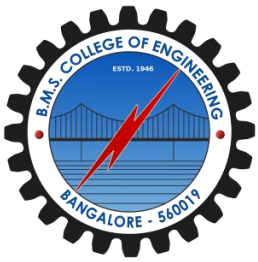
B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU



Lab Record

Software Engineering and Object-Oriented Modeling & Design

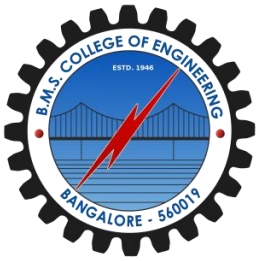
Submitted in partial fulfillment for the 6th Semester Laboratory

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:
Sanchay Agrawal
(1BM21CS186)

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,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the **Software Engineering and Object-Oriented Modelling and Design (22CS6PCSEO) laboratory** has been carried out by **Sanchay Agrawal (1BM21CS186)** during the 6th Semester Mar - June-2024.

Signature of the Faculty Incharge:
Dr. Latha N.R.
Assistant Professor
Department of Computer Science and Engineering
B.M.S. College of Engineering, Bangalore

Table Of Contents

S.No.	Title		Page No.
1	**Course Outcomes**		**1**
2	**Application Problems**		**1 - 36**
	2.1	**Hotel Management System**	**1 - 8**
		2.1.1 Software Requirement Specification	**1**
		2.1.2 Class Diagram	**5**
		2.1.3 State Diagram	**6**
		2.1.4 Use-Case Diagram	**7**
		2.1.5 Activity Diagram	**7**
		2.1.6 Sequence Diagram	**8**
	2.2	**Credit Card Processing**	**9 - 15**
		2.2.1 Software Requirement Specification	**9**
		2.2.2 Class Diagram	**13**
		2.2.3 State Diagram	**14**
		2.2.4 Use-Case Diagram	**14**
		2.2.5 Activity Diagram	**15**
		2.2.6 Sequence Diagram	**15**
	2.3	**Library Management System**	**16 - 22**
		2.3.1 Software Requirement Specification	**16**
		2.3.2 Class Diagram	**20**
		2.3.3 State Diagram	**21**
		2.3.4 Use-Case Diagram	**21**
		2.3.5 Activity Diagram	**22**
		2.3.6 Sequence Diagram	**22**
	2.4	**Stock Maintenance System**	**23 - 29**

		2.4.1	Software Requirement Specification	**23**
		2.4.2	Class Diagram	**27**
		2.4.3	State Diagram	**28**
		2.4.4	Use-Case Diagram	**28**
		2.4.5	Activity Diagram	**29**
		2.4.6	Sequence Diagram	**29**
	2.5	**Passport Automation System**		**30 - 36**
		2.5.1	Software Requirement Specification	**30**
		2.5.2	Class Diagram	**34**
		2.5.3	State Diagram	**34**
		2.5.4	Use-Case Diagram	**35**
		2.5.5	Activity Diagram	**36**
		2.5.6	Sequence Diagram	**36**

1. Course Outcomes:

CO1: Apply Software Engineering practices and Object oriented Design techniques for a given problem.

CO2: Analyze the System requirements, various design and testing techniques for a given system.

CO3: Design a object oriented model for a given scenario.

CO4: Ability to conduct practical experiment to solve a given problem using Unified Modelling language.

2. Application Problems

2.1 Hotel Management System:

2.1.1 Software Requirement Specification

1. Introduction:

1.1 Purpose of this Document:

The purpose of this document is to provide a detailed description of the Hotel Management System (HMS). It outlines the various requirements and specifications necessary for the development and implementation of the system. This document aims to ensure that the software development team and stakeholders have a clear understanding of the system's functionalities, constraints, and performance expectations.

1.2 Scope of this Document:

The scope of this document encompasses the entire Hotel Management System, including its objectives, user interactions, and expected outcomes. It provides value to the customer by detailing the development costs and time required, thereby facilitating better planning and resource allocation. The HMS aims to streamline hotel operations, enhance customer service, and improve overall management efficiency.

1.3 Overview:

The Hotel Management System is designed to automate and manage various aspects of hotel operations such as room booking, check-in/check-out processes, billing, and customer management. The system will offer an integrated solution to handle reservations, guest details, room availability, and payments. It aims to enhance operational efficiency and provide a seamless experience for both hotel staff and guests.

2. General Description:

2.1 Product Functions:

The HMS will provide the following functionalities:

- **Room Reservation:** Allows customers to book rooms online or through the hotel's front desk.
- **Check-In/Check-Out:** Manages the process of guest check-ins and check-outs, including room assignment and key management.

- **Billing and Payments:** Handles invoicing, payments, and receipt generation for guests.
- **Customer Management:** Maintains detailed records of guests, including personal information and stay history.
- **Inventory Management:** Tracks room availability, housekeeping status, and maintenance schedules.
- **Reporting:** Generates various reports such as occupancy rates, revenue, and customer feedback.

2.2 User Characteristics:
The primary users of the HMS include hotel staff (front desk, housekeeping, management), and guests. Hotel staff will require training to efficiently use the system, while guests will interact with the system through a user-friendly interface for bookings and service requests.

1. Hotel Guests:
Tech-savvy or non-tech-savvy individuals booking rooms online or in person. They seek a seamless booking experience and quick service. Primarily use the system for making reservations, checking in/out, and accessing services during their stay.

2. Front Desk Staff:
Hotel employees with basic computer skills who manage day-to-day guest interactions and reservations. Use the system to check room availability, process check-ins/outs, handle guest inquiries, and manage bookings.

3. Housekeeping Staff:
Staff responsible for maintaining cleanliness and order in guest rooms, typically requiring minimal system interaction. Use the system to update room statuses (e.g., clean, dirty, in maintenance) and receive notifications for housekeeping tasks.

4. Hotel Management:
Experienced personnel responsible for overall hotel operations, with a focus on efficiency, profitability, and customer satisfaction. Utilize the system for generating reports, monitoring performance metrics, managing staff, and making strategic decisions.

5. Maintenance Staff:
Staff in charge of the hotel's physical upkeep, including repairs and maintenance of facilities. Use the system to receive and track maintenance requests, update task statuses, and ensure timely resolution of issues.

6. IT Support Staff:
Technically skilled individuals responsible for maintaining and troubleshooting the HMS. Manage system updates, ensure data security, handle technical issues, and support other users with system-related problems.

2.3 Features and Benefits:
The HMS offers the following features and benefits:
- Streamlined booking process, reducing manual errors.
- Improved customer satisfaction through efficient service delivery.
- Real-time inventory management, optimizing room utilization.
- Comprehensive reporting tools for better decision-making.

- Enhanced data security and privacy for guest information.

3. Functional Requirements:

3.1 Room Reservation:
- The system shall allow users to search for room availability based on dates, room types, and occupancy.
- The system shall enable customers to make reservations and receive confirmation via email.
- The system shall support modifications and cancellations of reservations.

3.2 Check-In/Check-Out:
- The system shall manage guest check-in by assigning rooms and issuing key cards.
- The system shall handle guest check-out by processing payments and generating invoices.
- The system shall update room status post check-out for housekeeping.

3.3 Billing and Payments:
- The system shall generate bills including room charges, additional services, and taxes.
- The system shall support multiple payment methods (credit card, debit card, cash, digital wallets).
- The system shall provide guests with detailed receipts upon payment.

3.4 Customer Management:
- The system shall store and manage guest information securely.
- The system shall track guest preferences and history to personalize services.
- The system shall allow staff to access guest information as needed.

3.5 Inventory Management:
- The system shall track room availability in real-time.
- The system shall manage housekeeping schedules and room statuses.
- The system shall notify management of maintenance requirements.

3.6 Reporting:
- The system shall generate daily, weekly, and monthly occupancy reports.
- The system shall produce revenue and financial reports.
- The system shall compile customer feedback and service quality reports.

4. Interface Requirements:

4.1 User Interface:
- The system shall have an intuitive user interface for hotel staff to manage reservations and guest services.
- The system shall provide a web-based interface for guests to make reservations online.
- The system shall include mobile compatibility for access on smartphones and tablets.

4.2 Software Interfaces:
- The system shall integrate with existing property management systems (PMS) and accounting software.
- The system shall support API integration for third-party booking platforms.
- The system shall use secure communication protocols for data exchange.

5. Performance Requirements:

5.1 Speed and Efficiency:

- The system shall handle up to 500 concurrent users without performance degradation.
- The system shall process reservation requests within 2 seconds.

5.2 Reliability and Availability:

- The system shall have an uptime of 99.9%.
- The system shall perform regular backups to prevent data loss.

5.3 Scalability:

- The system shall support scalability to accommodate future expansion and additional features.

6. Design Constraints:

6.1 Technical Constraints:

- The system shall be developed using a scalable and secure architecture.
- The system shall adhere to industry standards for data security and privacy.

6.2 Hardware and Software Limitations:

- The system shall be compatible with existing hotel hardware (e.g., key card systems).
- The system shall operate on common operating systems (Windows, Linux, macOS).

7. Non-Functional Attributes:

7.1 Security:

- The system shall use encryption for data transmission and storage.
- The system shall implement role-based access control to limit user permissions.

7.2 Portability:

- The system shall support data export in common formats (CSV, XML) for integration with other systems.
- The system shall be deployable on both on-premise and cloud environments.

7.3 Reliability:

- The system shall perform regular data integrity checks to ensure consistency.
- The system shall provide robust error handling and recovery mechanisms.

7.4 Usability:

- The system shall have a user-friendly interface with clear navigation.
- The system shall provide comprehensive help documentation and user training resources.

7.5 Reusability:

- The system shall use modular code design to facilitate future enhancements and maintenance.

7.6 Compatibility:

- The system shall be compatible with common web browsers (Chrome, Firefox, Safari).
- The system shall integrate with major payment gateways and third-party services.

7.7 Data Integrity:

- The system shall ensure accurate and consistent data storage and retrieval.

- The system shall prevent unauthorized data access and modifications.

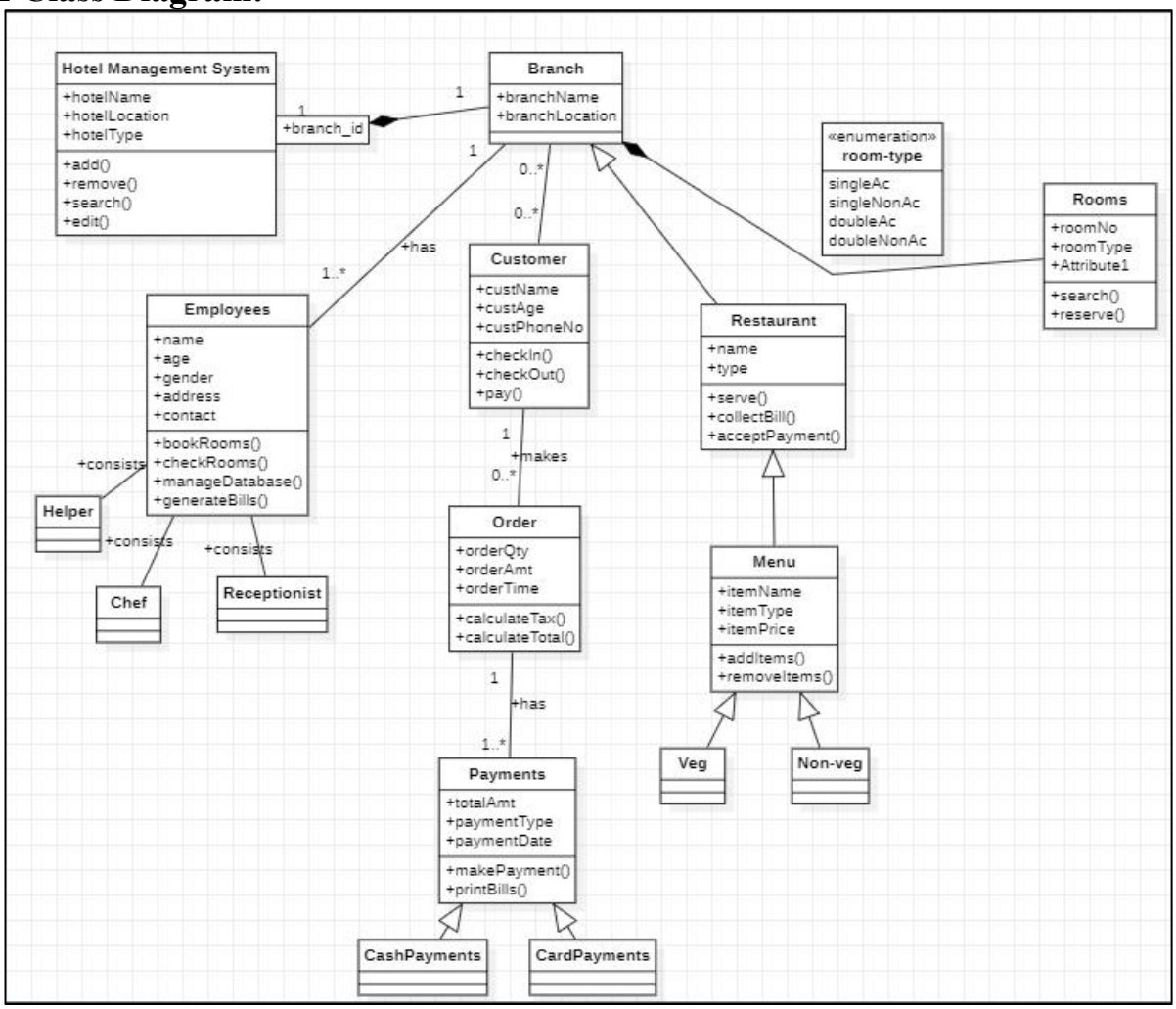
7.8 Scalability Capacity:

- The system shall support the addition of new features and increased user load without performance issues.

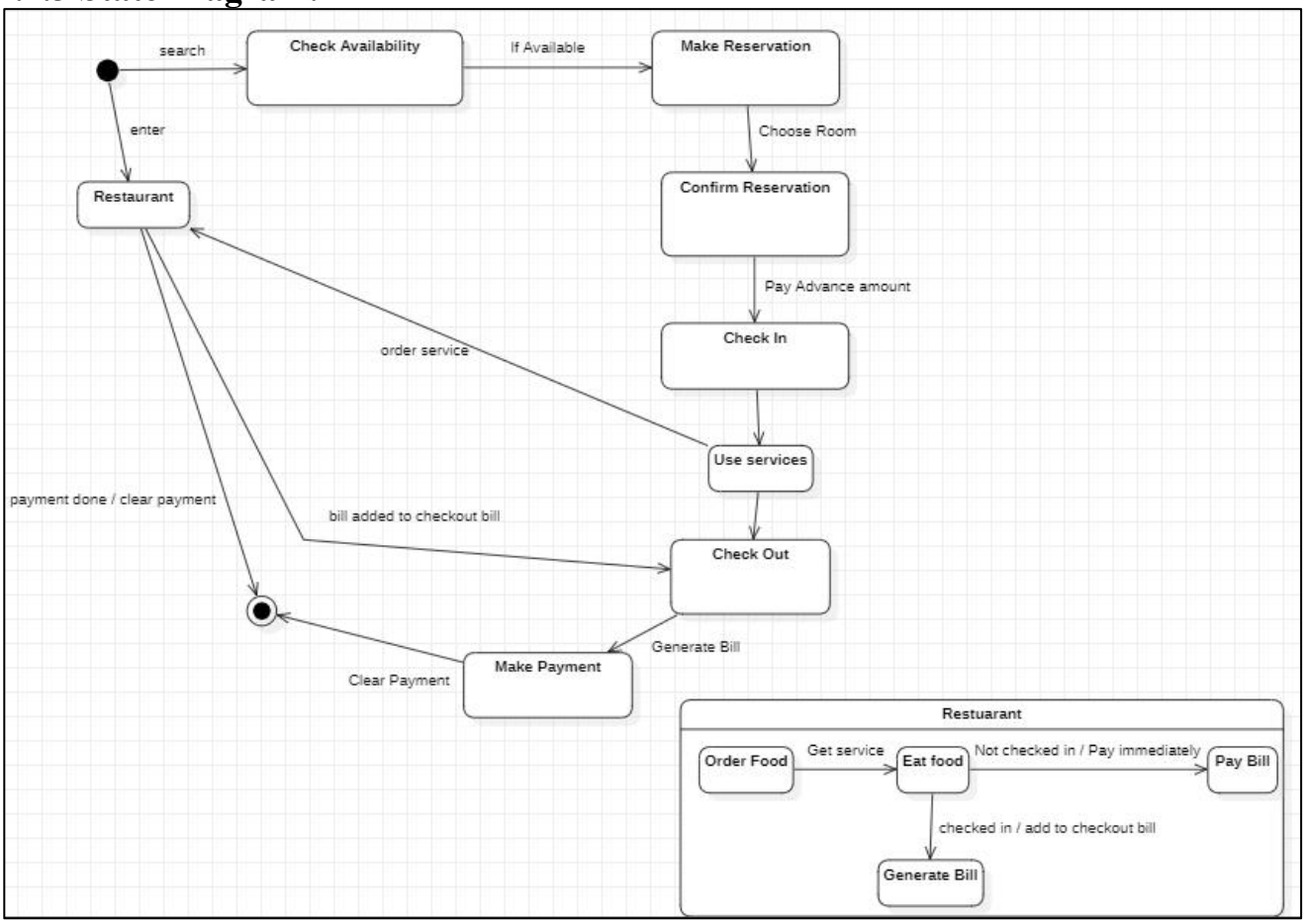
8. Preliminary Schedule and Budget:

The development of the HMS is estimated to take approximately 10 months with a budget of Rs. 12,00,000. This includes development, testing, deployment and maintenance cost.

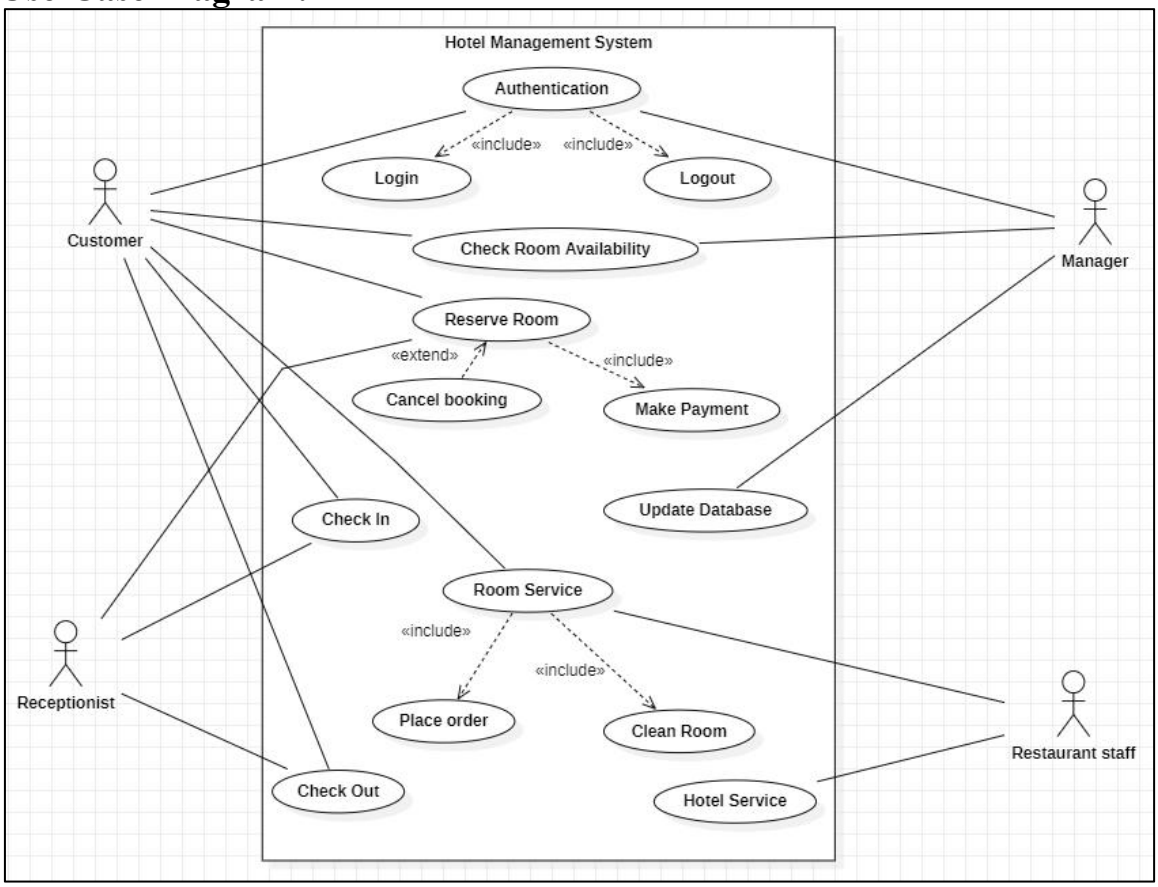
2.1.2 Class Diagram:



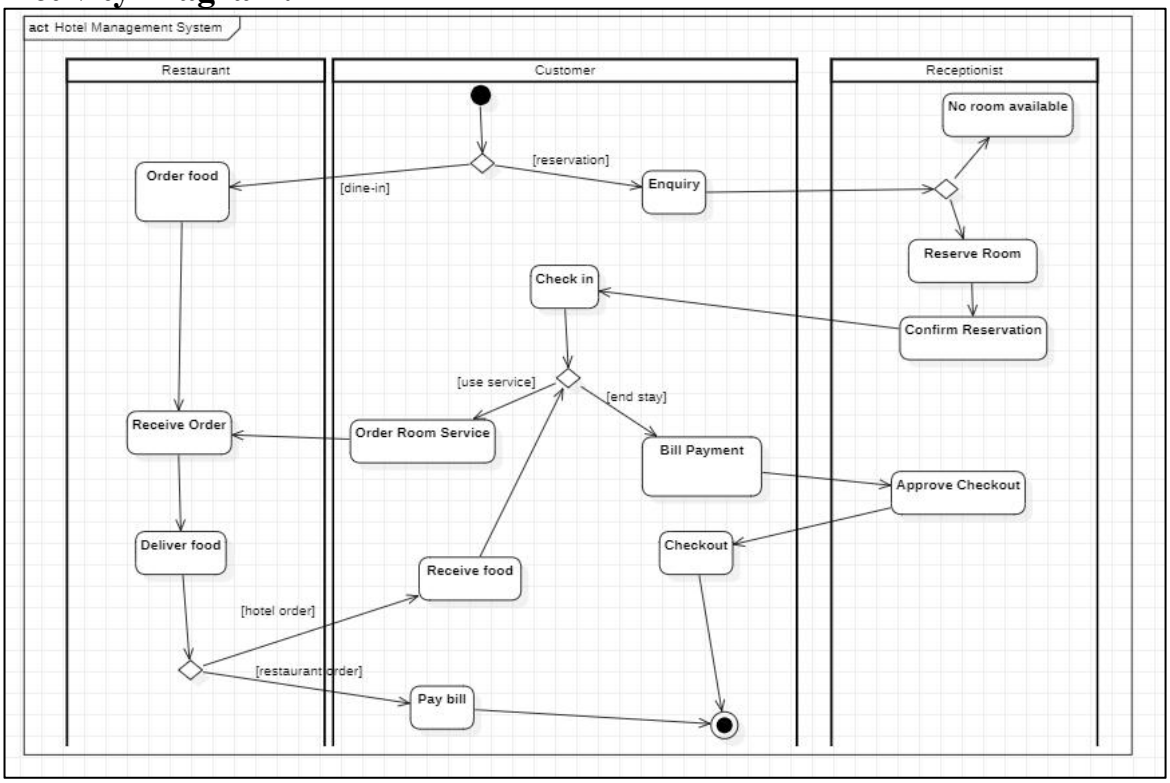
2.1.3 State Diagram:



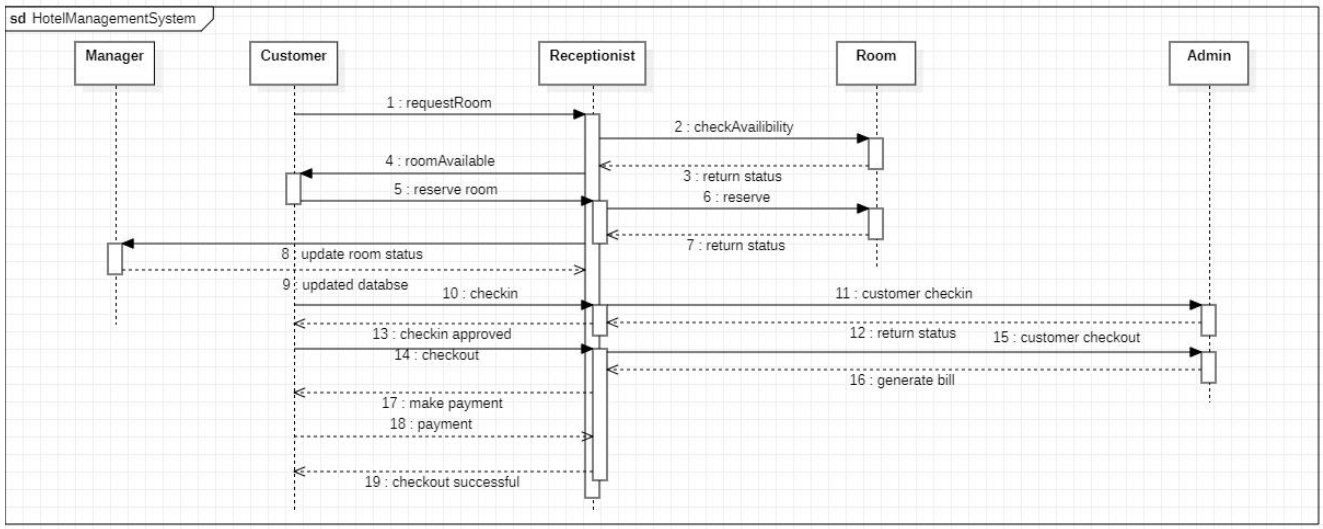
2.1.4 Use-Case Diagram:



2.1.5 Activity Diagram:



2.1.6 Sequence Diagram:



2.2 Credit Card Processing:

2.2.1 Software Requirement Specification

1. Introduction:

1.1 Purpose of this Document:

The purpose of this document is to provide a comprehensive description of the Credit Card Processing System. It outlines the system's functionalities, performance requirements, and design constraints. The document aims to ensure that all stakeholders, including developers, customers, and end-users, have a clear understanding of the system's requirements and expectations.

1.2 Scope of this Document:

The scope of this document covers the entire Credit Card Processing System, detailing its objectives, functionalities, user interactions, and anticipated outcomes. It provides value to the customer by outlining the development costs and time required, facilitating effective planning and resource allocation. The system aims to process credit card transactions securely and efficiently, ensuring compliance with industry standards.

1.3 Overview:

The Credit Card Processing System is designed to handle transactions, authorizations, settlements, and chargebacks for credit card payments. It provides an integrated solution for merchants to accept payments, manage customer information, and ensure secure transactions. The system aims to enhance transaction efficiency, improve security, and offer robust reporting capabilities.

2. General Description:

2.1 Product Functions:

The Credit Card Processing System will provide the following functionalities:

- **Transaction Processing:** Authorize, capture, and settle credit card transactions.
- **Fraud Detection:** Implement measures to detect and prevent fraudulent transactions.
- **Customer Management:** Maintain customer information and transaction history.
- **Reporting:** Generate reports on transactions, settlements, and chargebacks.
- **Compliance:** Ensure compliance with PCI DSS (Payment Card Industry Data Security Standard).

2.2 User Characteristics:

The primary users of the Credit Card Processing System include merchants, customers, payment gateway administrators, and financial institutions. These users will interact with the system through secure, user-friendly interfaces tailored to their specific needs.

1. Merchants:

Business owners or employees with basic to advanced computer skills, responsible for managing sales and transactions. Use the system to process transactions, generate reports, and manage customer information.

2. Customers:

Individuals making purchases with credit cards, generally seeking a secure and quick transaction process. Interact with the system to complete payments and view transaction history.

3. Payment Gateway Administrators:
Technically skilled personnel responsible for managing the payment gateway infrastructure and ensuring smooth operation. Use the system to monitor transactions, manage system settings, and troubleshoot issues.

4. Financial Institutions:
Banking professionals responsible for transaction settlement and reconciliation, requiring access to transaction data and reports. Use the system to ensure accurate settlement of transactions and compliance with financial regulations.

2.3 Features and Benefits:
The system offers the following features and benefits:
- **Secure Transactions:** Enhanced security features to protect sensitive cardholder information.
- **Efficient Processing:** Fast and reliable transaction processing to minimize delays.
- **Comprehensive Reporting:** Detailed reports to help merchants manage their business and comply with regulations.
- **Fraud Prevention:** Advanced fraud detection mechanisms to safeguard against unauthorized transactions.

3. Functional Requirements:
3.1 Transaction Processing:
- **Authorization:** The system shall authorize credit card transactions in real-time.
- **Capture:** The system shall capture authorized transactions for settlement.
- **Settlement:** The system shall settle transactions with the acquiring bank.
- **Refunds and Charge-backs:** The system shall process refunds and handle charge-backs as needed.

3.2 Fraud Detection:
- **Real-Time Analysis:** The system shall analyze transactions in real-time for potential fraud.
- **Alerts:** The system shall generate alerts for suspicious transactions.
- **Blocking:** The system shall block transactions identified as fraudulent.

3.3 Customer Management:
- **Data Storage:** The system shall securely store customer information and transaction history.
- **Data Access:** The system shall provide authorized users with access to customer data.
- **Data Update:** The system shall allow updates to customer information as necessary.

3.4 Reporting:
- **Transaction Reports:** The system shall generate daily, weekly, and monthly transaction reports.
- **Settlement Reports:** The system shall generate reports on settled transactions.
- **Chargeback Reports:** The system shall generate reports on chargebacks and disputes.

3.5 Compliance:
- **PCI DSS Compliance:** The system shall comply with PCI DSS requirements to protect cardholder data.
- **Audit Trails:** The system shall maintain audit trails for all transactions and access to data.

4. Interface Requirements:

4.1 User Interface:

- **Merchant Portal:** The system shall provide a secure, web-based portal for merchants to manage transactions and view reports.
- **Customer Interface:** The system shall provide an interface for customers to view their transaction history and manage their account details.
- **Admin Interface:** The system shall provide an interface for administrators to manage system settings and monitor performance.

4.2 Software Interfaces:

- **Payment Gateway Integration:** The system shall integrate with various payment gateways to process transactions.
- **Banking Systems:** The system shall interface with banking systems for settlement and reconciliation.
- **APIs:** The system shall provide APIs for third-party integration and custom applications.

5. Performance Requirements:

5.1 Speed and Efficiency:

- **Transaction Time:** The system shall process transactions within 2 seconds.
- **Concurrent Users:** The system shall support up to 1,000 concurrent users without performance degradation.

5.2 Reliability and Availability:

- **Uptime:** The system shall have an uptime of 99.9%.
- **Backups:** The system shall perform regular backups to prevent data loss.

5.3 Scalability:

- **Load Handling:** The system shall scale to handle increased transaction volumes during peak times.
- **Future Growth:** The system shall be designed to accommodate future growth and additional features.

6. Design Constraints:

6.1 Technical Constraints:

- **Development Framework:** The system shall be developed using secure, scalable frameworks and technologies.
- **Security Protocols:** The system shall implement industry-standard security protocols for data transmission and storage.

6.2 Hardware and Software Limitations:

- **Compatibility:** The system shall be compatible with existing merchant hardware and software systems.
- **Operating Systems:** The system shall operate on common operating systems (Windows, Linux, macOS).

7. Non-Functional Attributes:

7.1 Security:

- **Encryption:** The system shall use strong encryption for data at rest and in transit.

- **Access Control:** The system shall implement role-based access control to restrict access to sensitive information.

7.2 Portability:
- **Data Export:** The system shall support data export in common formats (CSV, XML) for integration with other systems.
- **Deployment:** The system shall be deployable on both on-premise and cloud environments.

7.3 Reliability:
- **Data Integrity:** The system shall perform regular data integrity checks to ensure consistency.
- **Error Handling:** The system shall provide robust error handling and recovery mechanisms.

7.4 Usability:
- **User-Friendly Interface:** The system shall have a user-friendly interface with clear navigation and comprehensive help documentation.
- **Training Resources:** The system shall provide training resources for users to facilitate smooth adoption.

7.5 Reusability:
- **Modular Design:** The system shall use modular code design to facilitate future enhancements and maintenance.

7.6 Compatibility:
- **Browser Compatibility:** The system shall be compatible with common web browsers (Chrome, Firefox, Safari).
- **Payment Gateway Integration:** The system shall integrate with major payment gateways and financial institutions.

7.7 Data Integrity:
- **Accurate Data:** The system shall ensure accurate and consistent data storage and retrieval.
- **Unauthorized Access:** The system shall prevent unauthorized data access and modifications.

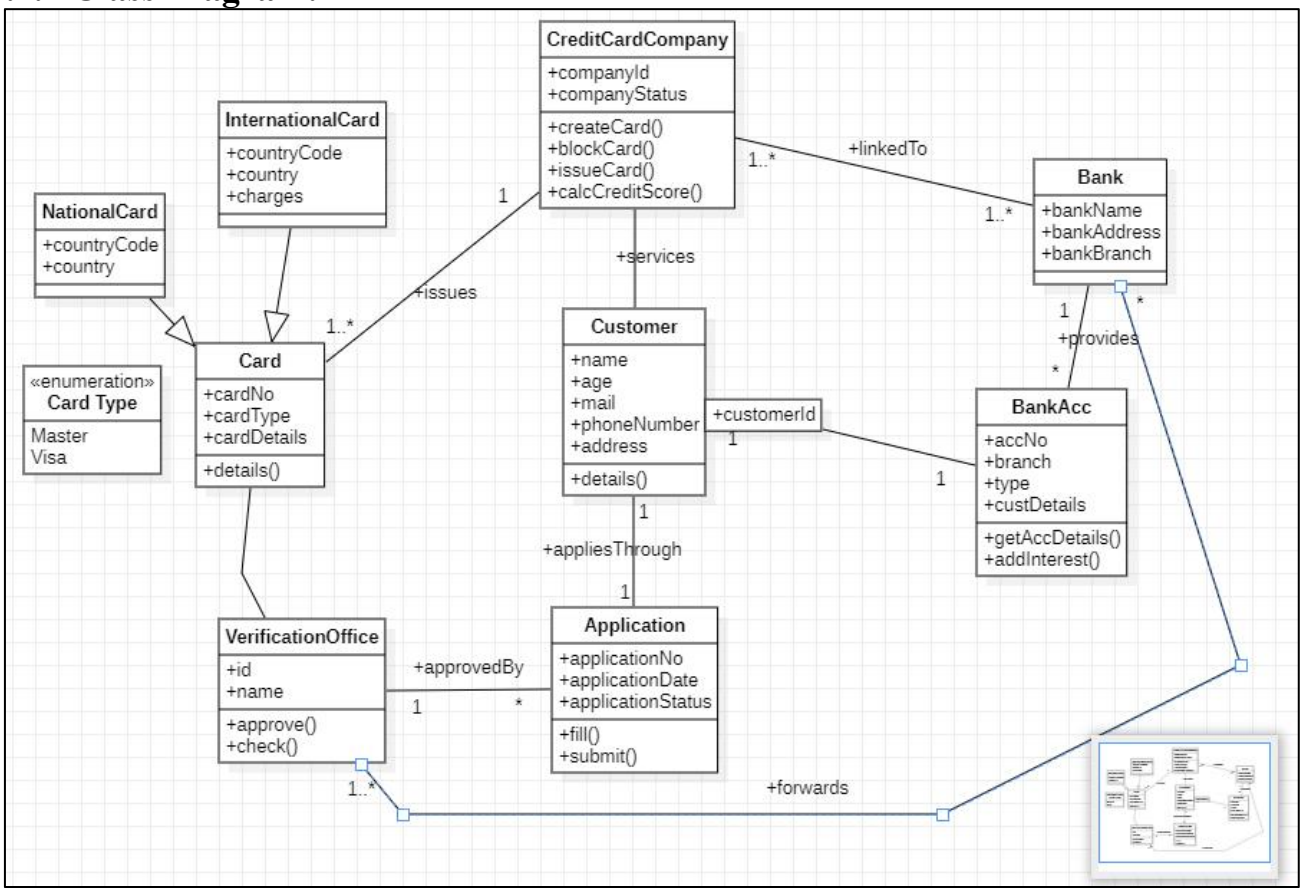
7.8 Scalability Capacity:
- **Feature Addition:** The system shall support the addition of new features and increased user load without performance issues.

8. Preliminary Schedule and Budget:

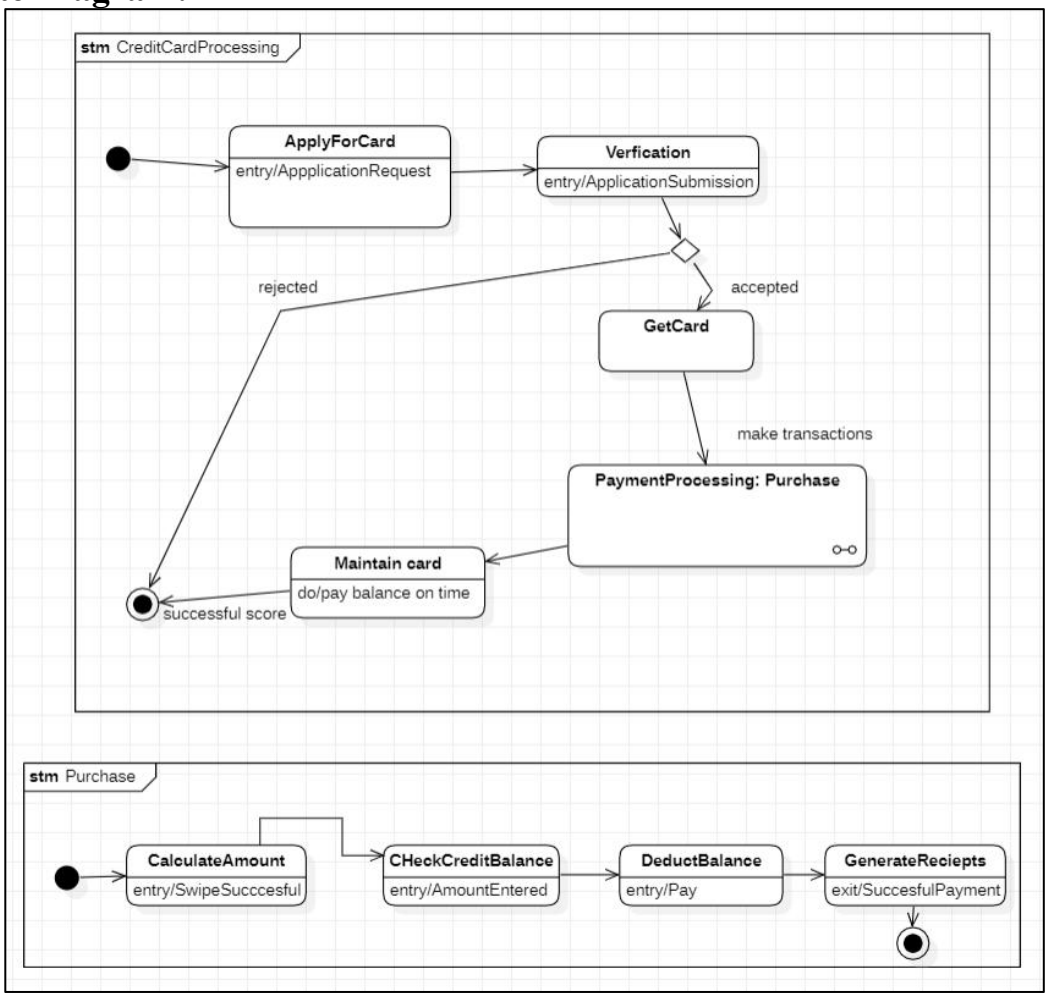
8.1 Schedule:

The development of the CCP system is estimated to take approximately 24 months with a budget of Rs. 10,00,000. This includes development, testing, deployment and maintenance cost.

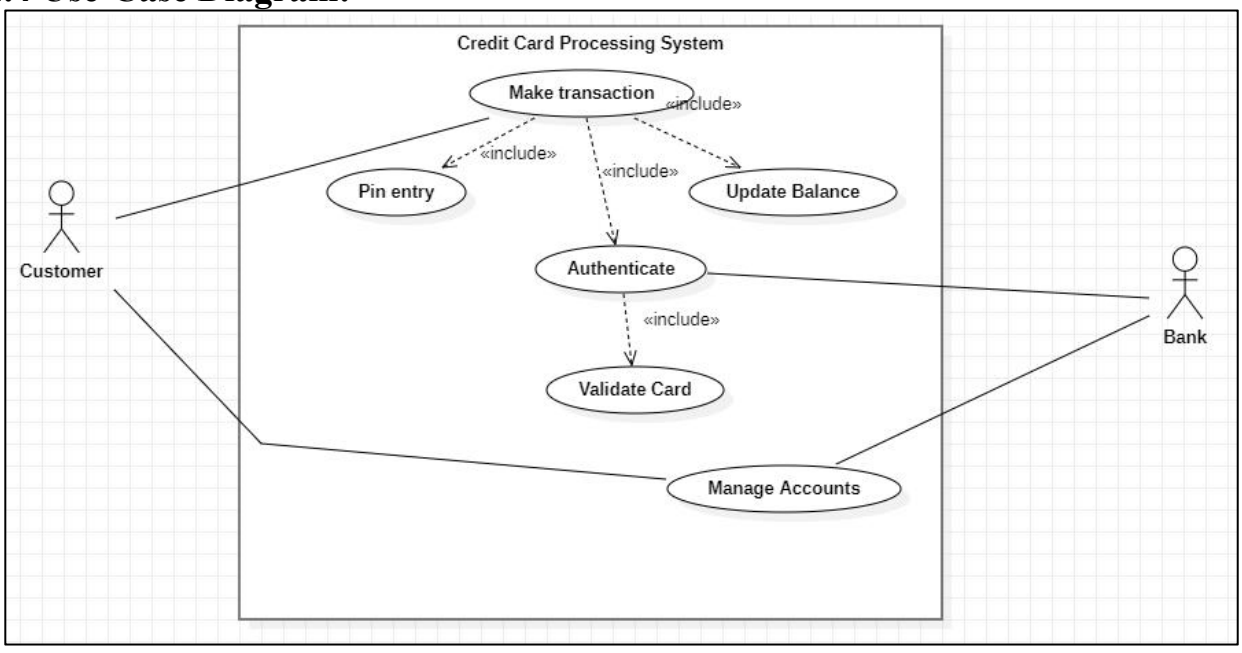
2.2.2 Class Diagram:



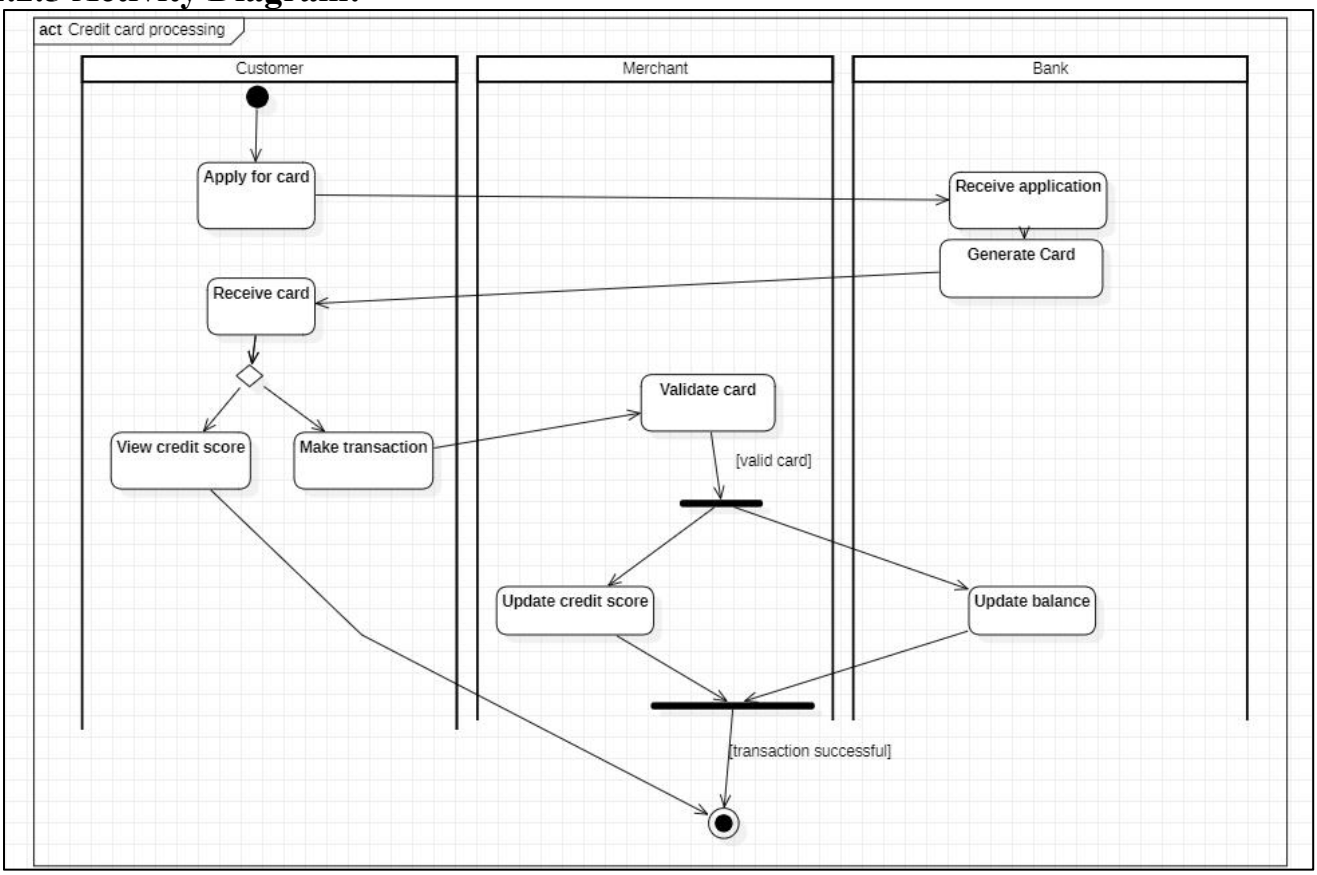
2.2.3 State Diagram:



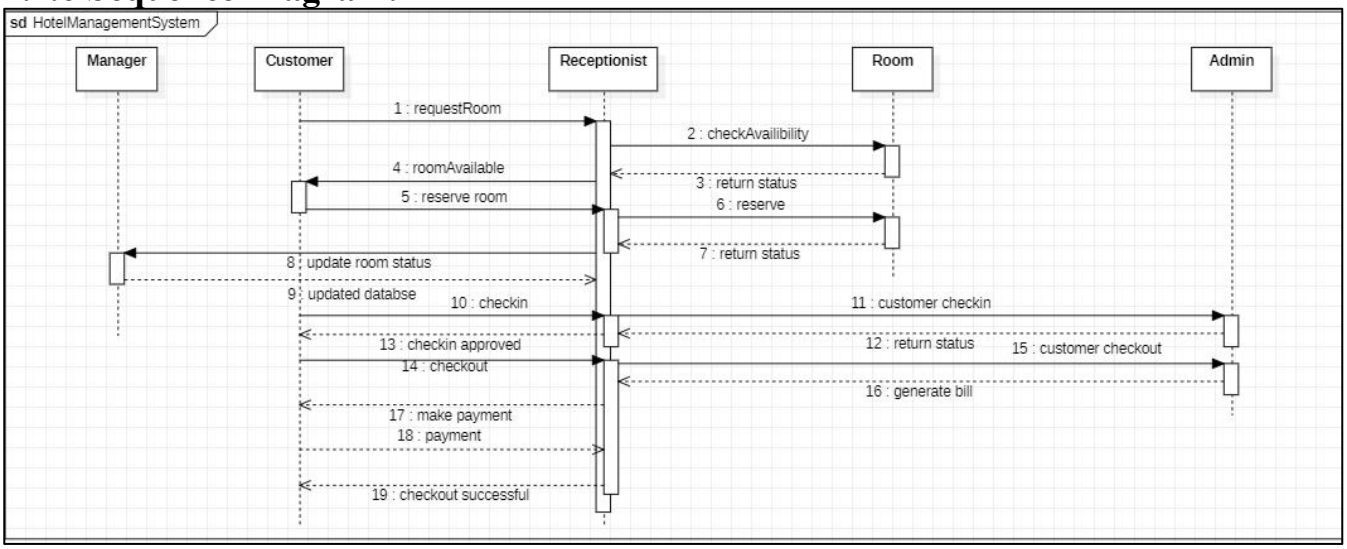
2.2.4 Use-Case Diagram:



2.2.5 Activity Diagram:



2.2.6 Sequence Diagram:



2.3 Library Management System:

2.3.1 Software Requirement Specification

1. Introduction:

1.1 Purpose of this Document:

The purpose of this document is to provide a detailed description of the Library Management System (LMS). It outlines the system's functionalities, performance requirements, and design constraints. The document aims to ensure that the development team, stakeholders, and end-users have a clear understanding of the system's requirements and expectations.

1.2 Scope of this Document:

The scope of this document covers the entire Library Management System, detailing its objectives, functionalities, user interactions, and anticipated outcomes. It provides value to the customer by detailing the development costs and time required, thereby facilitating better planning and resource allocation. The LMS aims to streamline library operations, improve user experience, and enhance the management of library resources.

1.3 Overview:

The Library Management System is designed to automate and manage various aspects of library operations such as book cataloging, user management, lending, returns, and fines. The system will provide an integrated solution to handle book inventory, user accounts, and transaction records, aiming to enhance operational efficiency and provide a seamless experience for both library staff and patrons.

2. General Description:

2.1 Product Functions:

The LMS will provide the following functionalities:

- **Catalog Management:** Catalog and manage books, journals, and other library materials.
- **User Management:** Handle user registrations, profiles, and authentication.
- **Transaction Management:** Manage borrowing, returning, and renewing of library materials.
- **Fine Calculation:** Calculate and manage overdue fines.
- **Search Functionality:** Allow users to search the catalog for available materials.
- **Reporting:** Generate reports on inventory, transactions, and user activity.

2.2 User Characteristics:

The primary users of the LMS include librarians, patrons (library members), and administrators. Each user group will interact with the system through secure, user-friendly interfaces tailored to their specific needs.

1. Librarians:

Library professionals with basic to advanced computer skills responsible for managing catalog and transactions. Use the system to catalog materials, process borrow and return transactions, and generate reports.

2. Patrons (Library Members):

Individuals who borrow and use library materials, generally seeking an easy-to-use interface for searching and managing their accounts. Interact with the system to search the catalog, borrow and renew items, and view their account status.

3. Administrators:
Technical staff responsible for system configuration, maintenance, and performance monitoring. Use the system to manage user access, configure settings, and ensure the system is running efficiently.

2.3 Features and Benefits:
The system offers the following features and benefits:
- **Efficient Cataloging:** Streamlined process for cataloging and managing library materials.
- **Enhanced User Experience:** User-friendly interface for easy searching and transaction management.
- **Accurate Record Keeping:** Automated transaction records to reduce errors and improve efficiency.
- **Comprehensive Reporting:** Detailed reports to help manage library operations and resources effectively.
- **Overdue Management:** Automated fine calculation to ensure timely returns and reduce overdue items.

3. Functional Requirements:
3.1 Catalog Management:
- **Book Entry:** The system shall allow librarians to add, update, and delete book records.
- **Classification:** The system shall support classification and categorization of library materials.
- **Inventory Tracking:** The system shall track the status and location of each item in the library.

3.2 User Management:
- **Registration:** The system shall allow users to register and create accounts.
- **Authentication:** The system shall authenticate users during login.
- **Profile Management:** The system shall allow users to update their personal information.

3.3 Transaction Management:
- **Borrowing:** The system shall allow users to borrow available materials.
- **Returning:** The system shall record the return of borrowed materials.
- **Renewing:** The system shall allow users to renew borrowed items.

3.4 Fine Calculation:
- **Overdue Tracking:** The system shall track overdue items.
- **Fine Calculation:** The system shall calculate fines based on overdue duration.
- **Payment Processing:** The system shall record fine payments made by users.

3.5 Search Functionality:
- **Catalog Search:** The system shall allow users to search the catalog by title, author, genre, and other criteria.
- **Advanced Search:** The system shall provide advanced search options for more specific queries.

3.6 Reporting:
- **Inventory Reports:** The system shall generate reports on current inventory and catalog status.
- **Transaction Reports:** The system shall generate reports on borrowing, returning, and overdue items.
- **User Activity Reports:** The system shall generate reports on user activity and transactions.

4. Interface Requirements:

4.1 User Interface:

- **Librarian Interface:** The system shall provide a secure, web-based interface for librarians to manage catalog and transactions.
- **Patron Interface:** The system shall provide a user-friendly interface for patrons to search the catalog, borrow items, and manage their accounts.
- **Admin Interface:** The system shall provide an interface for administrators to configure system settings and monitor performance.

4.2 Software Interfaces:

- **Database Integration:** The system shall integrate with a database to store and retrieve catalog and transaction data.
- **Authentication Services:** The system shall integrate with authentication services for user login and management.
- **API Integration:** The system shall provide APIs for third-party integration and custom applications.

5. Performance Requirements:

5.1 Speed and Efficiency:

- **Search Time:** The system shall process search queries within 3 seconds.
- **Transaction Processing:** The system shall process borrowing and returning transactions within 2 seconds.

5.2 Reliability and Availability:

- **Uptime:** The system shall have an uptime of 99.9%.
- **Backups:** The system shall perform regular backups to prevent data loss.

5.3 Scalability:

- **Load Handling:** The system shall scale to handle increased user and transaction volumes during peak times.
- **Future Growth:** The system shall be designed to accommodate future growth and additional features.

6. Design Constraints:

6.1 Technical Constraints:

- **Development Framework:** The system shall be developed using secure, scalable frameworks and technologies.
- **Security Protocols:** The system shall implement industry-standard security protocols for data transmission and storage.

6.2 Hardware and Software Limitations:

- **Compatibility:** The system shall be compatible with existing library hardware and software systems.
- **Operating Systems:** The system shall operate on common operating systems (Windows, Linux, macOS).

7. Non-Functional Attributes:

7.1 Security:

- **Encryption:** The system shall use strong encryption for data at rest and in transit.
- **Access Control:** The system shall implement role-based access control to restrict access to sensitive information.

7.2 Portability:

- **Data Export:** The system shall support data export in common formats (CSV, XML) for integration with other systems.
- **Deployment:** The system shall be deployable on both on-premise and cloud environments.

7.3 Reliability:

- **Data Integrity:** The system shall perform regular data integrity checks to ensure consistency.
- **Error Handling:** The system shall provide robust error handling and recovery mechanisms.

7.4 Usability:

- **User-Friendly Interface:** The system shall have a user-friendly interface with clear navigation and comprehensive help documentation.
- **Training Resources:** The system shall provide training resources for users to facilitate smooth adoption.

7.5 Reusability:

- **Modular Design:** The system shall use modular code design to facilitate future enhancements and maintenance.

7.6 Compatibility:

- **Browser Compatibility:** The system shall be compatible with common web browsers (Chrome, Firefox, Safari).
- **Integration:** The system shall integrate with major library management tools and software.

7.7 Data Integrity:

- **Accurate Data:** The system shall ensure accurate and consistent data storage and retrieval.
- **Unauthorized Access:** The system shall prevent unauthorized data access and modifications.

7.8 Scalability Capacity:

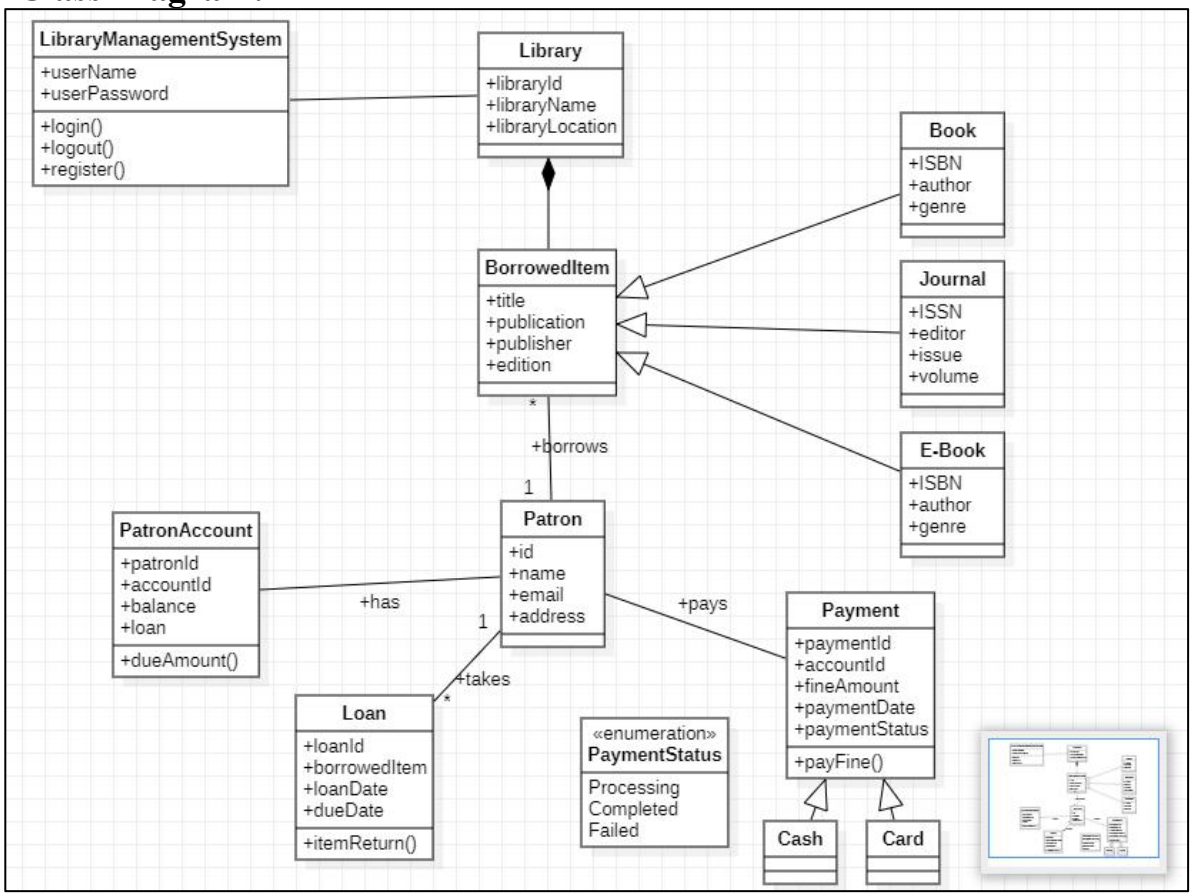
- **Feature Addition:** The system shall support the addition of new features and increased user load without performance issues.

8. Preliminary Schedule and Budget:

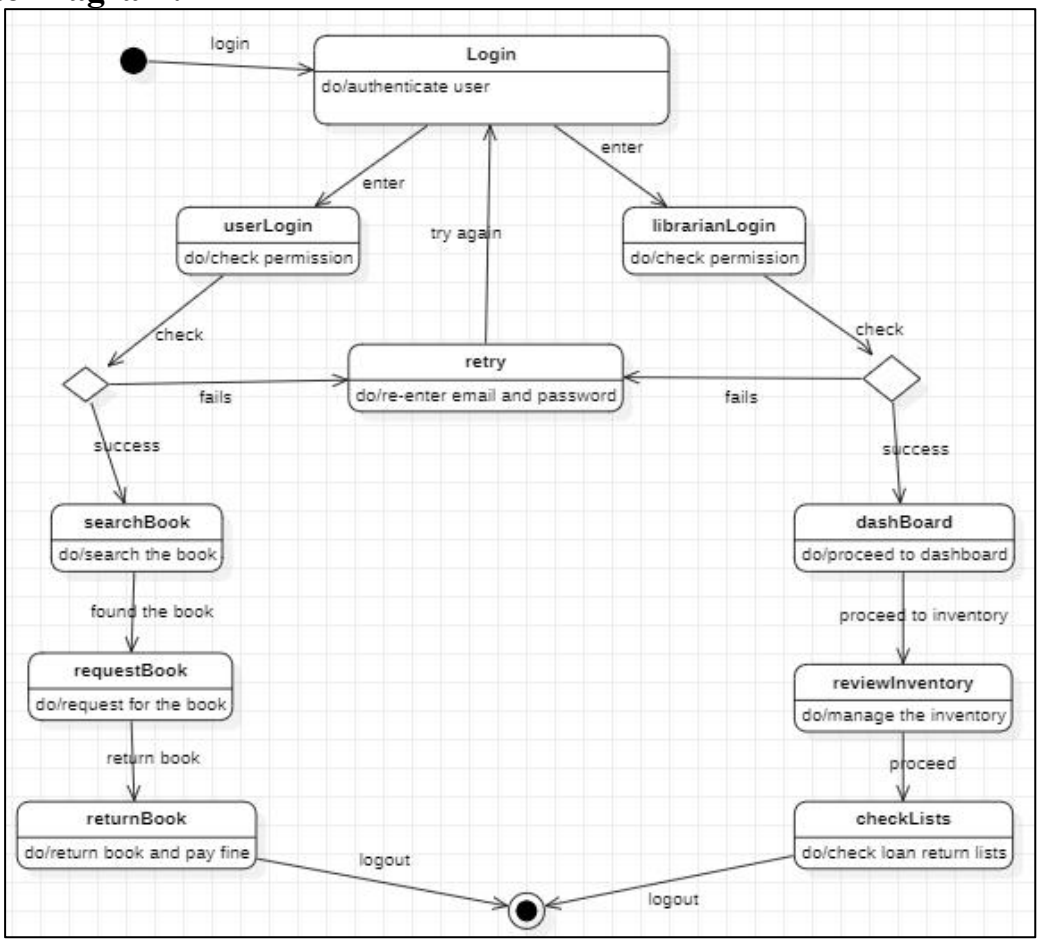
8.1 Schedule:

The development of the LMS is estimated to take approximately 5 months with a budget of Rs. 5,00,000. This includes development, testing, deployment and maintenance cost.

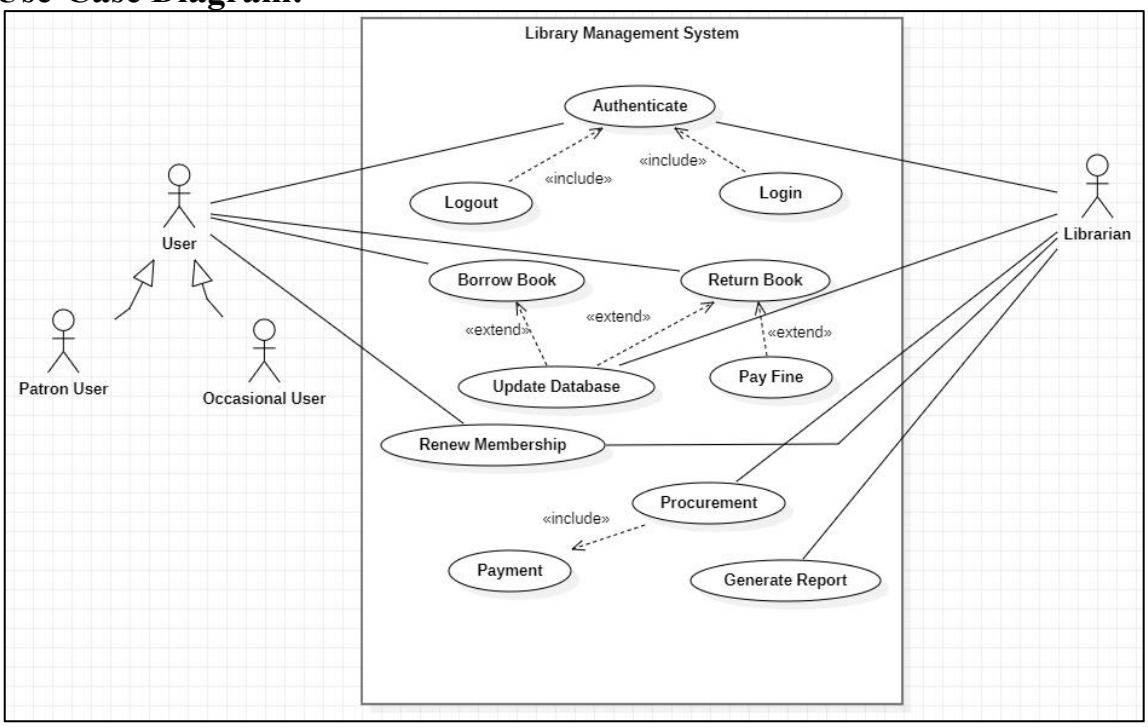
2.3.2 Class Diagram:



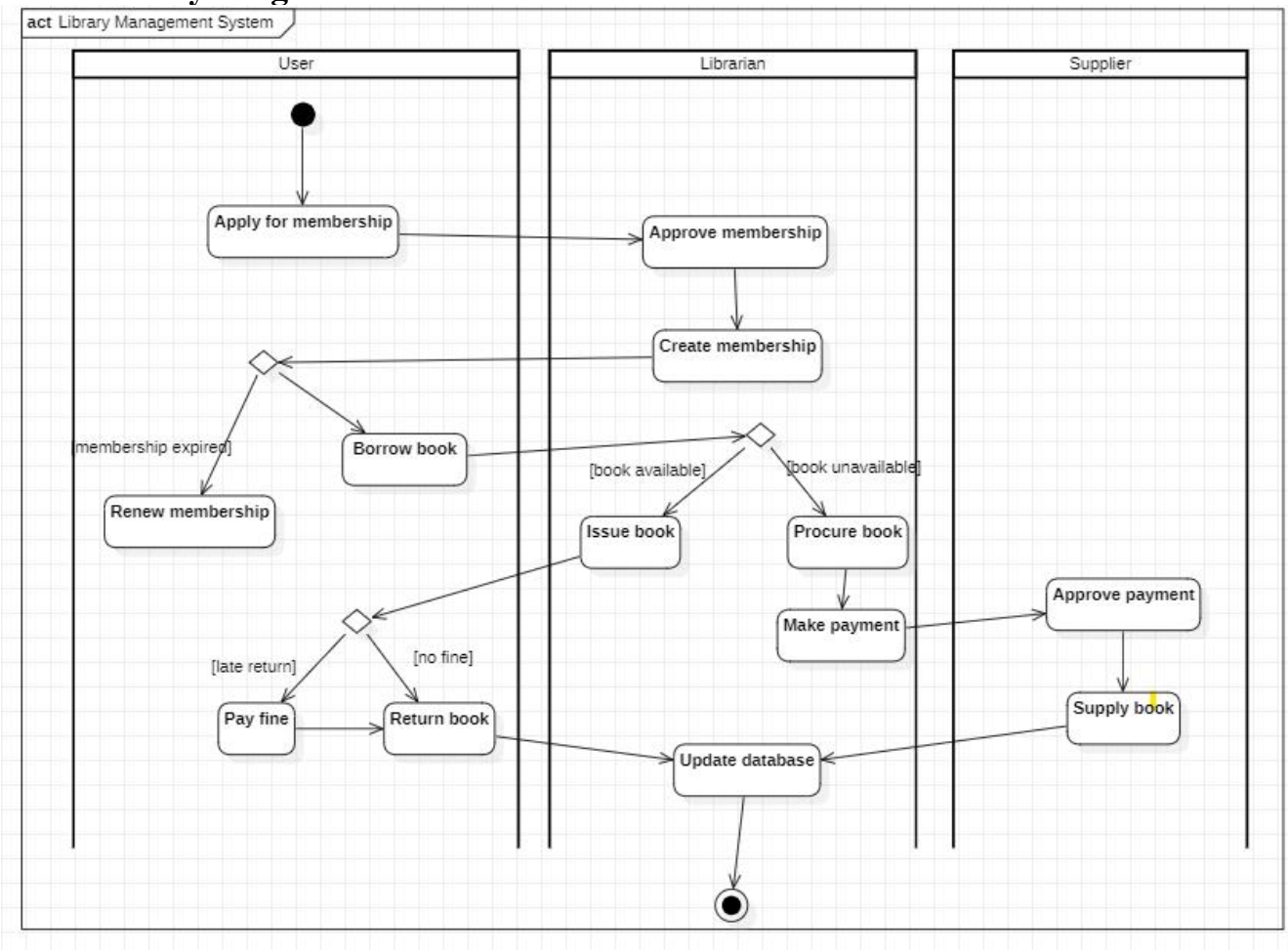
2.3.3 State Diagram:



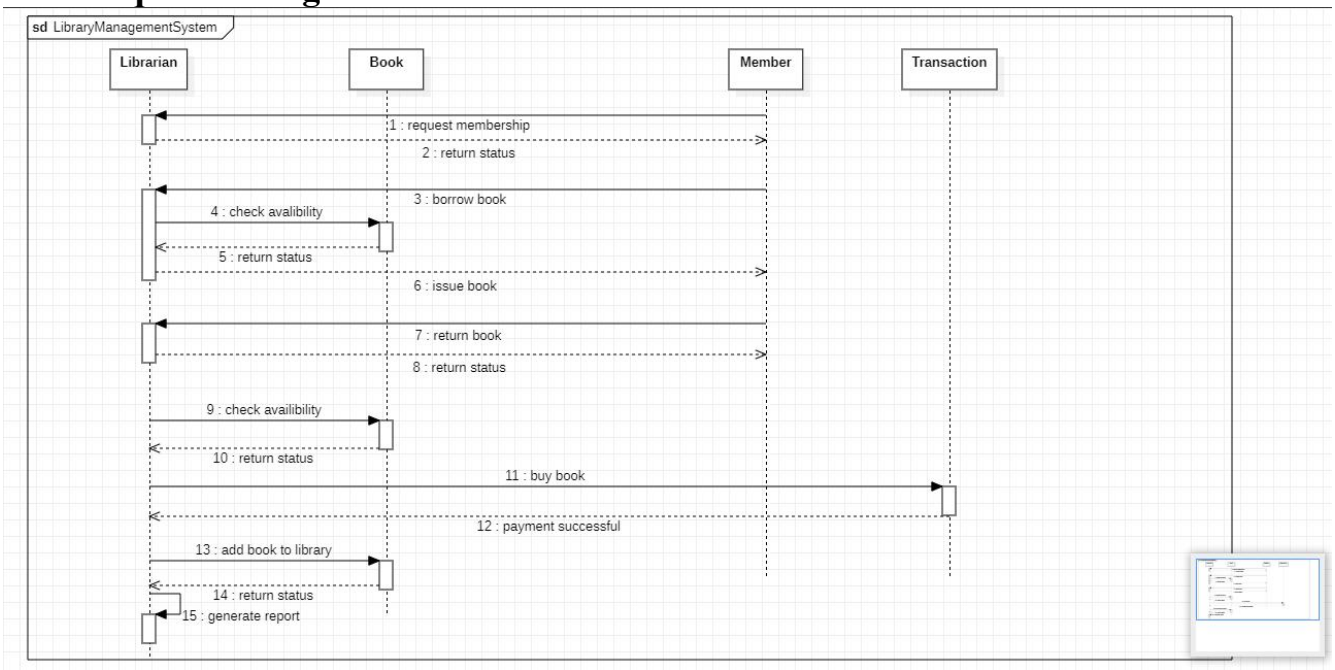
2.3.4 Use-Case Diagram:



2.3.5 Activity Diagram:



2.3.6 Sequence Diagram:



2.4 Stock Maintenance System:

2.4.1 Software Requirement Specification

1. Introduction:

1.1 Purpose of this Document:

The purpose of this document is to provide a comprehensive description of the Stock Maintenance System (SMS). It outlines the system's functionalities, performance requirements, and design constraints. This document aims to ensure that the development team, stakeholders, and end-users have a clear understanding of the system's requirements and expectations.

1.2 Scope of this Document:

The scope of this document covers the entire Stock Maintenance System, detailing its objectives, functionalities, user interactions, and anticipated outcomes. It provides value to the customer by detailing the development costs and time required, thereby facilitating better planning and resource allocation. The SMS aims to streamline stock management operations, improve user experience, and enhance the management of inventory resources.

1.3 Overview:

The Stock Maintenance System is designed to automate and manage various aspects of stock and inventory management such as item cataloging, stock tracking, purchase orders, and reporting. The system will provide an integrated solution to handle stock levels, user access, and transaction records, aiming to enhance operational efficiency and provide a seamless experience for both inventory managers and end-users.

2. General Description:

2.1 Product Functions:

The SMS will provide the following functionalities:

- **Catalog Management:** Catalog and manage stock items, including details like descriptions, quantities, and locations.
- **User Management:** Handle user registrations, profiles, and authentication.
- **Stock Tracking:** Monitor stock levels, movements, and adjustments.
- **Purchase Order Management:** Manage purchase orders, including creation, approval, and tracking.
- **Reporting:** Generate reports on stock levels, transactions, and user activity.

2.2 User Characteristics:

The primary users of the SMS include inventory managers, staff members, and administrators. Each user group will interact with the system through secure, user-friendly interfaces tailored to their specific needs.

1. Inventory Managers:

Professionals with intermediate to advanced knowledge of inventory management processes and software applications. Use the system to monitor stock levels, create purchase orders, and generate reports to ensure efficient inventory management.

2. Staff Members:
Personnel with basic to intermediate computer skills responsible for handling daily stock transactions and updates. Interact with the system to update stock records, process stock movements, and assist in stock-taking activities.

3. Administrators:
Technical staff responsible for configuring system settings, maintaining security protocols, and ensuring system performance. Use the system to manage user access, configure system parameters, and monitor overall system health and performance.

2.3 Features and Benefits:
The system offers the following features and benefits:
- **Efficient Cataloging:** Streamlined process for cataloging and managing stock items.
- **Enhanced User Experience:** User-friendly interface for easy tracking and management of stock levels.
- **Accurate Record Keeping:** Automated transaction records to reduce errors and improve efficiency.
- **Comprehensive Reporting:** Detailed reports to help manage inventory operations and resources effectively.
- **Purchase Order Management:** Automated purchase order process to ensure timely restocking and reduce shortages.

3. Functional Requirements:
3.1 Catalog Management:
- **Item Entry:** The system shall allow users to add, update, and delete stock item records.
- **Classification:** The system shall support classification and categorization of stock items.
- **Inventory Tracking:** The system shall track the status and location of each item in the inventory.

3.2 User Management:
- **Registration:** The system shall allow users to register and create accounts.
- **Authentication:** The system shall authenticate users during login.
- **Profile Management:** The system shall allow users to update their personal information.

3.3 Stock Tracking:
- **Stock Monitoring:** The system shall monitor and display current stock levels.
- **Stock Movements:** The system shall record and track stock movements, including additions, removals, and transfers.
- **Stock Adjustments:** The system shall allow authorized users to adjust stock levels as needed.

3.4 Purchase Order Management:
- **Order Creation:** The system shall allow users to create new purchase orders.
- **Approval Process:** The system shall provide an approval process for purchase orders.
- **Order Tracking:** The system shall track the status of purchase orders from creation to fulfillment.

3.5 Reporting:
- **Stock Reports:** The system shall generate reports on current stock levels and inventory status.
- **Transaction Reports:** The system shall generate reports on stock transactions and movements.
- **User Activity Reports:** The system shall generate reports on user activity and transactions.

4. Interface Requirements:

4.1 User Interface:

- **Manager Interface:** The system shall provide a secure, web-based interface for managers to oversee inventory and transactions.
- **Staff Interface:** The system shall provide a user-friendly interface for staff to manage stock levels and process transactions.
- **Admin Interface:** The system shall provide an interface for administrators to configure system settings and monitor performance.

4.2 Software Interfaces:

- **Database Integration:** The system shall integrate with a database to store and retrieve inventory and transaction data.
- **Authentication Services:** The system shall integrate with authentication services for user login and management.
- **API Integration:** The system shall provide APIs for third-party integration and custom applications.

5. Performance Requirements:

5.1 Speed and Efficiency:

- **Search Time:** The system shall process search queries within 3 seconds.
- **Transaction Processing:** The system shall process stock transactions within 2 seconds.

5.2 Reliability and Availability:

- **Uptime:** The system shall have an uptime of 99.9%.
- **Backups:** The system shall perform regular backups to prevent data loss.

5.3 Scalability:

- **Load Handling:** The system shall scale to handle increased user and transaction volumes during peak times.
- **Future Growth:** The system shall be designed to accommodate future growth and additional features.

6. Design Constraints:

6.1 Technical Constraints:

- **Development Framework:** The system shall be developed using secure, scalable frameworks and technologies.
- **Security Protocols:** The system shall implement industry-standard security protocols for data transmission and storage.

6.2 Hardware and Software Limitations:

- **Compatibility:** The system shall be compatible with existing hardware and software systems used in stock management.
- **Operating Systems:** The system shall operate on common operating systems (Windows, Linux, macOS).

7. Non-Functional Attributes:

7.1 Security:

- **Encryption:** The system shall use strong encryption for data at rest and in transit.

- **Access Control:** The system shall implement role-based access control to restrict access to sensitive information.

7.2 Portability:
- **Data Export:** The system shall support data export in common formats (CSV, XML) for integration with other systems.
- **Deployment:** The system shall be deployable on both on-premise and cloud environments.

7.3 Reliability:
- **Data Integrity:** The system shall perform regular data integrity checks to ensure consistency.
- **Error Handling:** The system shall provide robust error handling and recovery mechanisms.

7.4 Usability:
- **User-Friendly Interface:** The system shall have a user-friendly interface with clear navigation and comprehensive help documentation.
- **Training Resources:** The system shall provide training resources for users to facilitate smooth adoption.

7.5 Reusability:
- **Modular Design:** The system shall use modular code design to facilitate future enhancements and maintenance.

7.6 Compatibility:
- **Browser Compatibility:** The system shall be compatible with common web browsers (Chrome, Firefox, Safari).
- **Integration:** The system shall integrate with major inventory management tools and software.

7.7 Data Integrity:
- **Accurate Data:** The system shall ensure accurate and consistent data storage and retrieval.
- **Unauthorized Access:** The system shall prevent unauthorized data access and modifications.

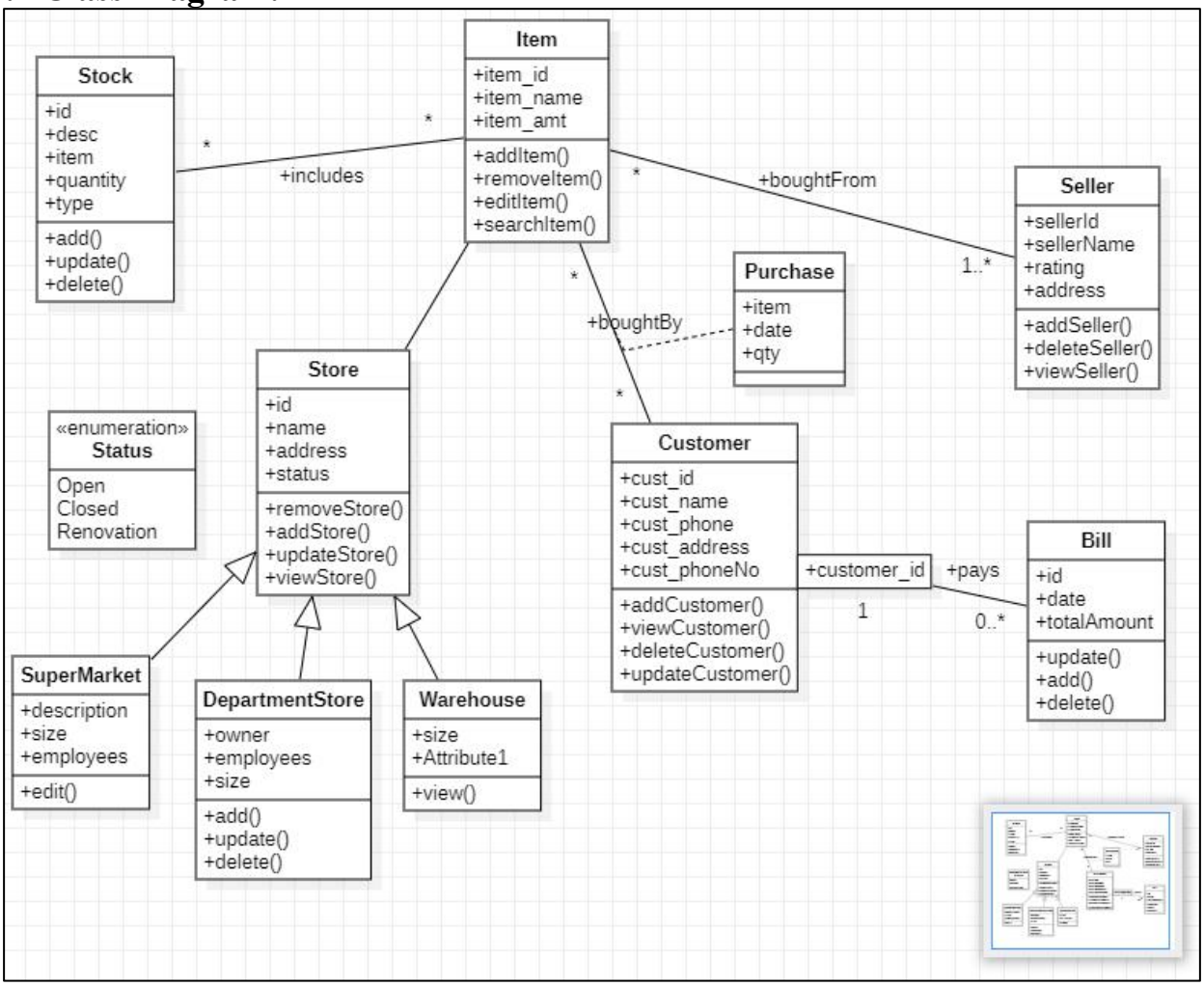
7.8 Scalability Capacity:
- **Feature Addition:** The system shall support the addition of new features and increased user load without performance issues.

8. Preliminary Schedule and Budget:

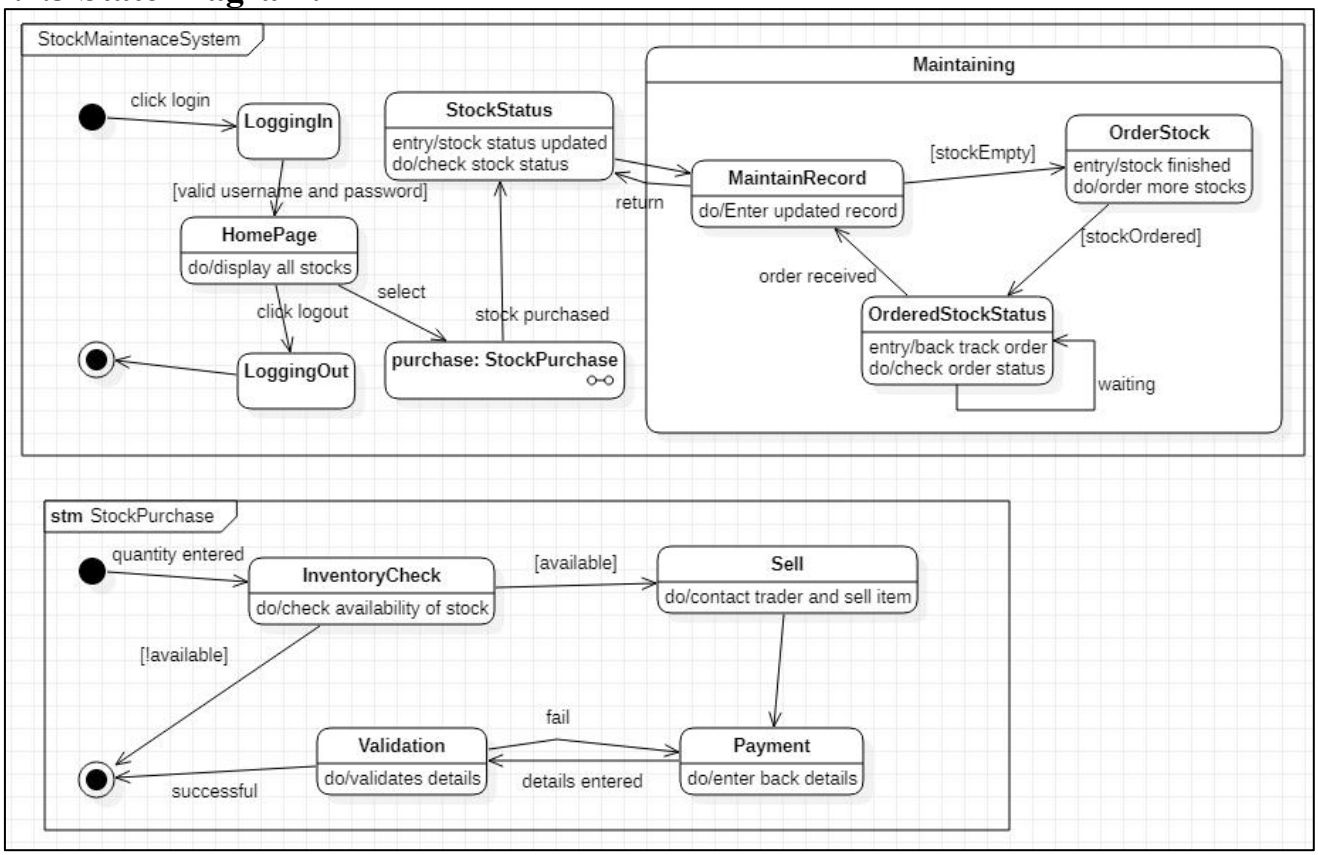
8.1 Schedule:

The development of the SMS is estimated to take approximately 15 months with a budget of Rs. 15,00,000. This includes development, testing, deployment and maintenance cost.

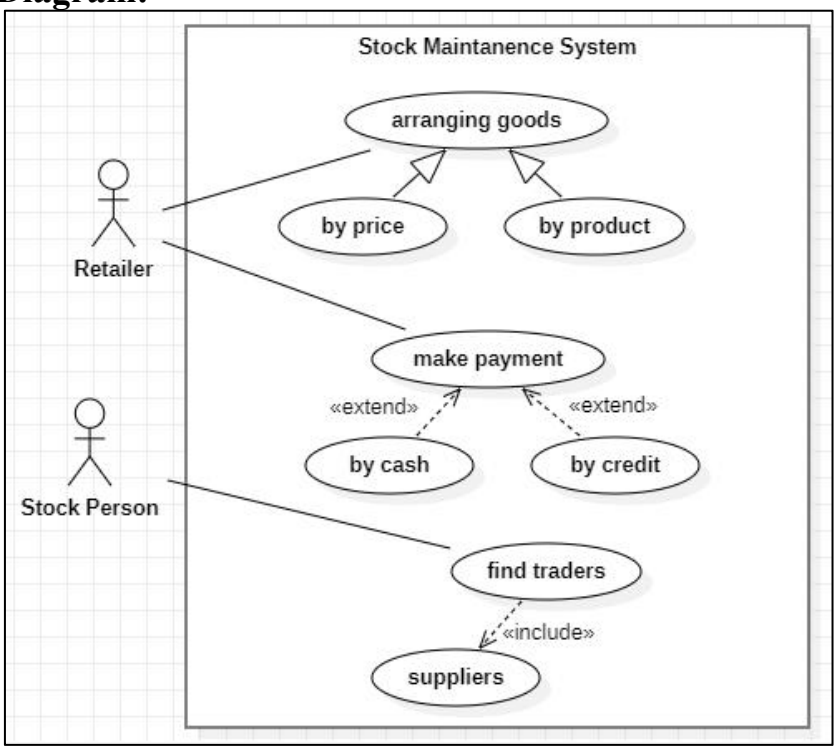
2.4.2 Class Diagram:



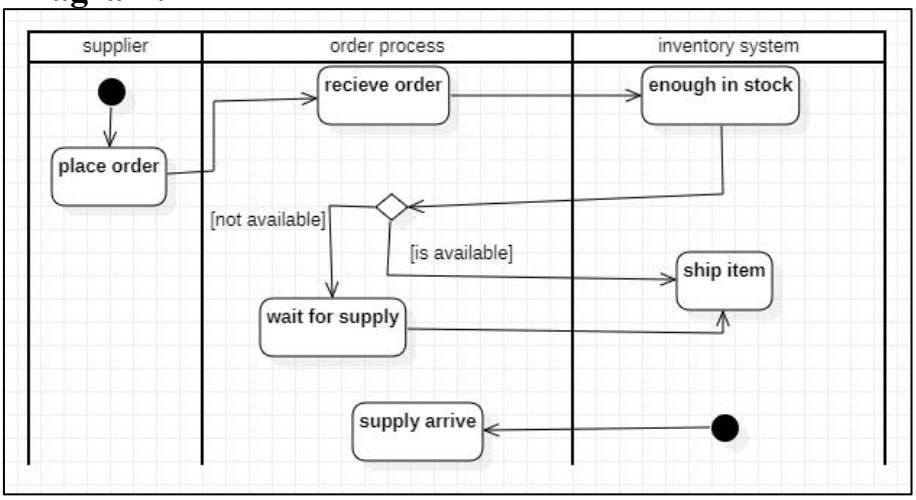
2.4.3 State Diagram:



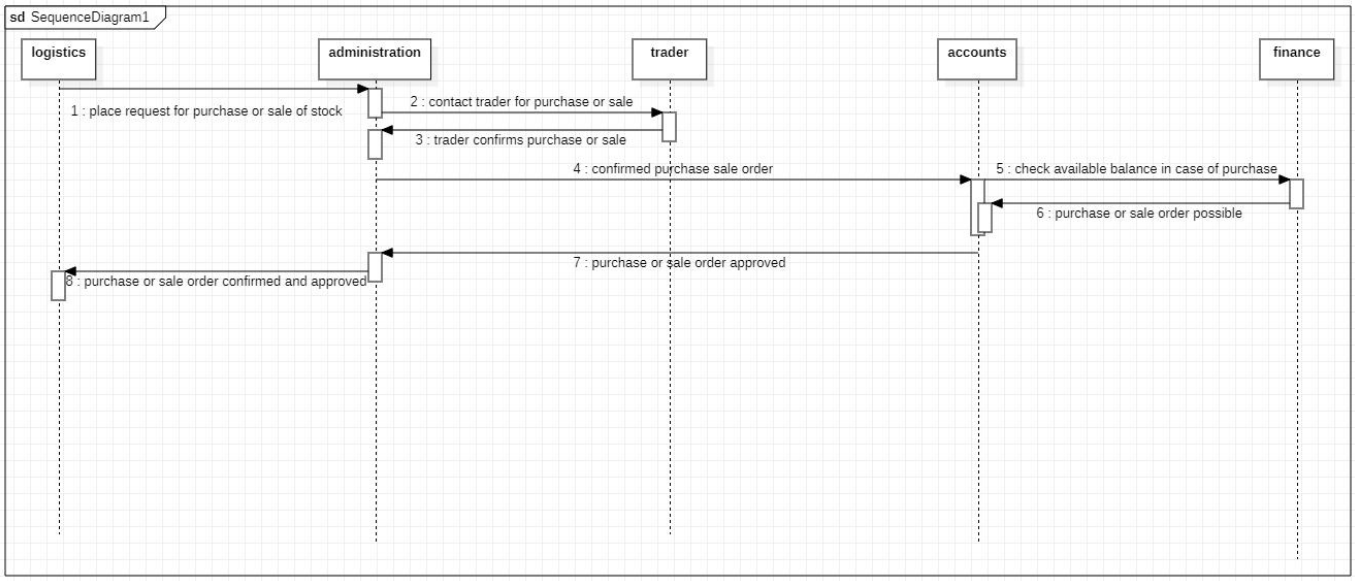
2.4.4 Use-Case Diagram:



2.4.5 Activity Diagram:



2.4.6 Sequence Diagram:



2.5 Passport Automation System:

2.5.1 Software Requirement Specification

1. Introduction:

1.1 Purpose of this Document:

The purpose of this document is to provide a detailed description of the Passport Automation System (PAS). It outlines the system's functionalities, performance requirements, design constraints, and overall objectives. This document aims to ensure that all stakeholders, including the development team, end-users, and regulatory bodies, have a clear understanding of the system's requirements and expectations.

1.2 Scope of this Document:

This document covers the entire Passport Automation System, detailing its objectives, functionalities, user interactions, and expected outcomes. It provides value to the customer by outlining development costs and timelines, facilitating better planning and resource allocation. The PAS aims to streamline passport application, processing, and issuance, enhancing user experience and operational efficiency.

1.3 Overview:

The Passport Automation System is designed to automate and manage various aspects of passport services such as application submission, verification, processing, and issuance. The system will provide an integrated solution to handle user data, document verification, status tracking, and notifications, aiming to improve the overall efficiency and transparency of the passport issuance process.

2. General Description:

2.1 Product Functions:

The PAS will provide the following functionalities:

- Application Submission: Online submission of passport applications with document uploads.
- User Management: Handling user registrations, profiles, and authentication.
- Verification Process: Automated and manual verification of submitted documents.
- Status Tracking: Real-time tracking of application status for users.
- Notification System: Automated notifications and alerts for application status updates.
- Reporting: Generation of reports for application statistics, processing times, and user activity.

2.2 User Characteristics:

The primary users of the PAS include applicants, passport office staff, and administrators. Each user group will interact with the system through secure, user-friendly interfaces tailored to their specific needs.

1. Applicants:

Individuals with basic to intermediate computer skills who are seeking to apply for or renew their passports. Use the system to submit applications, upload documents, track application status, and receive notifications.

2. Passport Office Staff:

Personnel with intermediate to advanced knowledge of passport processing and administrative tasks.
Interact with the system to process applications, verify documents, update application status, and generate reports.

3. Administrators:
Technical staff responsible for configuring system settings, maintaining security protocols, and ensuring system performance. Use the system to manage user access, configure system parameters, monitor performance, and ensure compliance with security standards.

2.3 Features and Benefits:
The system offers the following features and benefits:
- **Efficient Application Process:** Streamlined online application submission and document upload.
- **Enhanced User Experience:** User-friendly interface for tracking application status and receiving notifications.
- **Accurate Verification:** Automated verification processes to reduce errors and improve efficiency.
- **Comprehensive Reporting:** Detailed reports to help manage application processing and improve transparency.
- **Secure Data Handling:** Secure management of user data and application documents.

3. Functional Requirements:
3.1 Application Submission:
- **Online Form:** The system shall provide an online form for users to submit passport applications.
- **Document Upload:** The system shall allow users to upload necessary documents in various formats (PDF, JPEG, PNG).
- **Data Validation:** The system shall validate the entered data for completeness and correctness.

3.2 User Management:
- **Registration:** The system shall allow users to register and create accounts.
- **Authentication:** The system shall authenticate users during login.
- **Profile Management:** The system shall allow users to update their personal information and application details.

3.3 Verification Process:
- **Document Verification:** The system shall support both automated and manual verification of submitted documents.
- **Background Checks:** The system shall integrate with external databases for background checks and validation.

3.4 Status Tracking:
- **Real-time Updates:** The system shall provide real-time updates on the status of passport applications.
- **User Dashboard:** The system shall offer a user dashboard for applicants to view their application status and history.

3.5 Notification System:
- **Email Alerts:** The system shall send email notifications for status updates and required actions.
- **SMS Notifications:** The system shall support SMS notifications for critical updates.

3.6 Reporting:
- **Application Reports:** The system shall generate reports on application statistics and processing times.

- **User Activity Reports:** The system shall generate reports on user activity and interactions with the system.
- **Operational Reports:** The system shall provide reports for internal use by passport office staff to monitor workflow and performance.

4. Interface Requirements:

4.1 User Interface:

- **Applicant Interface:** The system shall provide a web-based interface for applicants to submit applications, track status, and receive notifications.
- **Staff Interface:** The system shall provide an interface for passport office staff to process applications and perform verifications.
- **Admin Interface:** The system shall provide an interface for administrators to configure system settings, manage user roles, and monitor performance.

4.2 Software Interfaces:

- **Database Integration:** The system shall integrate with a database to store and retrieve application and user data.
- **Authentication Services:** The system shall integrate with authentication services for secure user login and management.
- **API Integration:** The system shall provide APIs for third-party integration and data exchange with external systems.

5. Performance Requirements:

5.1 Speed and Efficiency:

- **Form Submission:** The system shall process application form submissions within 5 seconds.
- **Document Upload:** The system shall handle document uploads within 3 seconds per document.
- **Status Updates:** The system shall update application status within 2 seconds of a status change.

5.2 Reliability and Availability:

- **Uptime:** The system shall have an uptime of 99.9%.
- **Backups:** The system shall perform regular backups to prevent data loss and ensure data recovery.

5.3 Scalability:

- **Load Handling:** The system shall scale to handle increased user registrations and application submissions during peak times.
- **Future Growth:** The system shall be designed to accommodate future growth and additional features without performance degradation.

6. Design Constraints:

6.1 Technical Constraints:

- **Development Framework:** The system shall be developed using secure, scalable frameworks and technologies.
- **Security Protocols:** The system shall implement industry-standard security protocols for data transmission and storage.

6.2 Hardware and Software Limitations:

- **Compatibility:** The system shall be compatible with existing hardware and software systems used by passport offices.

- **Operating Systems:** The system shall operate on common operating systems (Windows, Linux, macOS).

7. Non-Functional Attributes:

7.1 Security:

- **Encryption:** The system shall use strong encryption for data at rest and in transit.
- **Access Control:** The system shall implement role-based access control to restrict access to sensitive information.

7.2 Portability:

- **Data Export:** The system shall support data export in common formats (CSV, XML) for integration with other systems.
- **Deployment:** The system shall be deployable on both on-premise and cloud environments.

7.3 Reliability:

- **Data Integrity:** The system shall perform regular data integrity checks to ensure consistency.
- **Error Handling:** The system shall provide robust error handling and recovery mechanisms.

7.4 Usability:

- **User-Friendly Interface:** The system shall have a user-friendly interface with clear navigation and comprehensive help documentation.
- **Training Resources:** The system shall provide training resources for users to facilitate smooth adoption.

7.5 Reusability:

- **Modular Design:** The system shall use modular code design to facilitate future enhancements and maintenance.

7.6 Compatibility:

- **Browser Compatibility:** The system shall be compatible with common web browsers (Chrome, Firefox, Safari).
- **Integration:** The system shall integrate with major government and third-party verification tools and software.

7.7 Data Integrity:

- **Accurate Data:** The system shall ensure accurate and consistent data storage and retrieval.
- **Unauthorized Access:** The system shall prevent unauthorized data access and modifications.

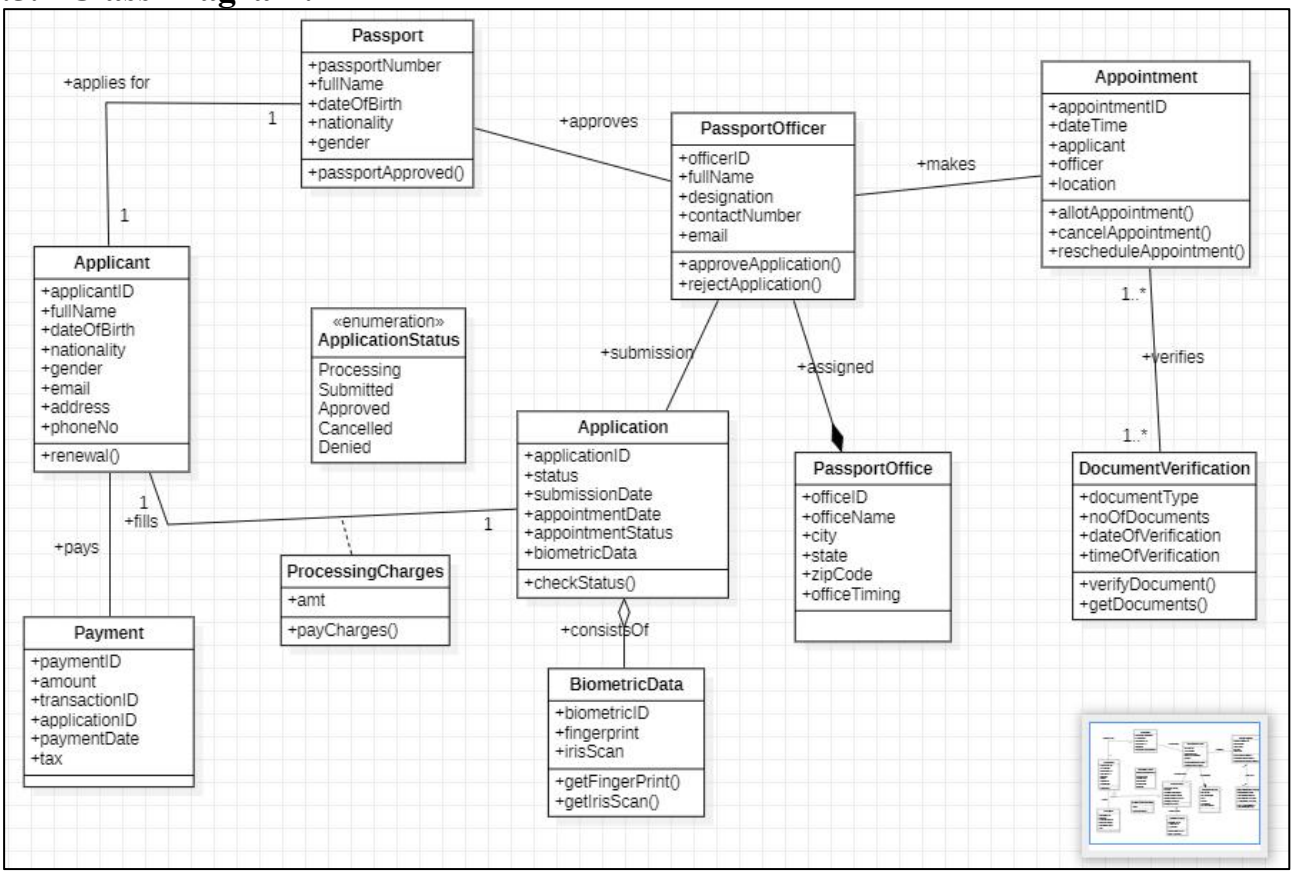
7.8 Scalability Capacity:

- **Feature Addition:** The system shall support the addition of new features and increased user load without performance issues.

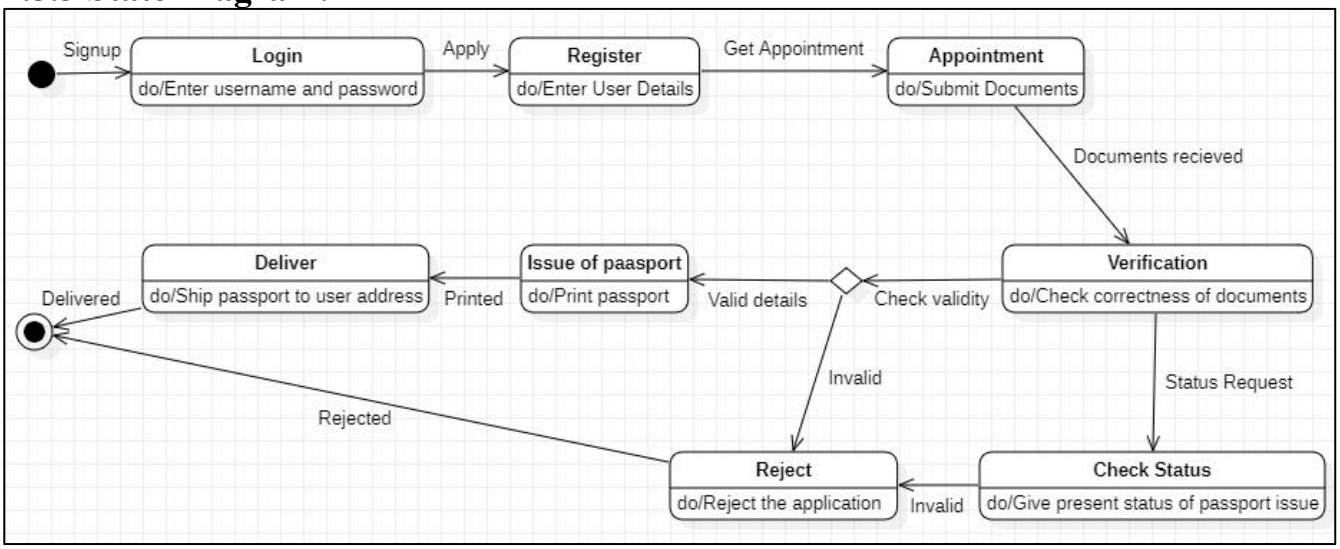
8. Preliminary Schedule and Budget:

The development of the HMS is estimated to take approximately 15 months with a budget of Rs. 20,00,000. This includes development, testing, deployment and maintenance cost.

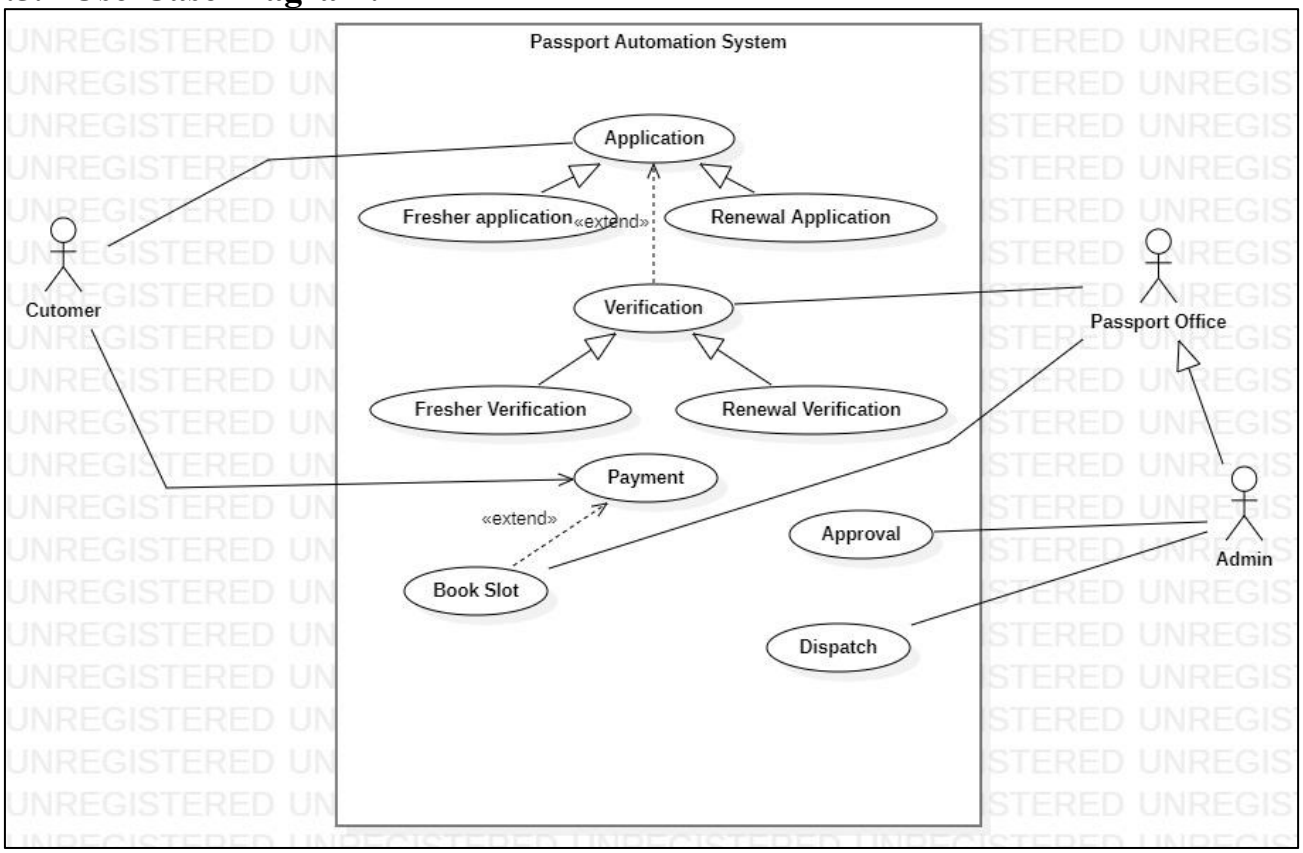
2.5.2 Class Diagram:



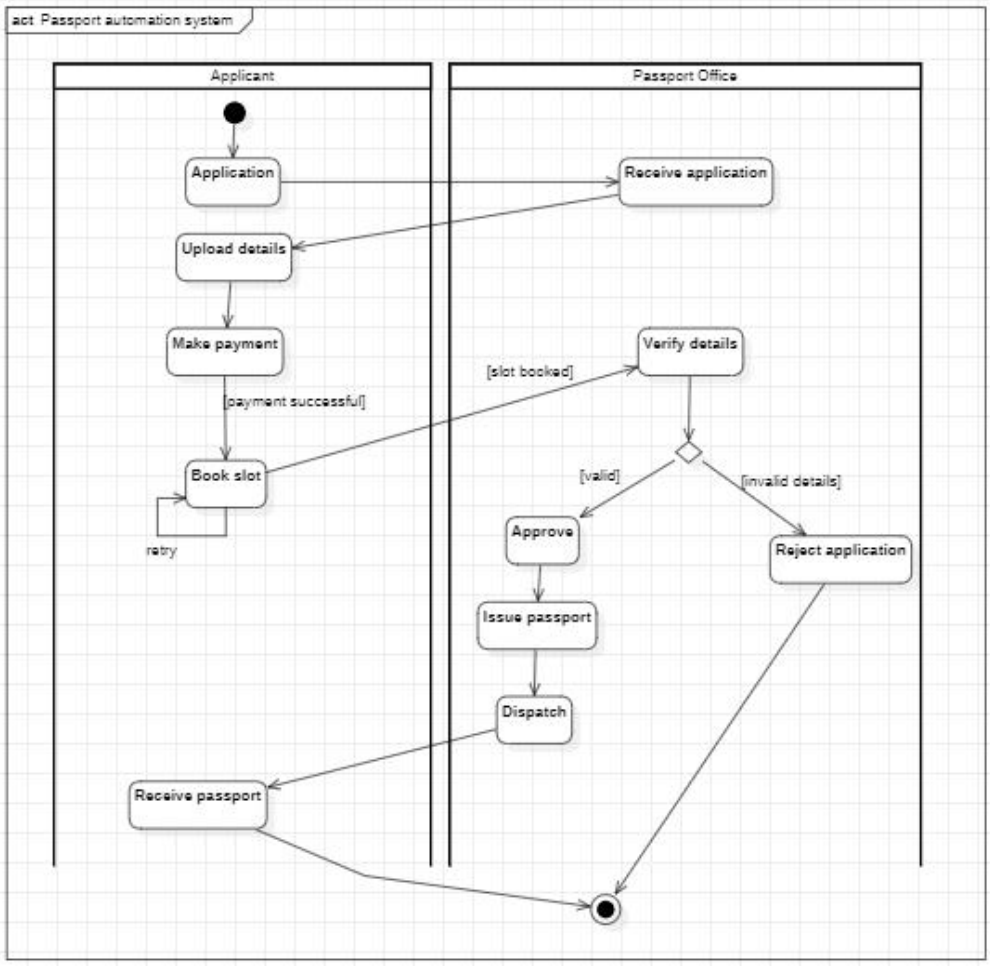
2.5.3 State Diagram:



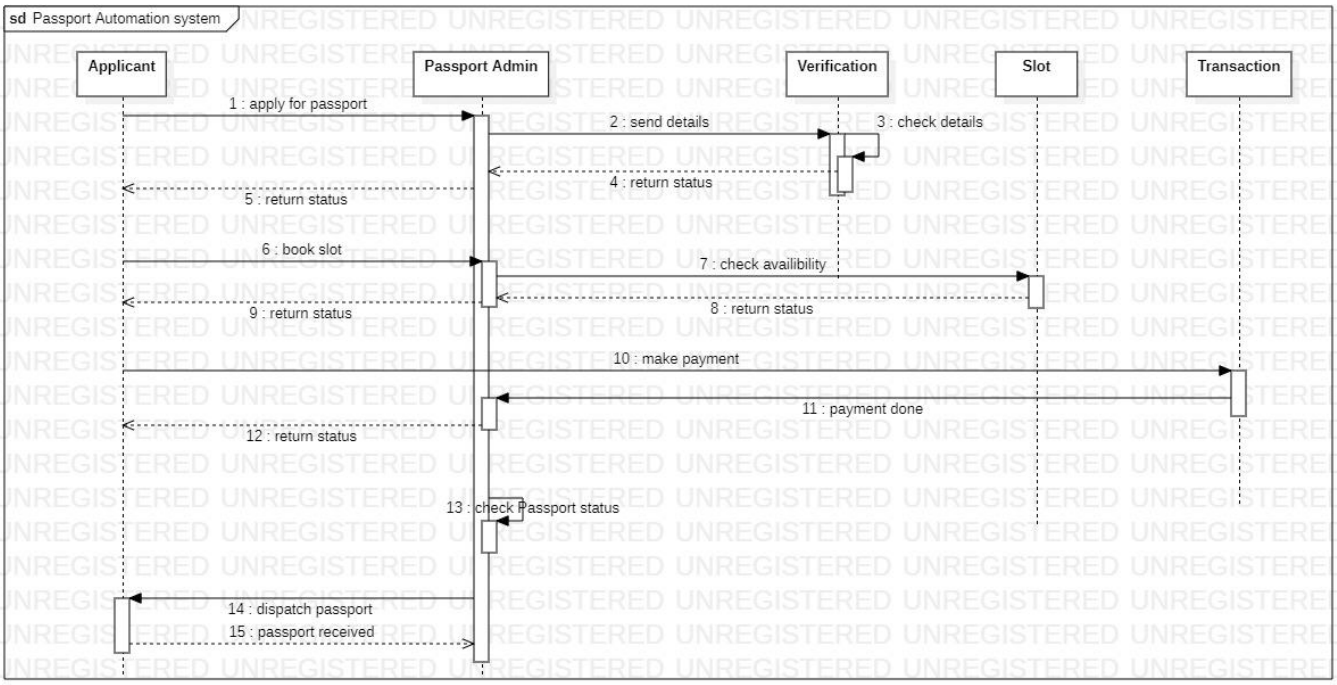
2.5.4 Use-Case Diagram:



2.5.5 Activity Diagram:



2.5.6 Sequence Diagram:



UNREGISTERED