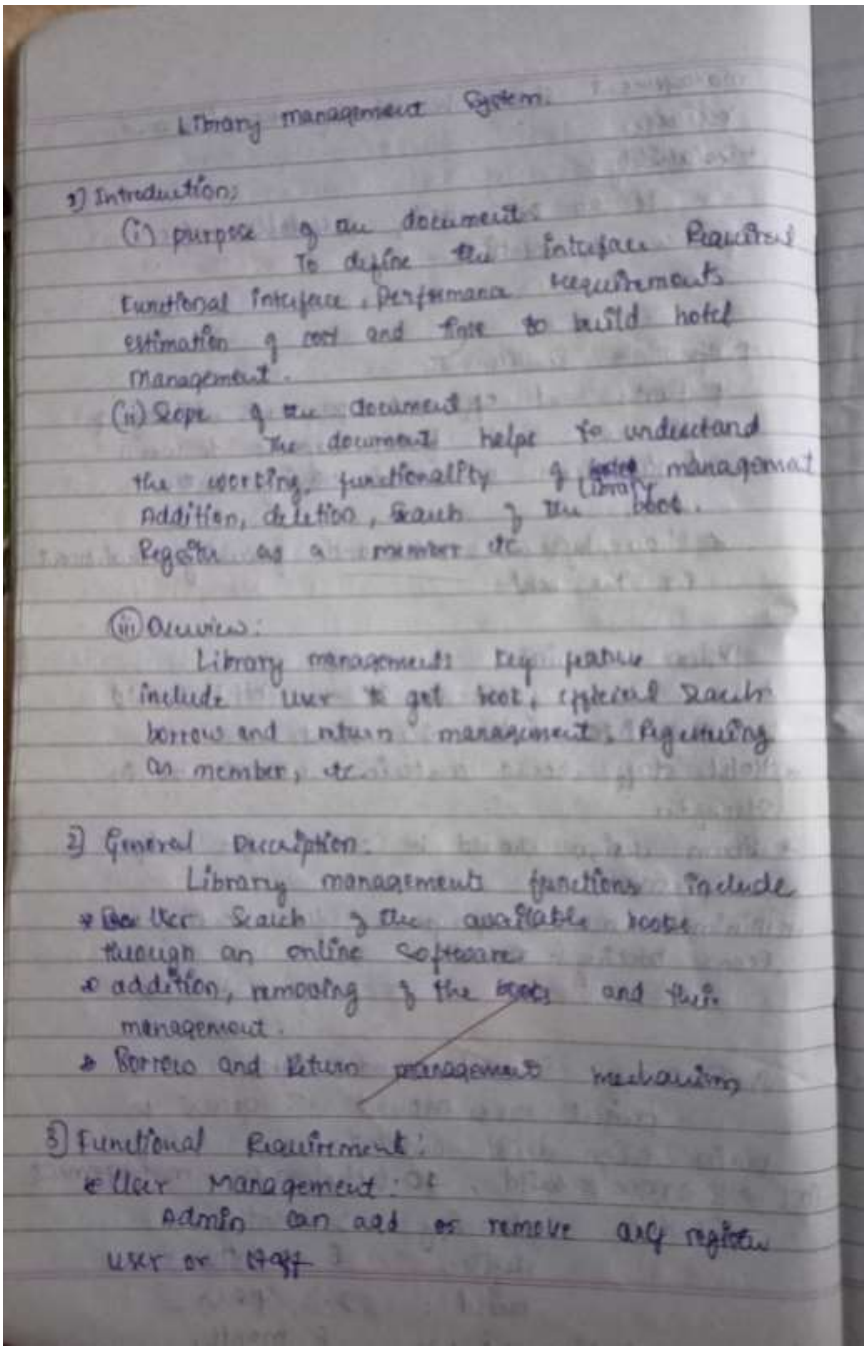


Fig 2.5 Credit Card Processing System - Activity Diagram

The activity diagram illustrates the process of a credit card transaction. The customer starts by entering their card details and making a payment request to the gateway. The gateway validates the payment details. If the details are valid, the gateway sends a request to the issuer's bank to check the credit. If the credit check is successful, the bank authorizes the payment and the gateway transfers the amount. The customer then receives a notification of successful payment. If the payment details are invalid or the credit check fails, the transaction is rejected, and the customer receives a notification of payment failure.

3. Library Management System

Software Requirement Specification



Library management System.

1) Introduction:

(i) purpose of the document:

To define the Interface Requirements Functional Interface, Performance requirements estimation of cost and time to build hotel Management.

(ii) Scope of the document:

The document helps to understand the working functionality of Library management Addition, deletion, Search of the book. Register as a member etc.

(iii) Overview:

Library management key feature include User to get book, efficient Search borrow and return management, Registering as member, etc.

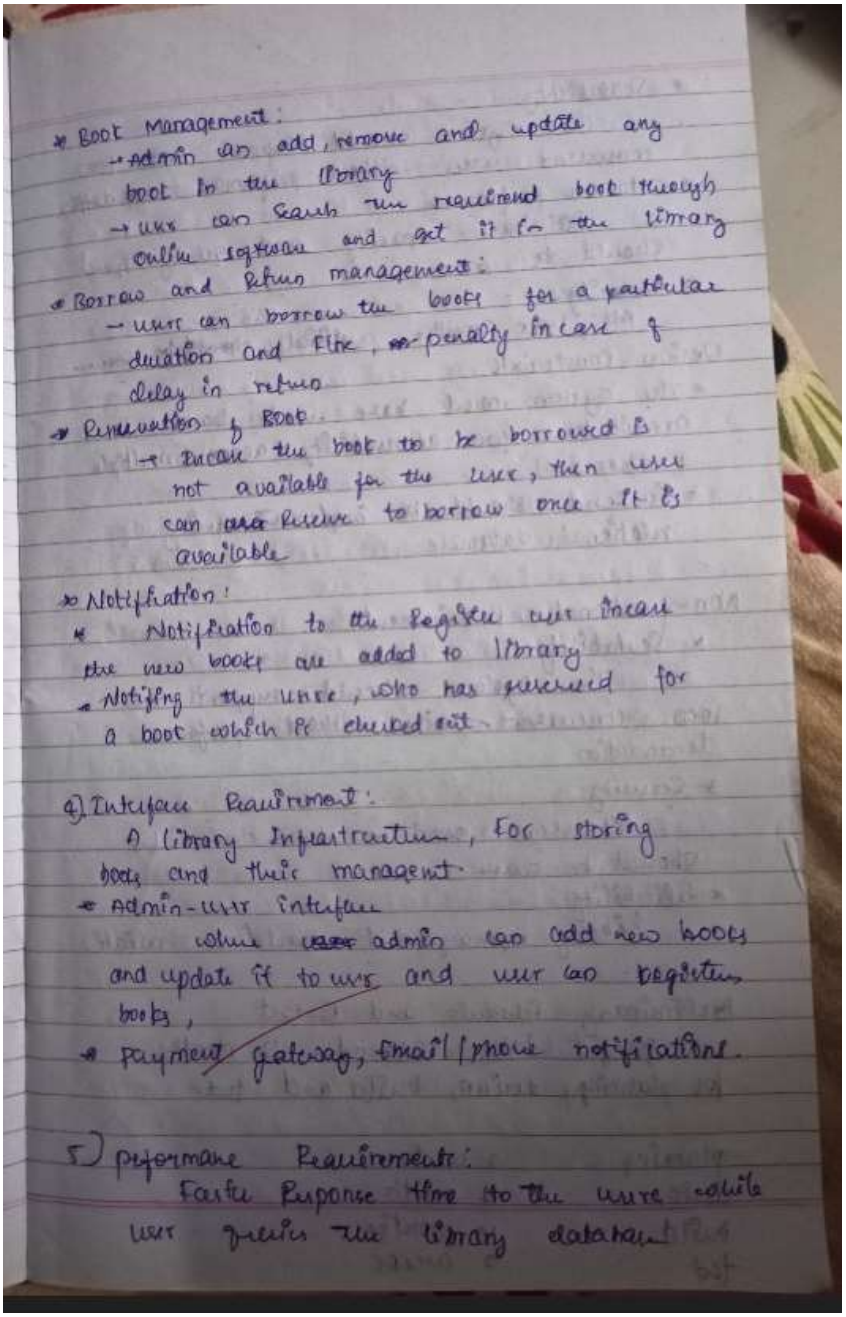
2) General Description:

Library management functions include
- User Search of the available books through an online software.
- addition, removing of the books and their management.
- Borrow and Return management mechanism.

3) Functional Requirement:

- User Management:

Admin can add or remove any register user or staff.



* Book Management:
  - Admin can add, remove and update any book in the library
  - User can search the required book through online software and get it from the library
* Borrow and Return management:
  - user can borrow the books for a particular duration and fine, penalty in case of delay in return
* Reservation of Book
  - Incase the book to be borrowed is not available for the user, then user can Reserve to borrow once it is available
* Notification:
  - Notification to the Register user incase the new books are added to library
  - Notifying the user, who has reserved for a book which is checked out

4) Interface Requirement:
   A library Infrastructure, For storing books and their management
* Admin-user interface
  where admin can add new books and update it to user and user can register books,
* Payment gateway, Email/phone notifications.

5) Performance Requirements:
   Fastly Response time to the user while user prints the library database

* Scalability:
The system should support upto 1000 concurrent users without performance degradation.
* Database performance
Database capacity and query speed should be more
* ~~Security~~:
All user sensitive ~~credentials~~ should be ~~secure~~

Design Constraints:
* The system must have web based architecture for accessibility across multiple platforms.
* Backend should be implemented using relational database

Non-functional Requirements:
* Scalability:
The system should support upto 1000 concurrent users without performance degradation
* Security:
All users sensitive data, credentials should be secure.
* Reliability:
Library management should be reliable.

Preliminary Schedule and Budget.
Project take around 12 months for planning, design, build and test.

planning	2 weeks.
design	1 month.
build	10 months
test	3 weeks.

Class Diagram

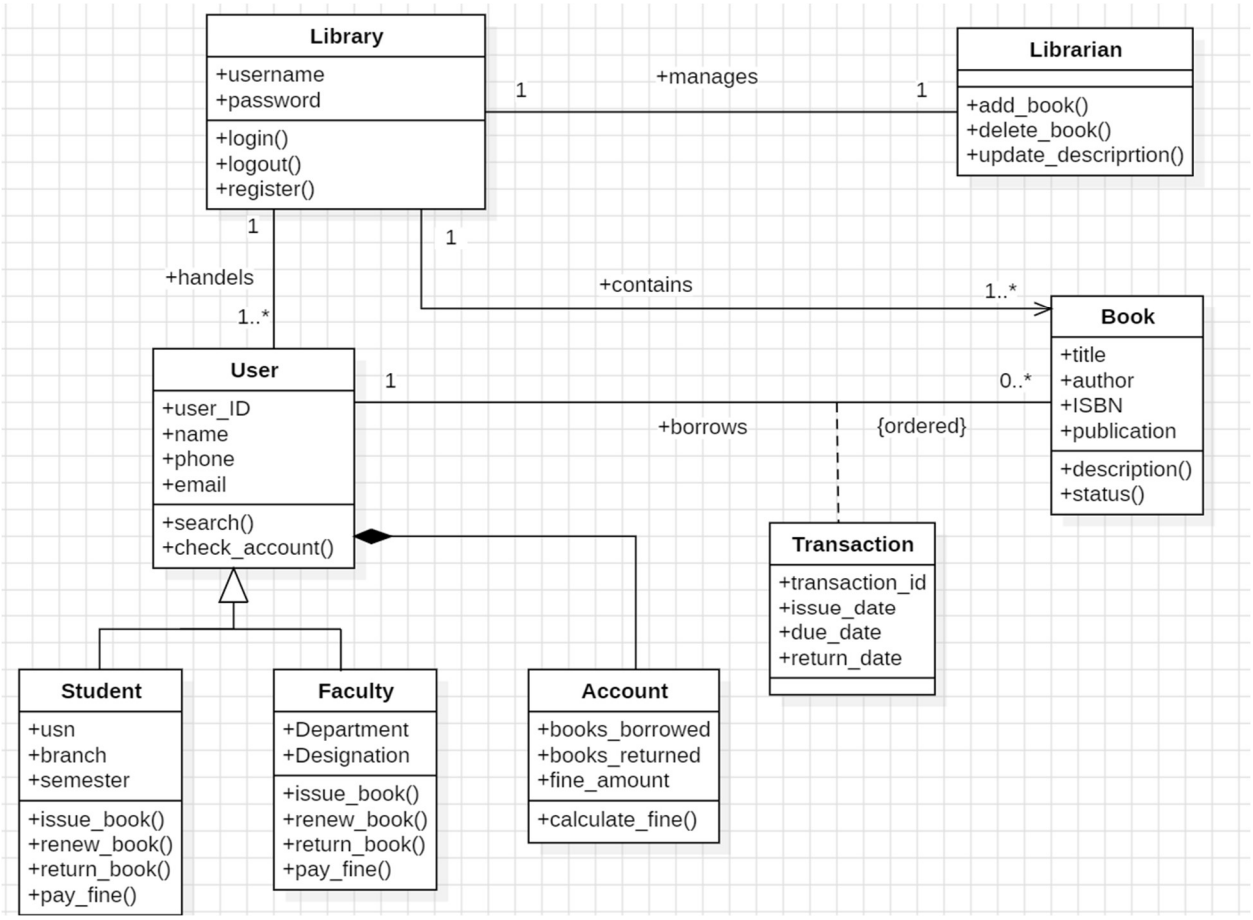


Fig 3.1 Library Management System -Class Diagram

The class diagram represents a library management system, showcasing entities like Library, Librarian, User, Book, Account, and Transaction. The Library handles the system's operations, managed by a Librarian who adds, updates, and deletes books. Users are divided into Students and Faculty, each with functionalities like issuing, renewing, and returning books, managed via their respective Accounts that track borrowed books and fines. Books store details like title, author, and status, while Transactions record borrowing and returning activities. The relationships between these entities ensure seamless management of books, users, and transactions.

State Diagram

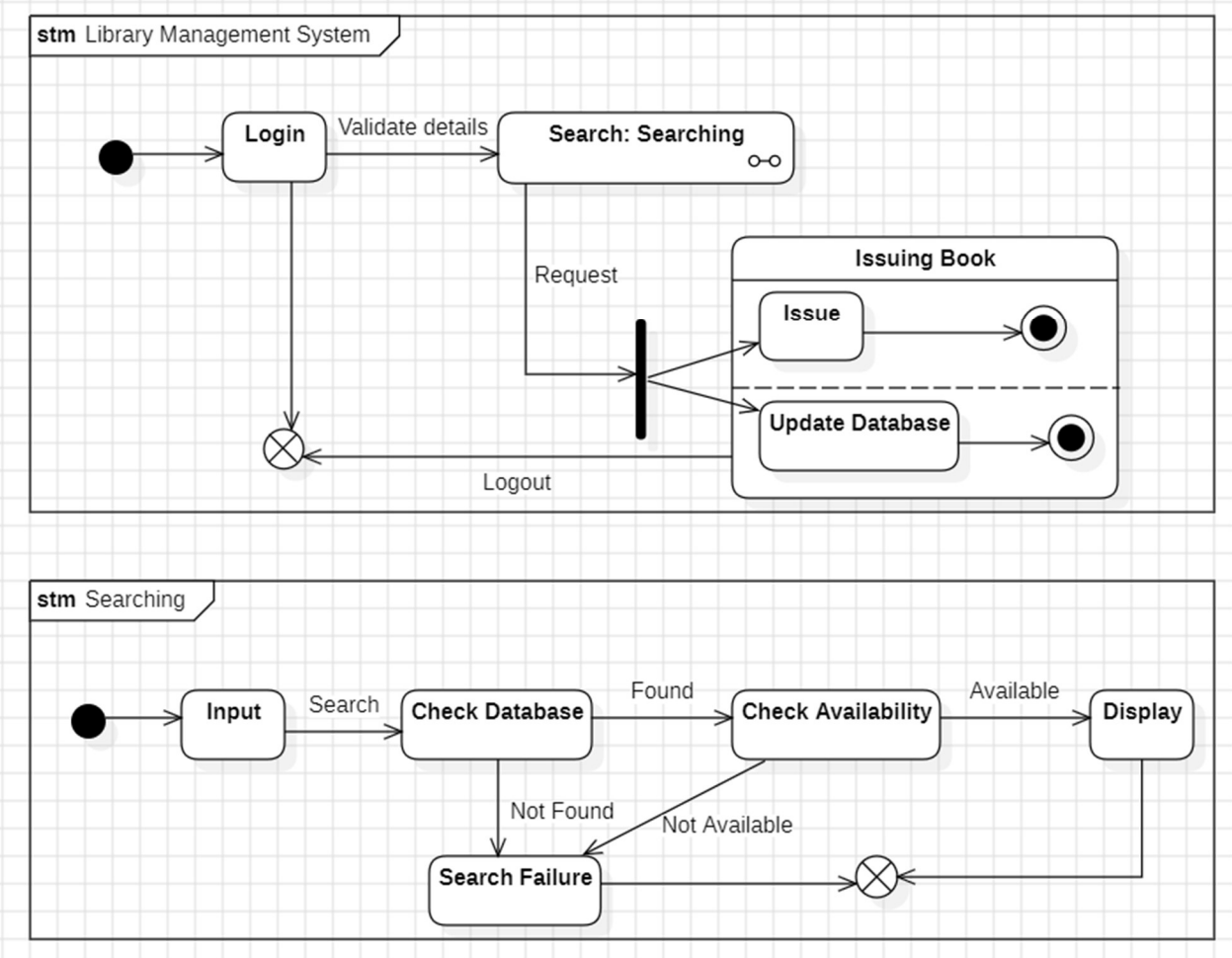


Fig 3.2 Library Management System - State Diagram

The state diagram illustrates the workflow of a library management system. It begins with user login, followed by credential validation. Upon successful login, the system enters the "Searching" state, where the user can search for books. The search process involves checking the database for matches and then checking availability. If a book is available, its details are displayed to the user, who can then request to issue it. The system updates its database accordingly. If the search yields no results or the book is unavailable, the system transitions to the "Search Failure" state. At any point, the user can log out of the system.

Use Case Diagram

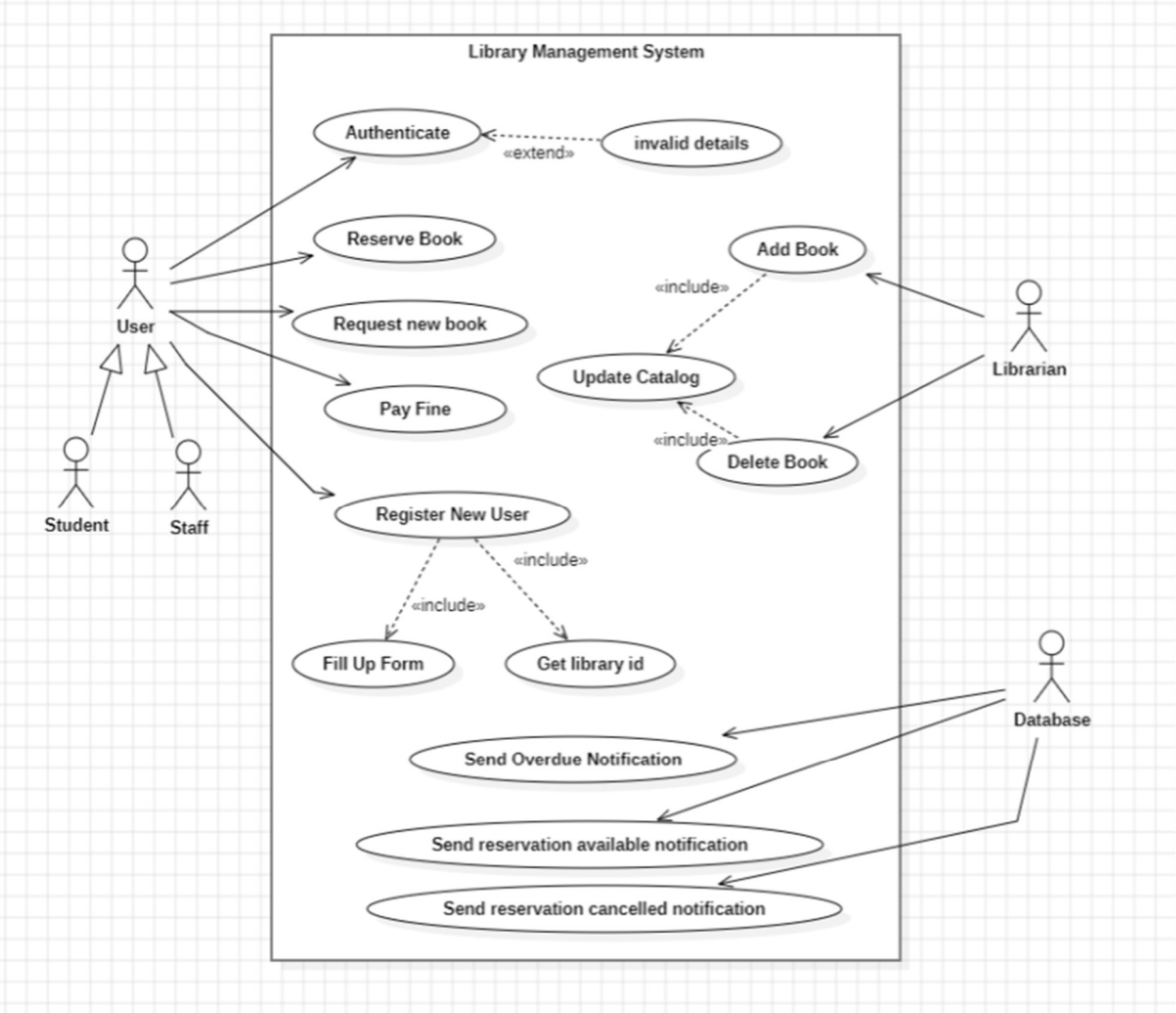


Fig 3.3 Library Management System - Use Case Diagram

The Library Management System is designed to manage the library's resources and user interactions. The system has three main actors: User, Librarian, and Database. The User can reserve books, request new books, pay fines, and register as a new user. The Librarian can add books to the catalog, update the catalog, delete books, and send overdue notifications. The Database stores and manages all the information related to the library, users, and books. The system includes use cases for authentication, filling up forms, and getting library IDs, which are further elaborated by the "include" relationships. This system aims to streamline library operations and provide a convenient experience for users.

Sequence Diagram

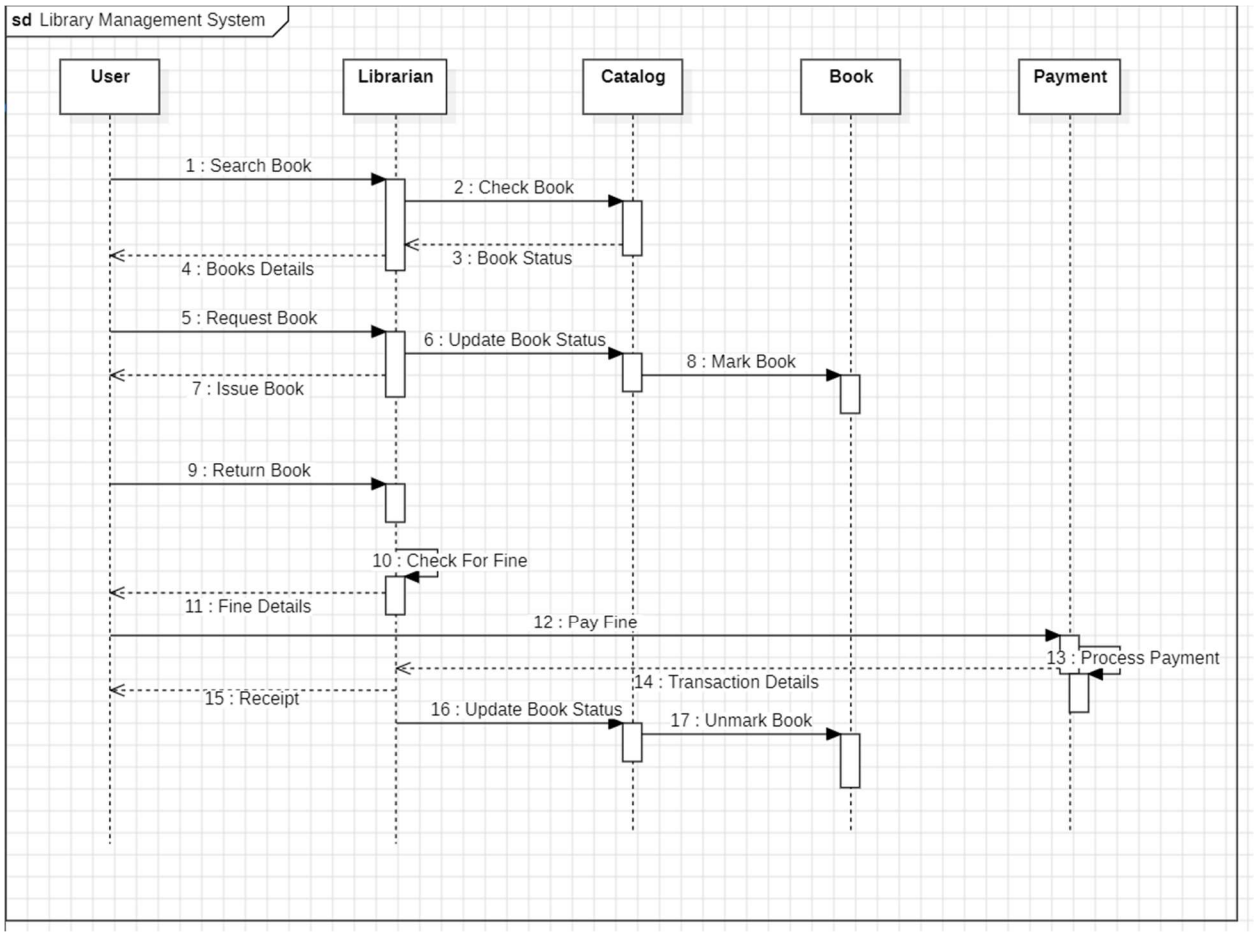


Fig 3.4 Library Management System - Sequence Diagram

The sequence diagram illustrates the process of a user borrowing a book from the library. The user begins by searching for a book in the library catalog. The catalog then searches for the book and returns the results to the user. The user then requests to borrow the book, and the library system checks its availability. If the book is available, the system issues the book to the user and updates its records. Finally, the user receives a receipt confirming the checkout. This diagram highlights the automated steps involved in the process and the interactions between the user and the library system.

Activity Diagram

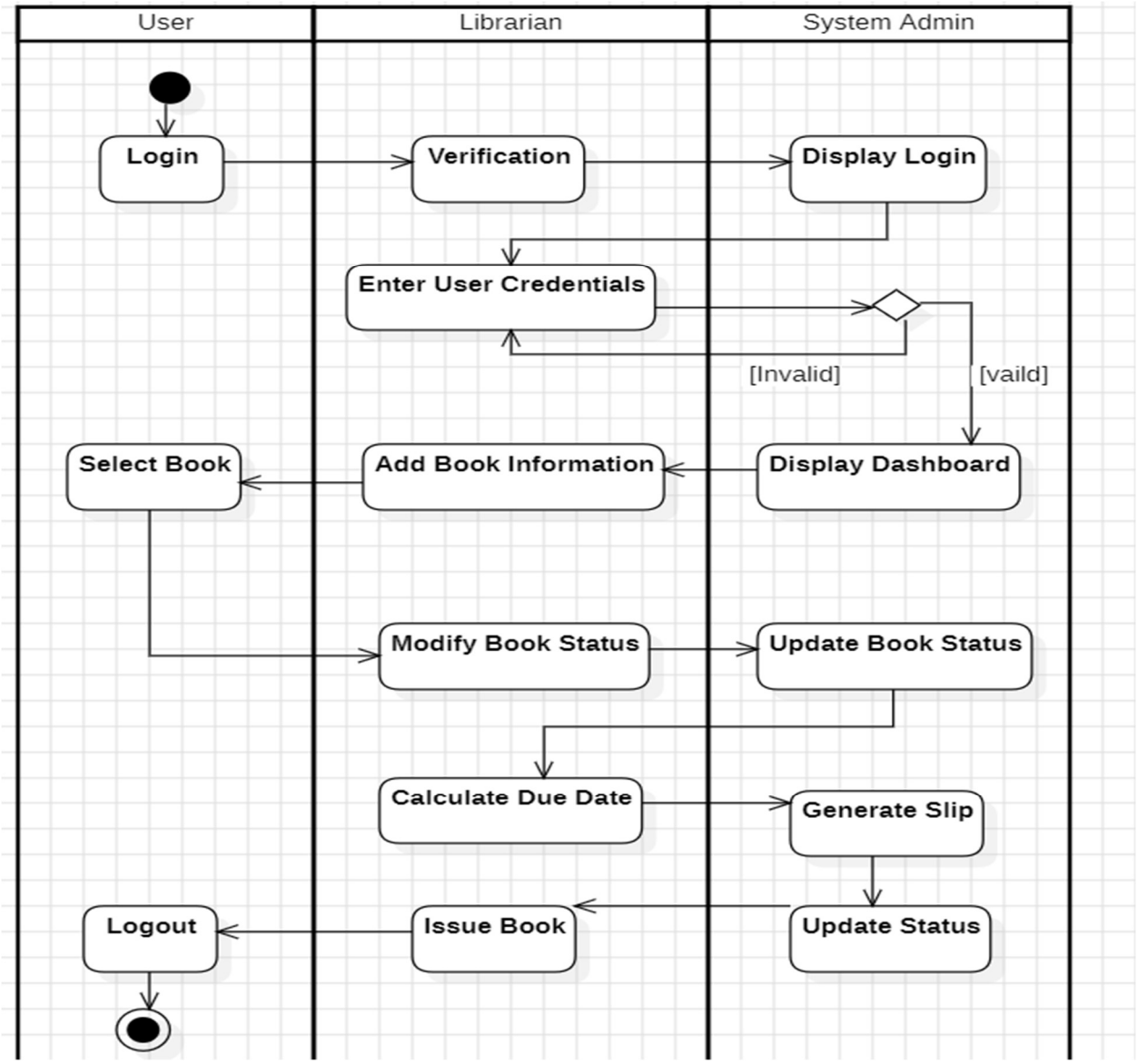
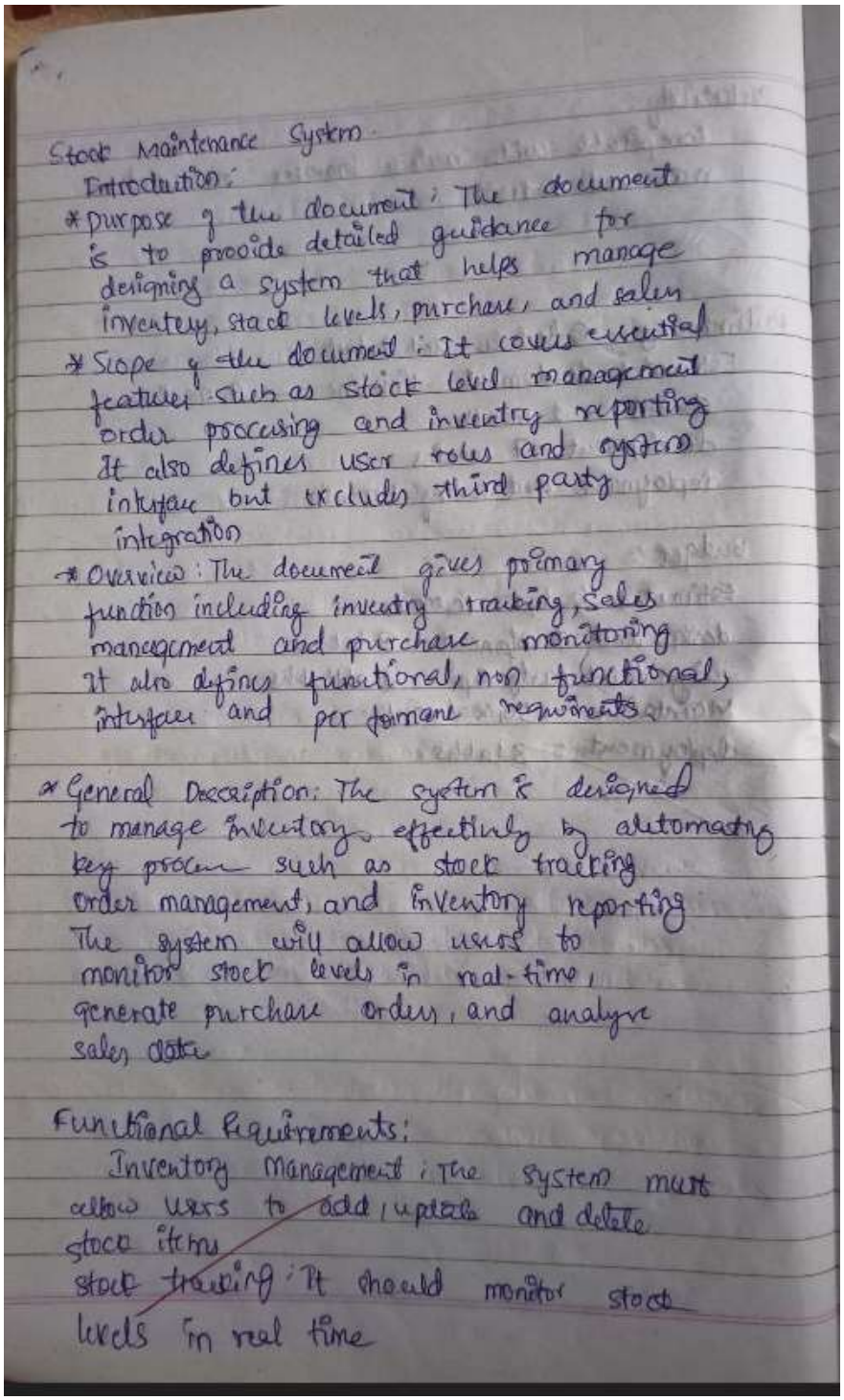


Fig 3.5 Library Management System - Activity Diagram

The activity diagram outlines the workflow of a library management system. It starts with a user logging in, followed by credential verification. Successful login grants access to book selection for the user. Simultaneously, librarians can add new books or modify existing book information. System administrators possess the authority to update book statuses and generate slips related to book transactions. Upon book selection, the system calculates the due date and issues the book to the user, updating the database accordingly. Finally, the system administrator updates the overall book status, and the user can log out. The diagram illustrates the interconnected roles and actions of users, librarians, and system administrators in the library's book borrowing process.

4. Stock Maintenance System

Software Requirement Specification



Stock Maintenance System

Introduction:

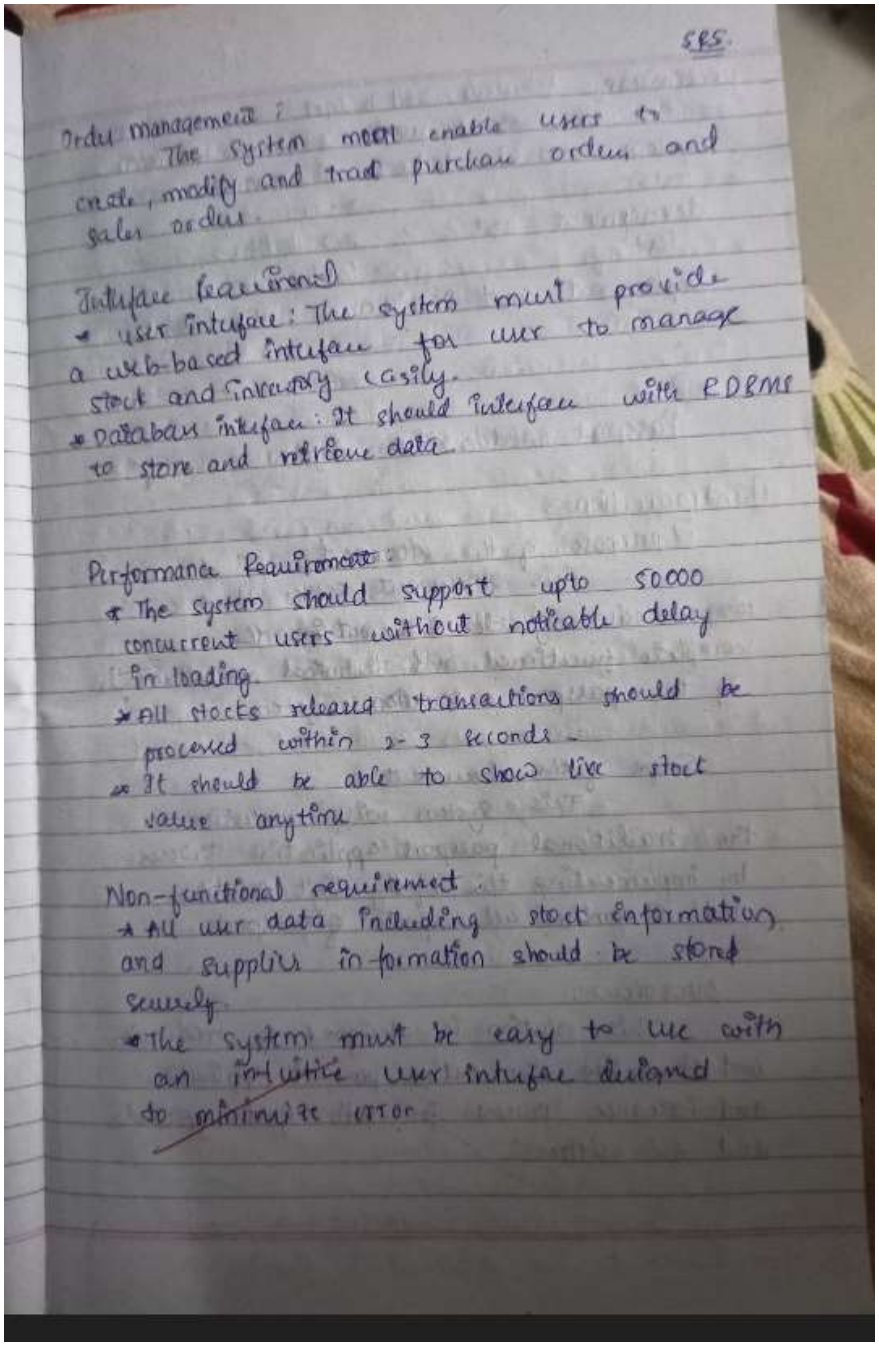
* Purpose of the document: The document is to provide detailed guidance for designing a system that helps manage inventory, stock levels, purchase, and sales
* Scope of the document: It covers essential features such as stock level management order processing and inventory reporting It also defines user roles and system interface but excludes third party integration
* Overview: The document gives primary function including inventry tracking, Sales management and purchase monitoring It also defines functional, non functional, interface and performance requirements

* General Description: The system is designed to manage inventory effectively by automating key process such as stock tracking order management, and inventory reporting The system will allow users to monitor stock levels in real-time, generate purchase orders, and analyze sales data

Functional Requirements:

Inventory Management: The system must allow users to add, update and delete stock items

Stock tracking: It should monitor stock levels in real time



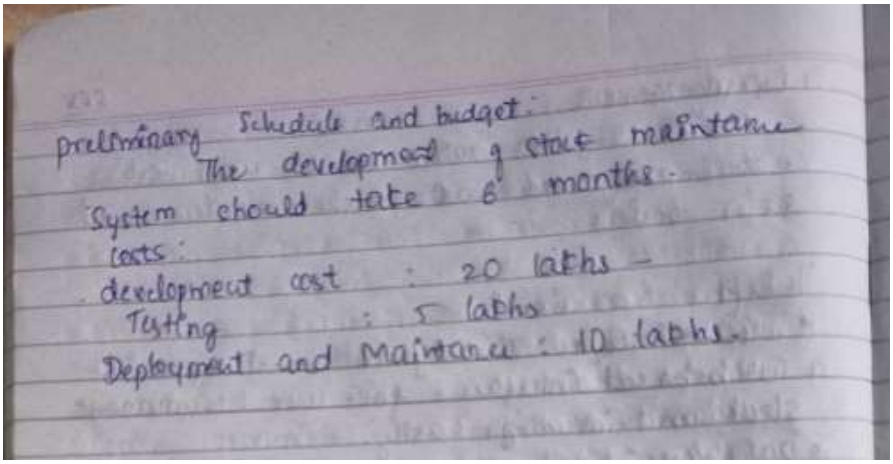
SRS

Order management :
The System must enable users to create, modify and track purchase orders and sales orders.

Interface Requirement
* User interface: The system must provide a web-based interface for user to manage stock and inventory easily.
* Database interface: It should interface with RDBMS to store and retrieve data.

Performance Requirement :
* The system should support upto 50000 concurrent users without noticable delay in loading.
* All stocks related transactions should be processed within 2-3 seconds.
* It should be able to show live stock value anytime

Non-functional requirement :
* All user data including stock information and supplier information should be stored securely.
* The system must be easy to use with an intuitive user interface designed to minimize error


Preliminary Schedule and budget:
The development of store maintance System should take 6 months.
costs:
development cost : 20 lakhs
Testing : 5 lakhs
Deployment and Maintance : 10 lakhs.

Class Diagram

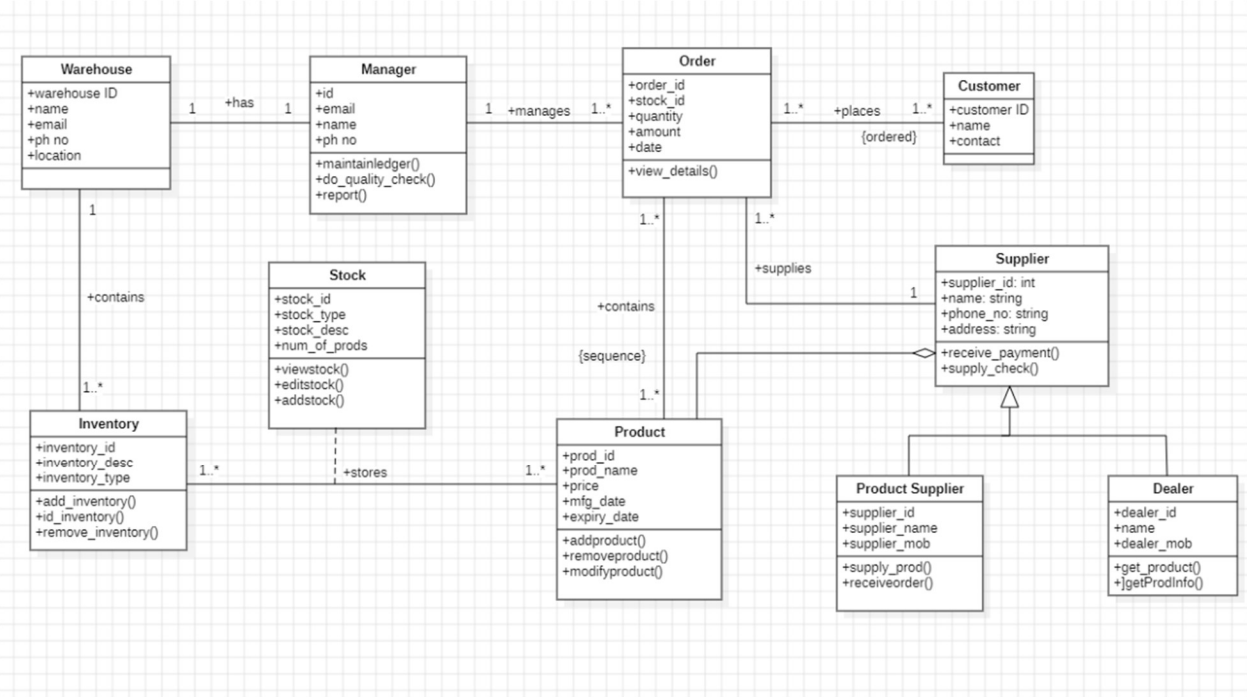


Fig 4.1 Stock Maintenance System - Class Diagram

The class diagram illustrates a warehouse inventory and order management system. The Warehouse contains multiple Inventory items, managed by a Manager who oversees operations like reporting and quality checks. Stock stores details about products, which are managed with functionalities like addition and modification. Orders placed by Customers link products to quantities and amounts, while Suppliers, including Product Suppliers and Dealers, handle the supply of products to the warehouse. The diagram highlights the interactions between inventory, stock, orders, and suppliers within the system.

State Diagram

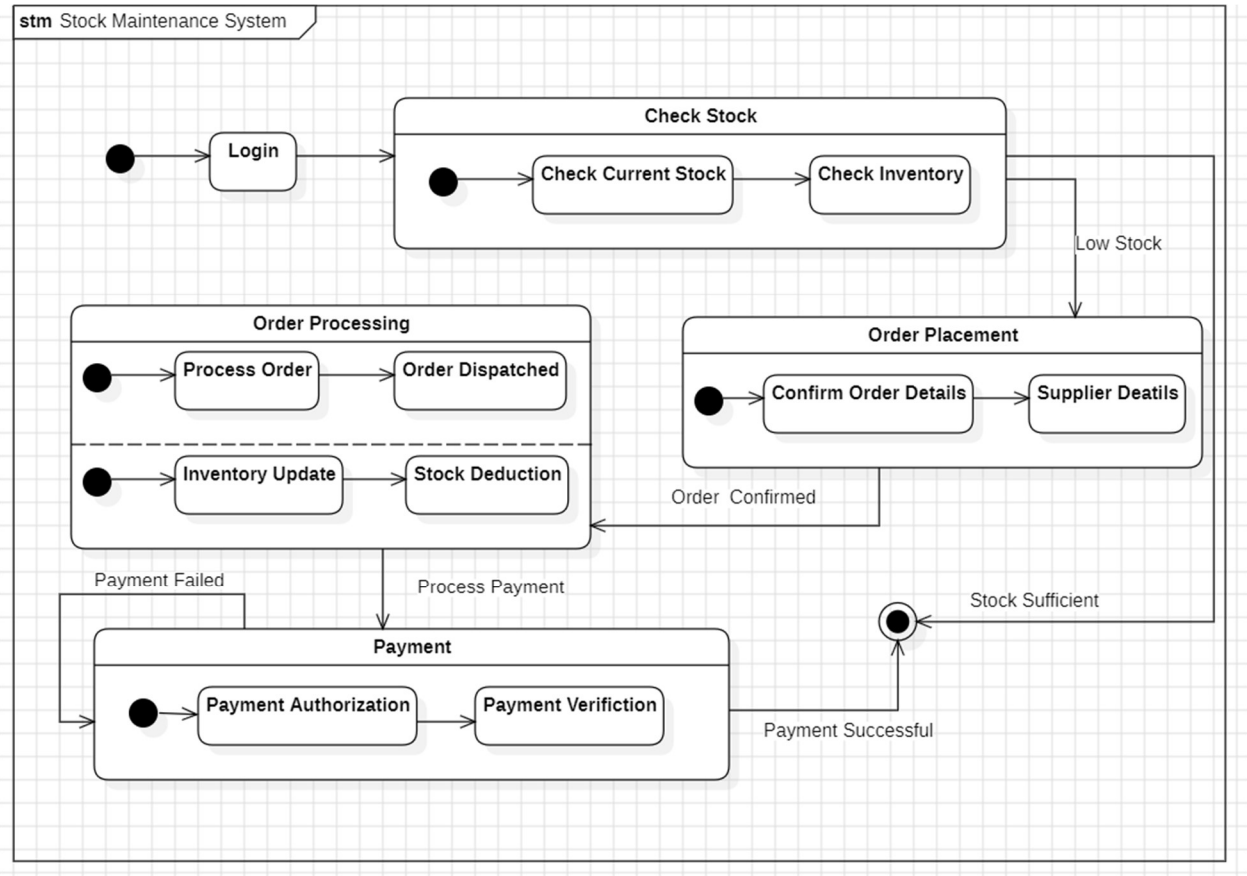


Fig 4.2 Stock Maintenance System - State Diagram

The state diagram illustrates the stock maintenance system's workflow. It starts with a user logging in. The system then checks current stock and inventory levels. If stock is low, the system transitions to the "Order Placement" state, where order details are confirmed and supplier details are obtained. After the order is confirmed, the system moves to the "Order Processing" state, where the order is processed and dispatched. During order processing, stock is deducted and inventory is updated. Finally, the system transitions to the "Payment" state, where payment is authorized and verified. Upon successful payment, the system returns to the "Check Stock" state to monitor inventory levels.

Use Case Diagram

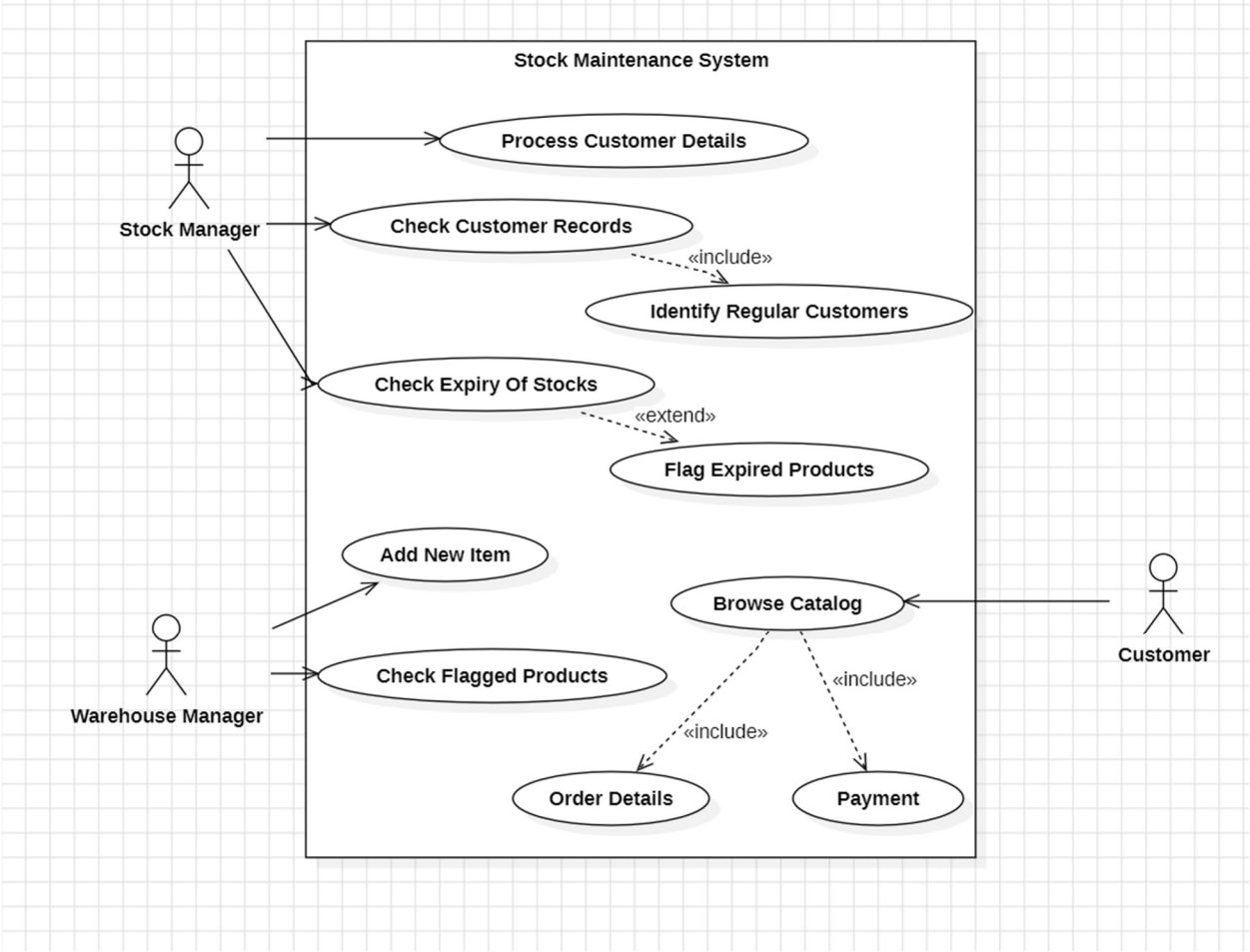


Fig 4.3 Stock Maintenance System - Use Case Diagram

The Stock Maintenance System is designed to manage inventory and customer interactions for a business. The system has three main actors: Stock Manager, Warehouse Manager, and Customer. The Stock Manager can process customer details, check customer records, and identify regular customers. They can also check the expiry of stocks and flag expired products. The Warehouse Manager can add new items to the inventory and check flagged products. The Customer can browse the catalog, place orders, and make payments. The system includes use cases for order details and payment, which are further elaborated by the "include" relationships. This system aims to streamline inventory management and provide a seamless experience for customers.

Sequence Diagram

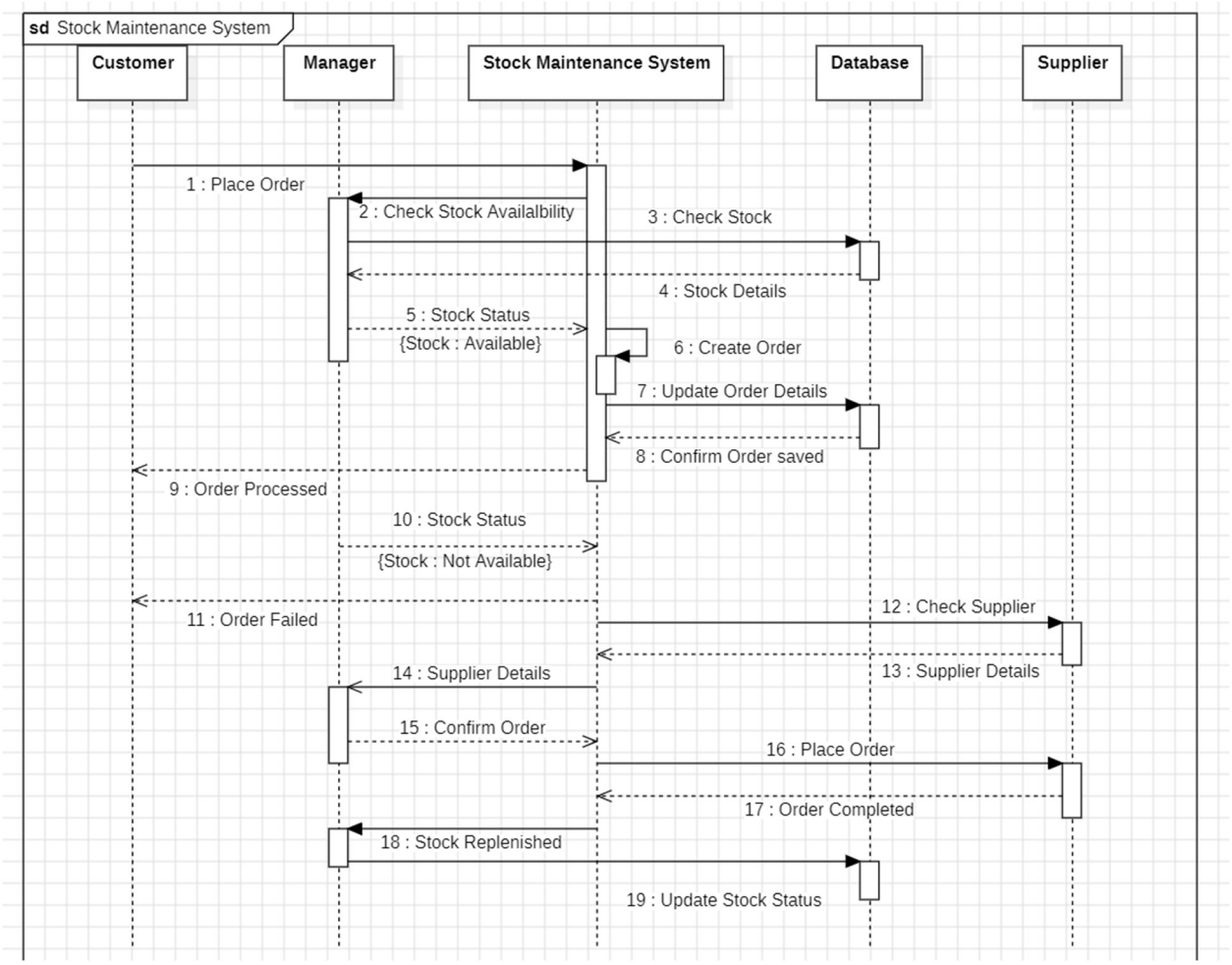


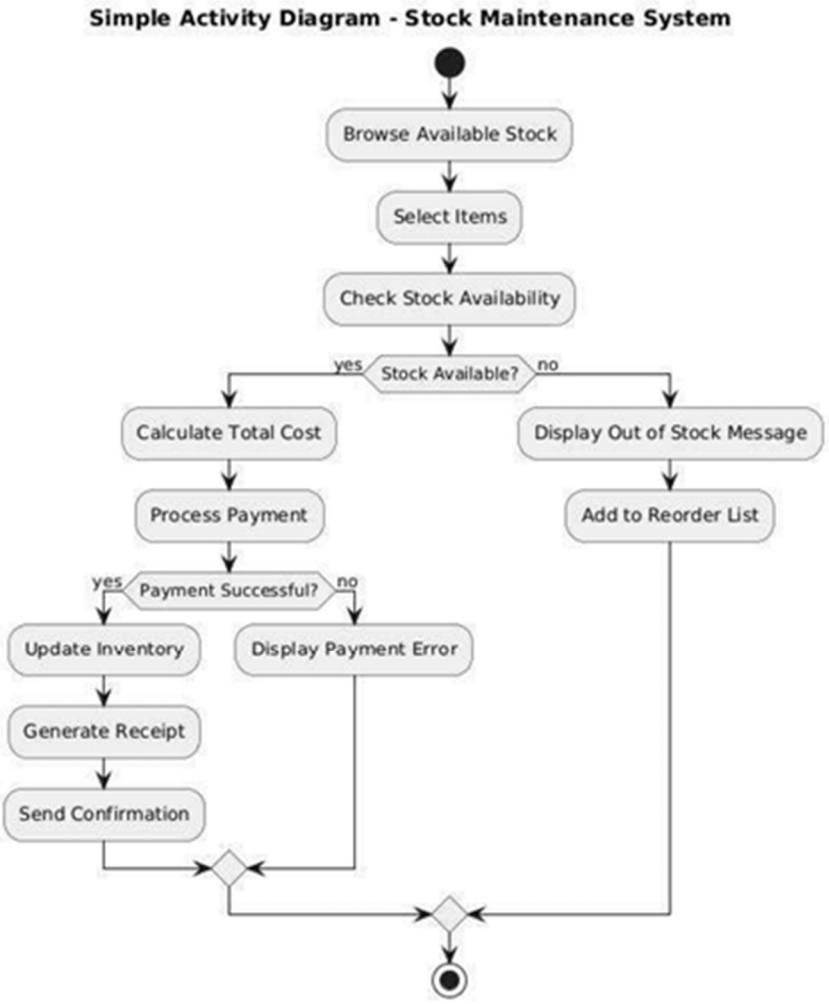
Fig 4.4 Stock Maintenance System - Sequence Diagram

The sequence diagram outlines the order fulfillment process in a stock maintenance system. It begins with the customer placing an order. The manager then checks stock availability, and the system verifies stock levels in the database. If stock is sufficient, the order is created and processed. If stock is insufficient, the system checks with suppliers, places orders, and updates stock levels once the replenishment is complete. The system communicates order status updates to the customer throughout the process. This diagram illustrates the interactions between the customer, manager, database, and suppliers, highlighting the steps involved in fulfilling an order effectively.

Activity Diagram

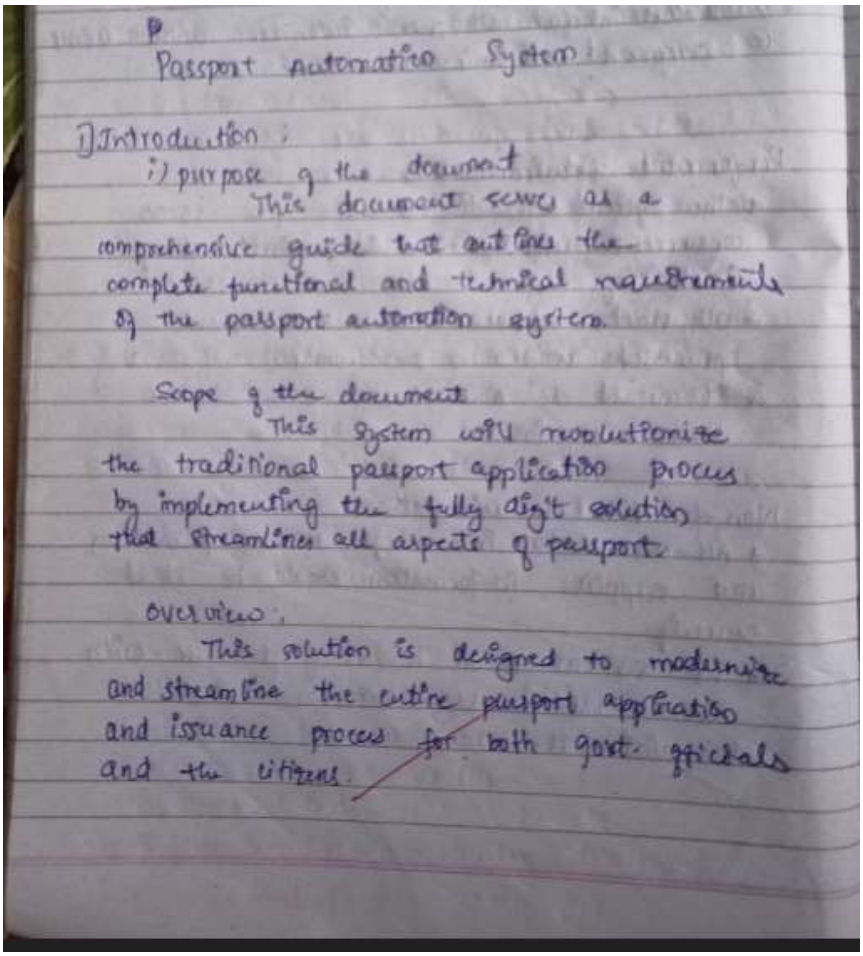
Fig 4.5 Stock Maintenance System - Activity Diagram

Fig 4.5 Stock Maintenance System - Activity Diagram



5. Passport Automation System

Software Requirement Specification



P

Passport Automation System

1) Introduction :

i) purpose of the document

This document serves as a comprehensive guide that outlines the complete functional and technical requirements of the passport automation system.

Scope of the document

This system will revolutionize the traditional passport application process by implementing the fully digit solution that streamlines all aspects of passport

overview :

This solution is designed to modernize and streamline the entire passport application and issuance process for both govt. officials and the citizens.

General discription:

The passport automation system is designed to serve a diverse user base including citizens seeking passport service, passport office personnel, verification officers, and system administrators.

Functional Requirements

* User Registration and login:
  Applicants must be able to register and login securely.

* Application Submission:
  Applicants should be able to fill out a passport application form, including personal details, address and other relevant info.

* Status Tracking:
  Tracking status such as "under review" Approved or Rejected.

* Notification System:
  Notifying user via SMS or email about the status or issue.

* Data validation and verification
  Interface for officers to verify application and automated validation of info

Interface Requirements:

* User Interface: A web based user friendly interface both for applicants and passport officers.

* Backend and Database: Secure storage and retrival of applicants data, status and document verificati

Performance Requirements:
* Scalability: Capable to handle 1000 concurrent users at peak usage times.
* Efficiency: Speed, It should load the interface in less than 2 seconds.
* Error Rate: Error free document uploads and validation.

Design Constraints:
→ Database Structure:
limitations on storing sensitive data securely with encryption.
* Using only government approved SMS and email providers for notification.

Non-functional Attributes
* Reliability: System availability must be 100%.
* Security: Ensure user authentication, data encryption and secure document storage
* portability: Compatible with modern browsers across windows, mac, android.
* Scalability: Easily scalable for future expansion.

Preliminary Budget and Schedule
⇒ 6 months - including
design - 1 month
development 4.5 months.
testing and deployment - 2 weeks.
Budget → ₹5,00,000 covering development testing, other service fees.

Class Diagram

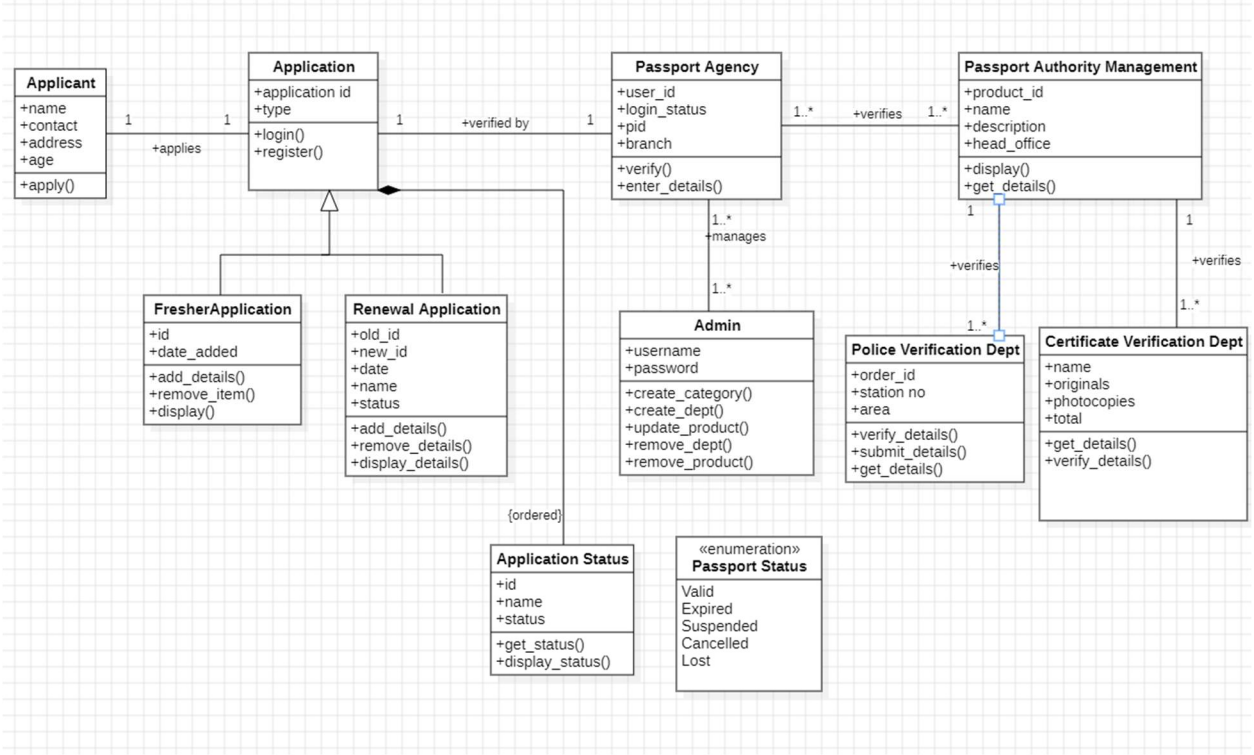


Fig 5.1 Passport Automation System - Class Diagram

The class diagram depicts the structure of a passport application and verification system. It illustrates various entities, such as Applicant, Application, and its specialized forms: FresherApplication and RenewalApplication. The Application class is associated with Applicant, who can apply and register for passport services. The system includes a Passport Agency and its management under Passport Authority Management, which oversees verification processes through departments like Police Verification and Certificate Verification. The diagram also involves an Admin class responsible for managing categories, departments, and products. Key features include status tracking through Application Status and Passport Status enumeration. Relationships between classes are depicted with multiplicity, inheritance, and composition, highlighting functionalities like verifying details, managing applications, and updating statuses

State Diagram

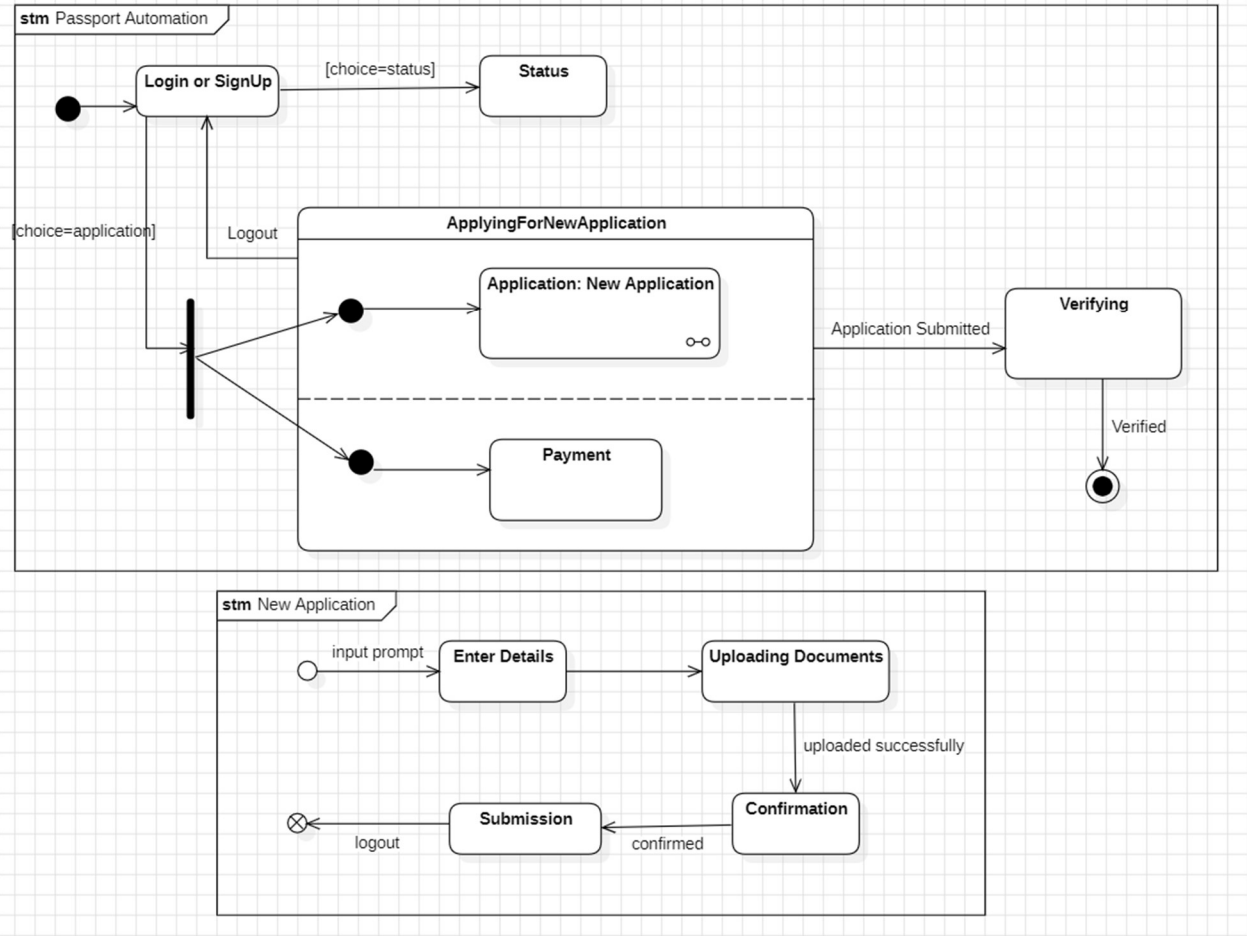


Fig 5.2 Passport Automation System - State Diagram

The state diagram illustrates the passport automation system. The system starts with the user logging in or signing up. After login, the user can choose to check the status of their application or apply for a new one. If the user chooses to apply, they enter the "ApplyingForNewApplication" state. Within this state, the user fills out the application form, uploads documents, and submits the application. Once submitted, the application enters the "Verifying" state. If the application is verified successfully, the user receives a confirmation. The user can also log out at any point during the process.

Use Case Diagram

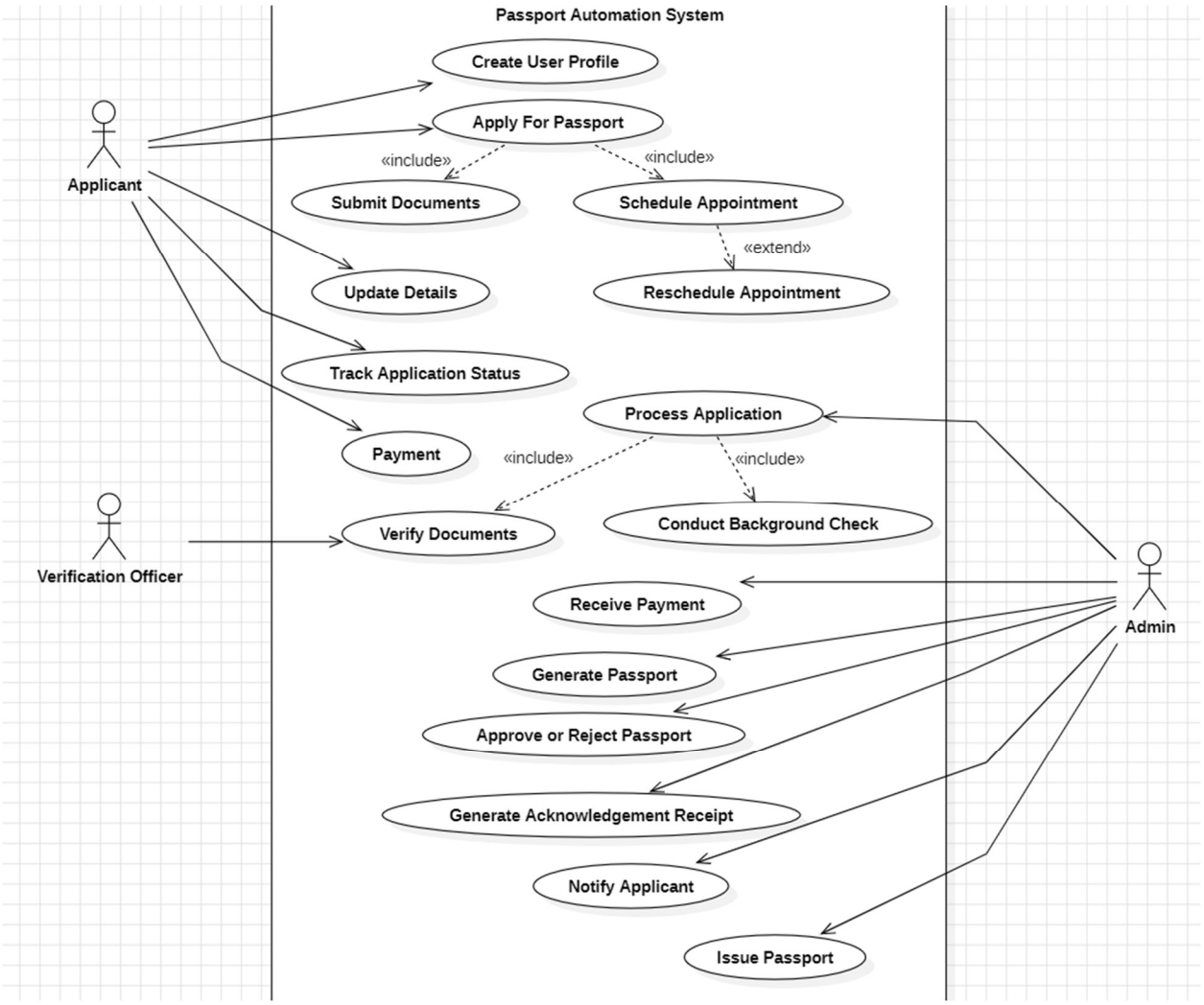


Fig 5.3 Passport Automation System - Use Case Diagram

The diagram illustrates a Use Case Diagram for a Passport Automation System, outlining the interactions between the system and its primary actors: Applicant, Verification Officer, and Admin. The Applicant begins by creating a user profile and applying for a passport, which includes submitting documents, scheduling (or rescheduling) appointments, making payments, and tracking application status. The Verification Officer is responsible for verifying documents and supporting the application processing. The Admin plays a key role in processing applications, conducting background checks, receiving payments, approving or rejecting passport requests, and issuing passports. Additional use cases include generating acknowledgments and notifying applicants of the application's status. This diagram effectively demonstrates the workflow and responsibilities of each actor in the passport issuance process.

Sequence Diagram

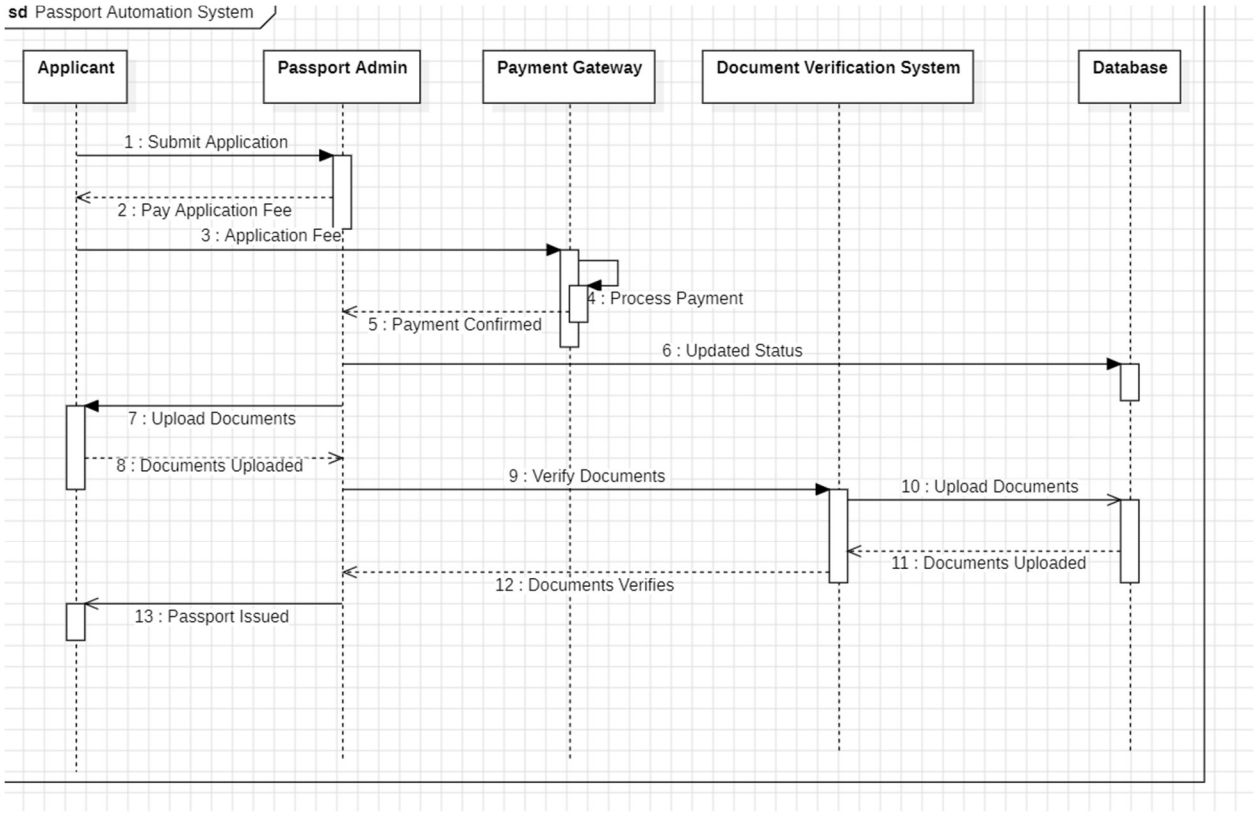


Fig 5.4 Passport Automation System - Sequence Diagram

The sequence diagram illustrates the process of applying for a passport. The applicant starts by submitting an application and then pays the application fee. The payment gateway processes the payment and updates the status. The applicant then uploads the required documents, which are verified by the document verification system. Once the documents are verified, the passport is issued to the applicant. This diagram shows the interactions between the applicant, passport admin, payment gateway, document verification system, and database throughout the passport application process.

Activity Diagram

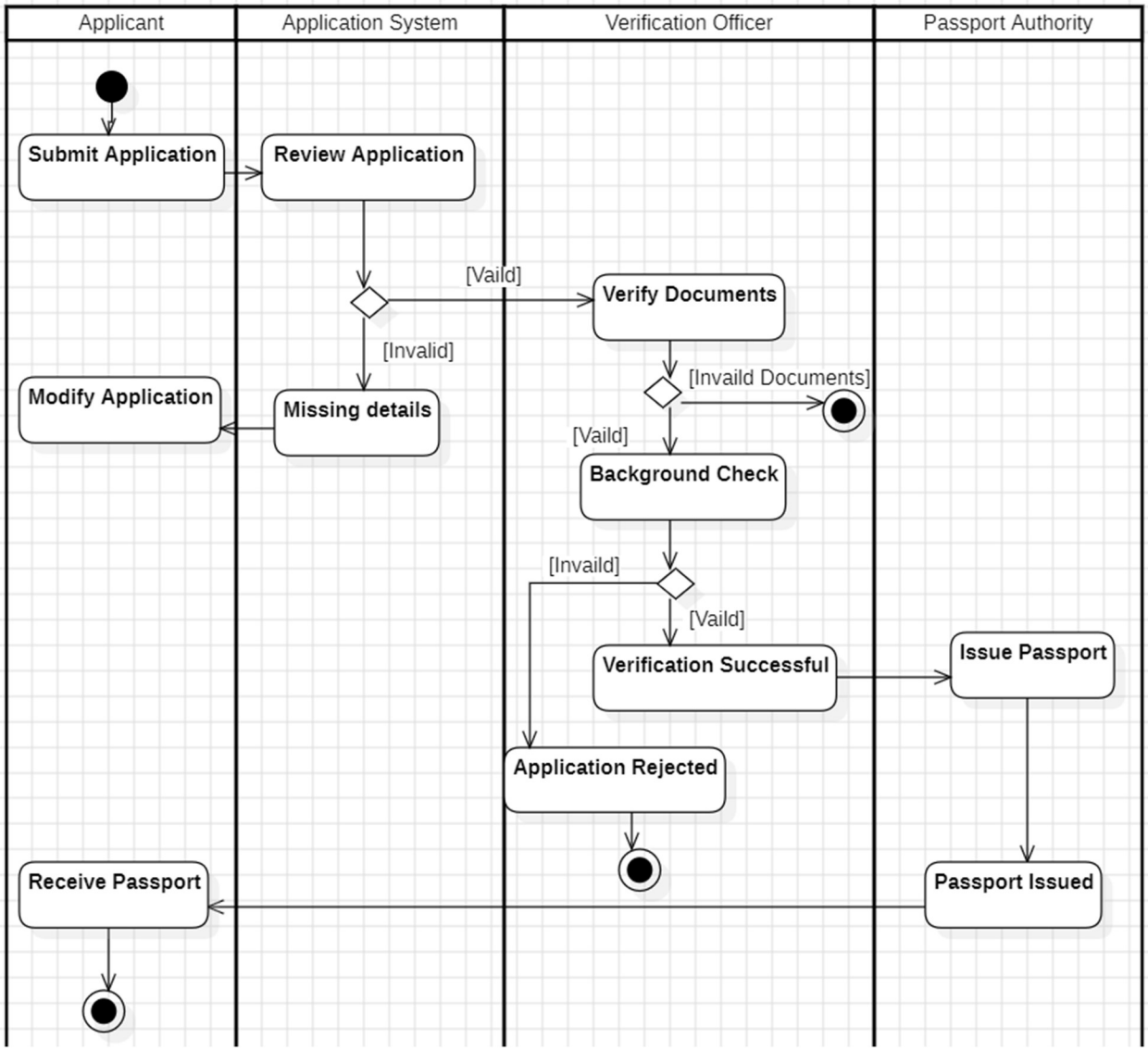


Fig 5.5 Passport Automation System - Activity Diagram

The activity diagram illustrates the passport application process. It starts with the applicant submitting an application. The application system reviews the application. If the application is complete, it proceeds to document verification. If invalid documents are found, the application is rejected. If valid, a background check is conducted. If the background check is clear, the verification is successful, and the passport is issued. If any stage fails, the application is rejected. The applicant can receive the passport once it's issued.