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:

SNEHA PRASANNA
1BM22CS284

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 SNEHA PRASANNA(1BM22CS284) during the 5th Semester Oct24-Jan2025.

Signature of the Faculty Incharge:

NAME OF THE FACULTY: Prof. Prameetha Pai

Department of Computer Science and Engineering

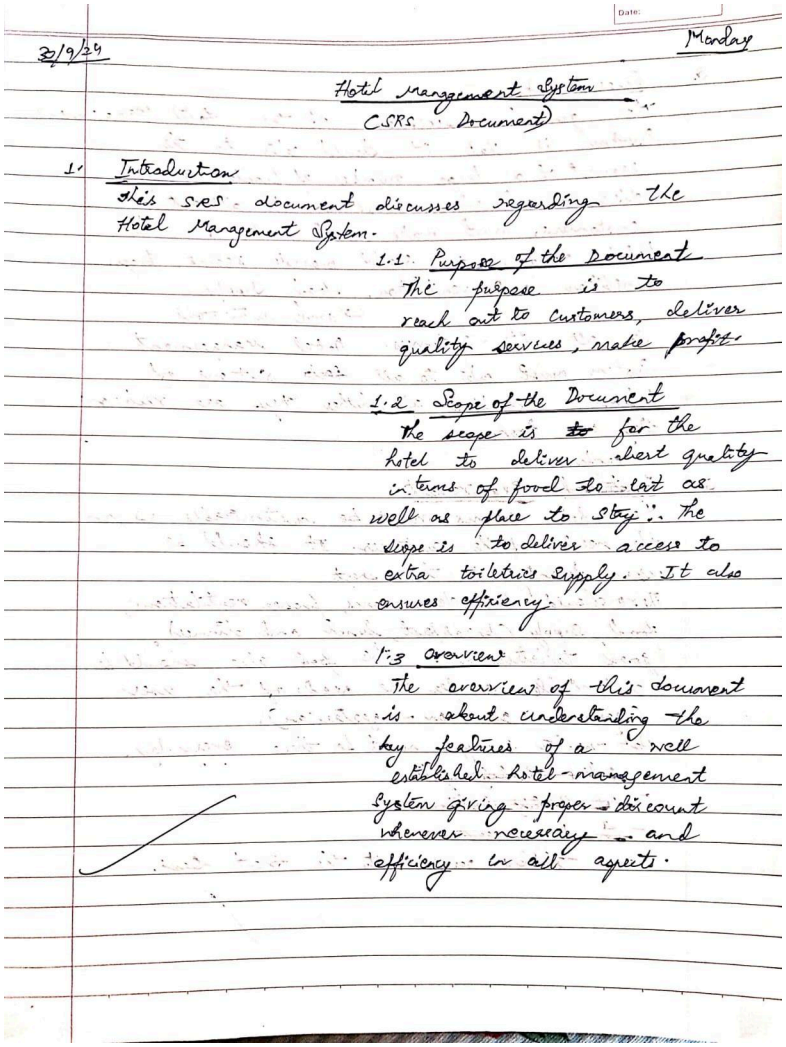
Table of Contents

1. HOTEL MANAGEMENT SYSTEM

1.1.Problem Statement

The hospitality industry is a fast-paced and dynamic environment where efficient management is crucial for smooth operations and customer satisfaction. A Hotel Management System is needed to streamline operations, enhance guest experiences, and reduce manual errors. The system should automate key processes such as room booking, guest check-in/check-out, payment processing, and inventory management, while providing robust analytics and reporting for business decision making.

1.2.SRS-Software Requirement Specification



23/9/24 Monday

Hotel management system
(SRS Document)

1. Introduction
This SRS document discusses regarding the Hotel Management System.

1.1 Purpose of the Document
The purpose is to reach out to customers, deliver quality services, make profit.

1.2 Scope of the Document
The scope is ~~to~~ for the hotel to deliver client quality in terms of food to eat as well as place to stay. The scope is to deliver access to extra toiletries supply. It also ensures efficiency.

1.3 overview
The overview of this document is about understanding the key features of a well established hotel management system giving proper discount whenever necessary and efficiency in all aspects.

2. General Description

The general description of the Hotel Management System is that it should cater to the request of a large number of people giving discount whenever necessary to attract customers, must make good profit. The hotel management should provide rooms, proper sanitation and ventilation, [food supply, good nutritions] to all its customers. Hotel management System must cater to all ~~sect~~ sections of people regardless of whether they are rich or they are poor.

Performance

3. ~~Functional~~ Requirements:

The hotel rooms must be customizable as per the needs of the customer. It should be user friendly in this aspect.

There should be clean rooms, proper ventilation, food supply (breakfast, lunch and dinner), good toiletries kit. The food also should be customizable as per the needs of the user (vegetarian or non-vegetarian).

Hot-water supply must be there everyday.

- ~~Online Booking system~~
- ~~App Booking System~~
- ~~In Room Service~~

Proper telephone service to the Front Desk.

4. Functional Requirements

The functional requirements are as follows:-

- → Online Booking System
- → In App Booking System
- → Walk-in Booking System
- → In Room Service
- → Room cleanliness.

5. Interface Requirements

The interface requirements for a hotel management system is website for bookings stay, no faults in the website, proper and efficient transactions both online and offline without any bugs or errors. App version of the website is also necessary.

6. Design Constraints

The design constraints for a hotel management system are it should be a fully functional website, faster transaction, no error in transaction, & should have app version of the website also. It should work regardless of whether there is internet or not.

7. Non-functional requirements

- → Security
- → Performance
- → Speed
- → Reliability
- → Scalability
- → Efficiency

8. Preliminary Schedule and Budget.

The project (total management) should take approximately 12–13 months to complete from the start date with a Budget of $8,000.

1.3.Class Diagram

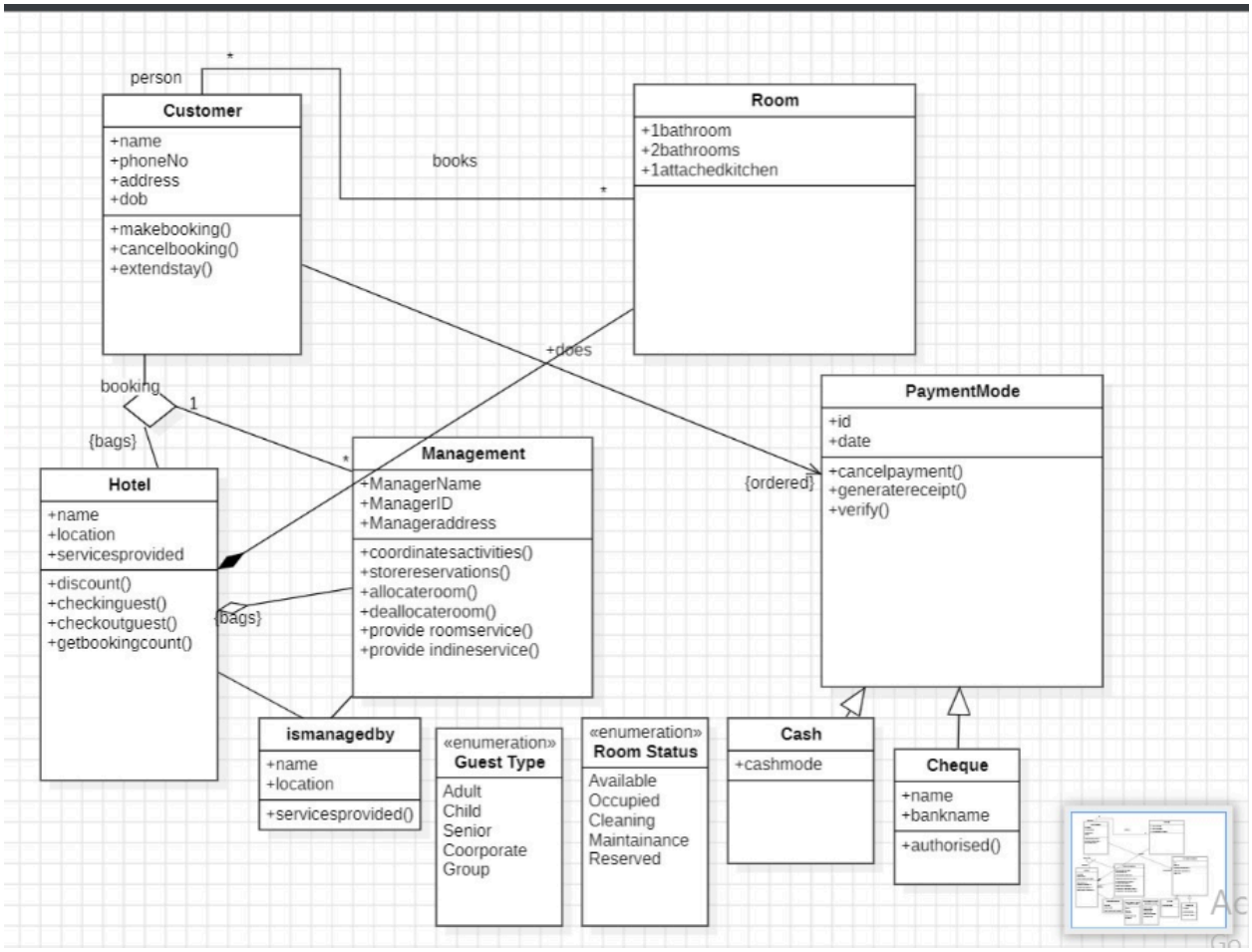


Fig 1.3.1

Brief Description

Customer:

- Attributes: name, phoneNo, address, dob.
- Methods: makeBooking(), cancelBooking(), extendStay().
- Relationship:
    - Has an association with Hotel (uses the books relationship).
    - Connected to PaymentMode through a dependency.

Hotel:

- Attributes: name, location, servicesProvided.
- Methods: discount(), checkInGuest(), checkOutGuest(), getBookingCount().

- Relationships:
    - Managed by Management (uses the isManagedBy relationship).
    - Has a dependency with Room.
    - Aggregates multiple Customer instances (via booking).

Room:

- Attributes: 1Bathroom, 2Bathrooms, 1AttachedKitchen.
- Enumeration: RoomStatus (states include Available, Occupied, Cleaning, Maintenance, Reserved).
- Relationships:
    - Connected to Hotel and used in room allocation.

Management:

- Attributes: ManagerName, ManagerID, ManagerAddress.
- Methods: coordinateActivities(), storeReservations(), allocateRoom(), deallocateRoom(), provideRoomService(), provideIn-DineService().
- Relationships:
    - Manages Hotel via isManagedBy.
    - Coordinates bookings and room allocations.

PaymentMode:

- Attributes: id, date.
- Methods: cancelPayment(), generateReceipt(), verify().
- Subclasses:
    - Cash: Includes cashMode.
    - Cheque: Includes name, bankName, authorized().

Enumerations:

- GuestType: Includes values such as Adult, Child, Senior, Corporate, Group.
- RoomStatus: Defines the states of a room.

1.4.State Diagram

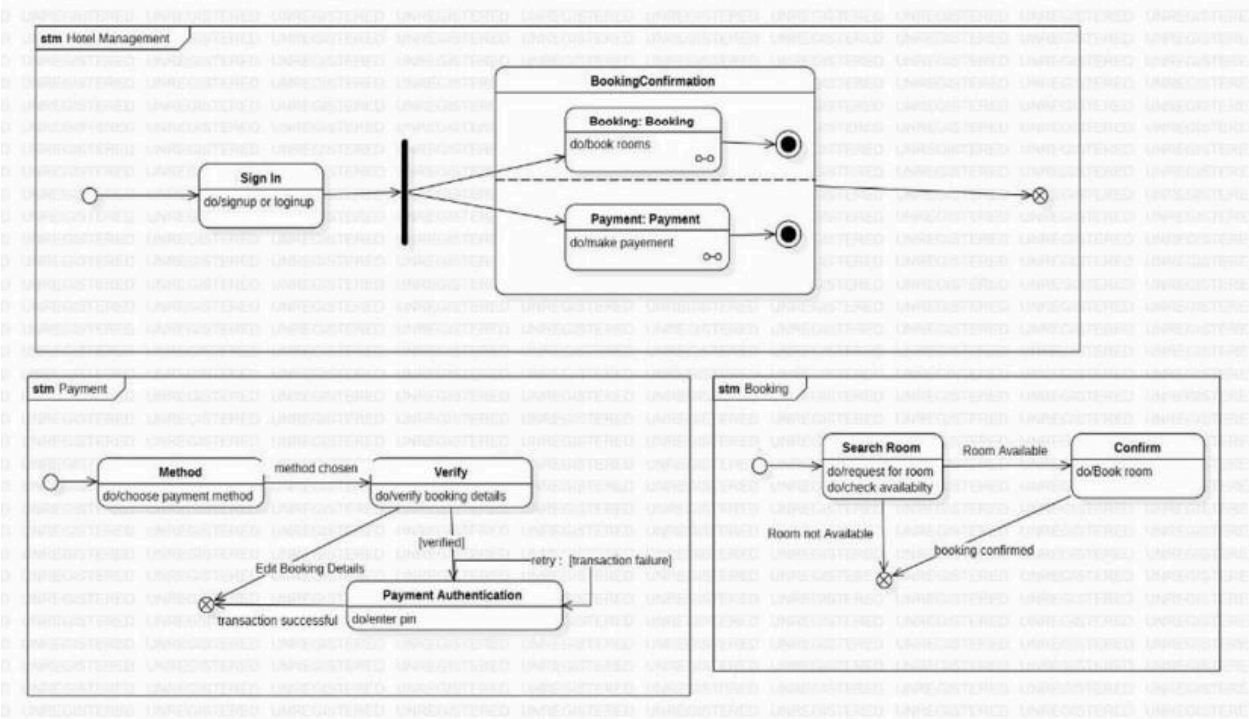


Fig 1.4.1

Brief Explanation

The state diagram for the Hotel Management System (HMS) represents the lifecycle of two key processes: Booking and Payment, detailing the stages and transitions for each.

- BookingProcess: The booking process begins with the Booking Initiated state when a guest starts reserving a room. It then transitions to the Room Selected state once a room is chosen. From here, the process moves to Booking Confirmed if the booking is finalized or to Booking Canceled if the guest or staff cancels it.

- PaymentProcess: Payment starts in the Payment Pending state when the booking is confirmed. It transitions to Payment In Process when the payment begins. Depending on the outcome, it moves to Payment Successful if the transaction is completed or to Payment Failed in case of errors like insufficient funds or technical issues, allowing retries if needed.

1.5.Use Case Diagram

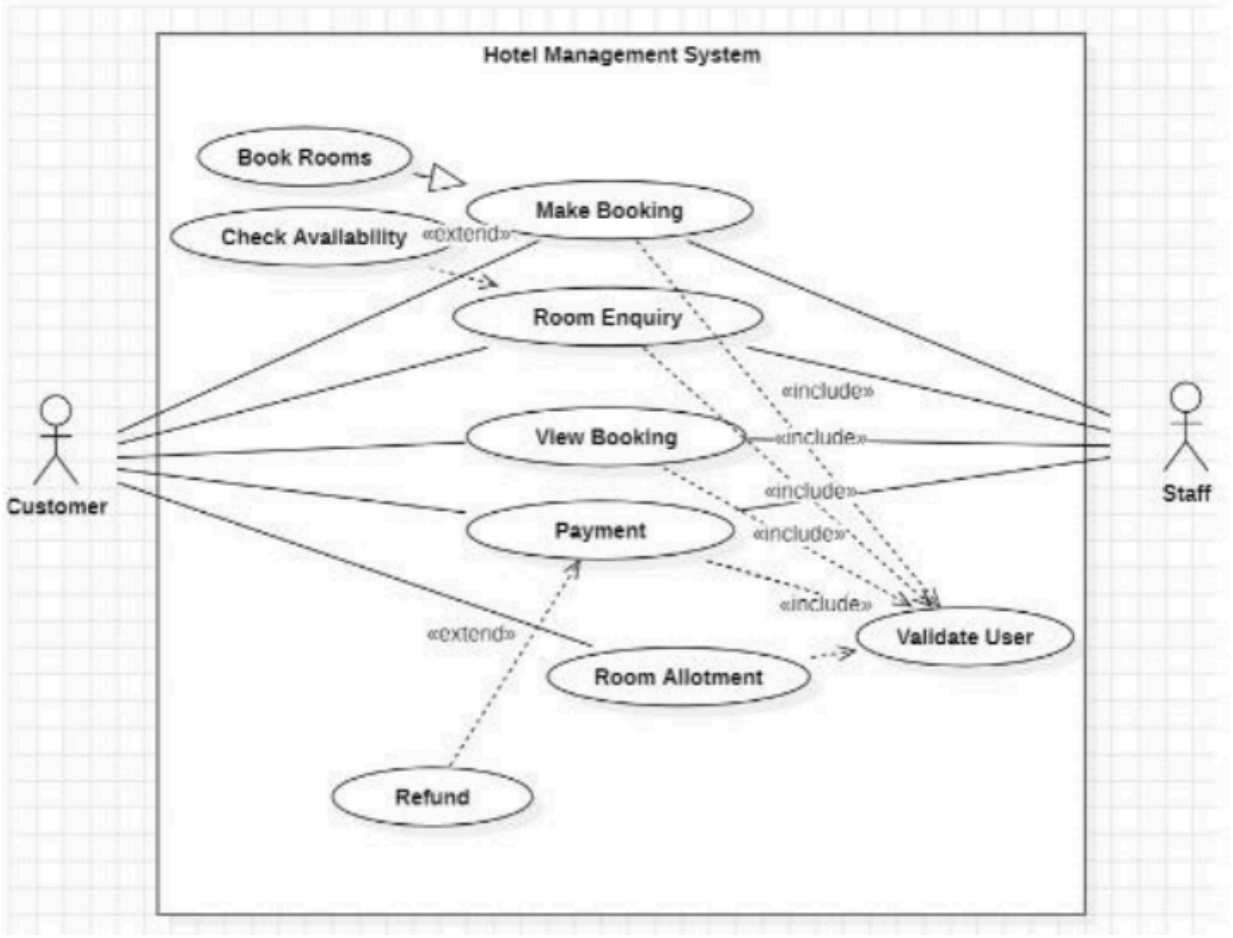


Fig 1.5.1

Brief Description

The use case diagram illustrates the interactions between users and the system by identifying the primary use cases and actors.

● Actors:

○ Guests (Customer), Staff (Receptionist, Manager, Housekeeping).

● Use Cases:

○ Book Room, Check Room Availability, Make Payment, Generate Reports, Manage Staff, Update Inventory.

- Relationships:
    - Guests interact with use cases like Book Room and Make Payment.
    - Staff interacts with Check Room Availability and Manage Inventory.
    - Administrator handles Generate Reports and Manage Staff

1.6.Sequence Diagram

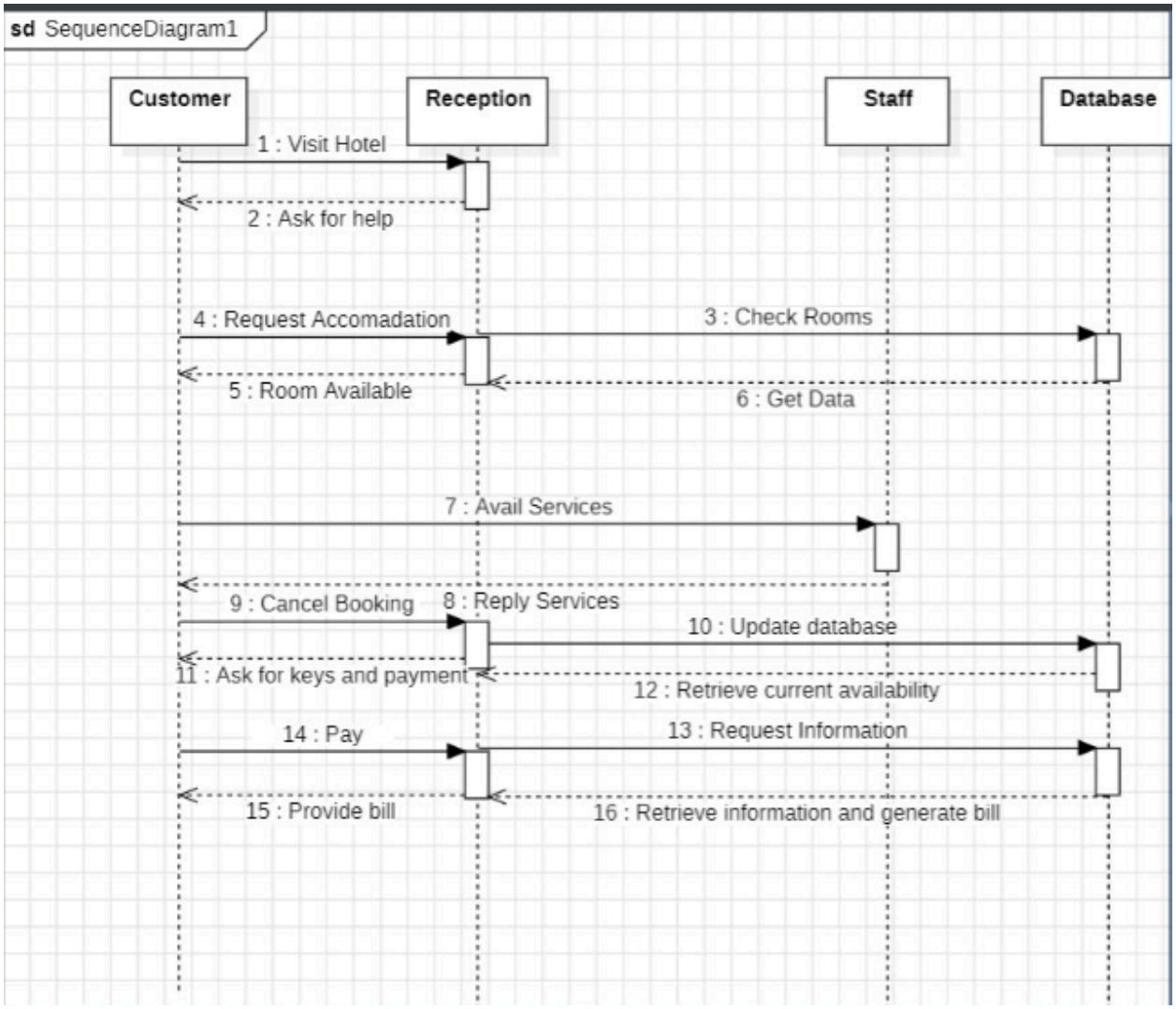


Fig 1.6.1

Brief Description

1. Customer Interaction:

- The customer visits the hotel, requests help, and asks for accommodation.

2. Reception's Role:

- The reception checks room availability with the staff and provides updates to the customer.

3. Staff and Database:

- The staff retrieves room data and availability from the database and communicates it back to the reception.

4. Service and Payment:

- The customer avails services, cancels bookings if needed, requests keys, and makes payment.

5. Bill Generation:

- The database retrieves required information, and the reception provides the bill to the customer.

1.7.Activity Diagram

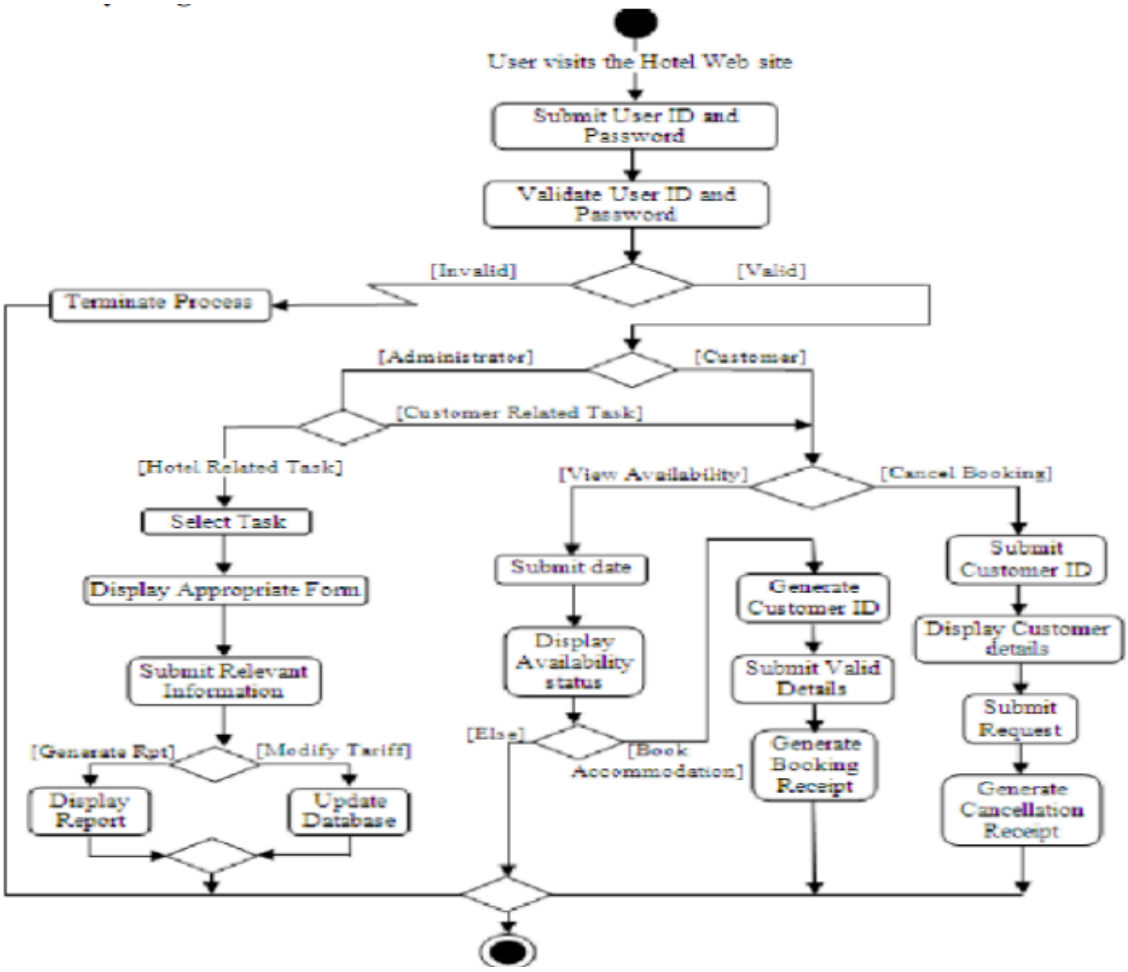


Fig.1.7.1

Brief description

The activity diagram focuses on workflows and processes within the HMS, showing how tasks are performed and decision points in the workflow.

Example: Guest Check-In Process

Activities:

- Guests arrive at reception.
- Receptionist retrieves booking details.
- If a booking exists, proceed to assign the room.
- If no booking exists, create a new booking.
- Process payment.
- Hand over room keys and update system status to "Occupied".

Decision Points:

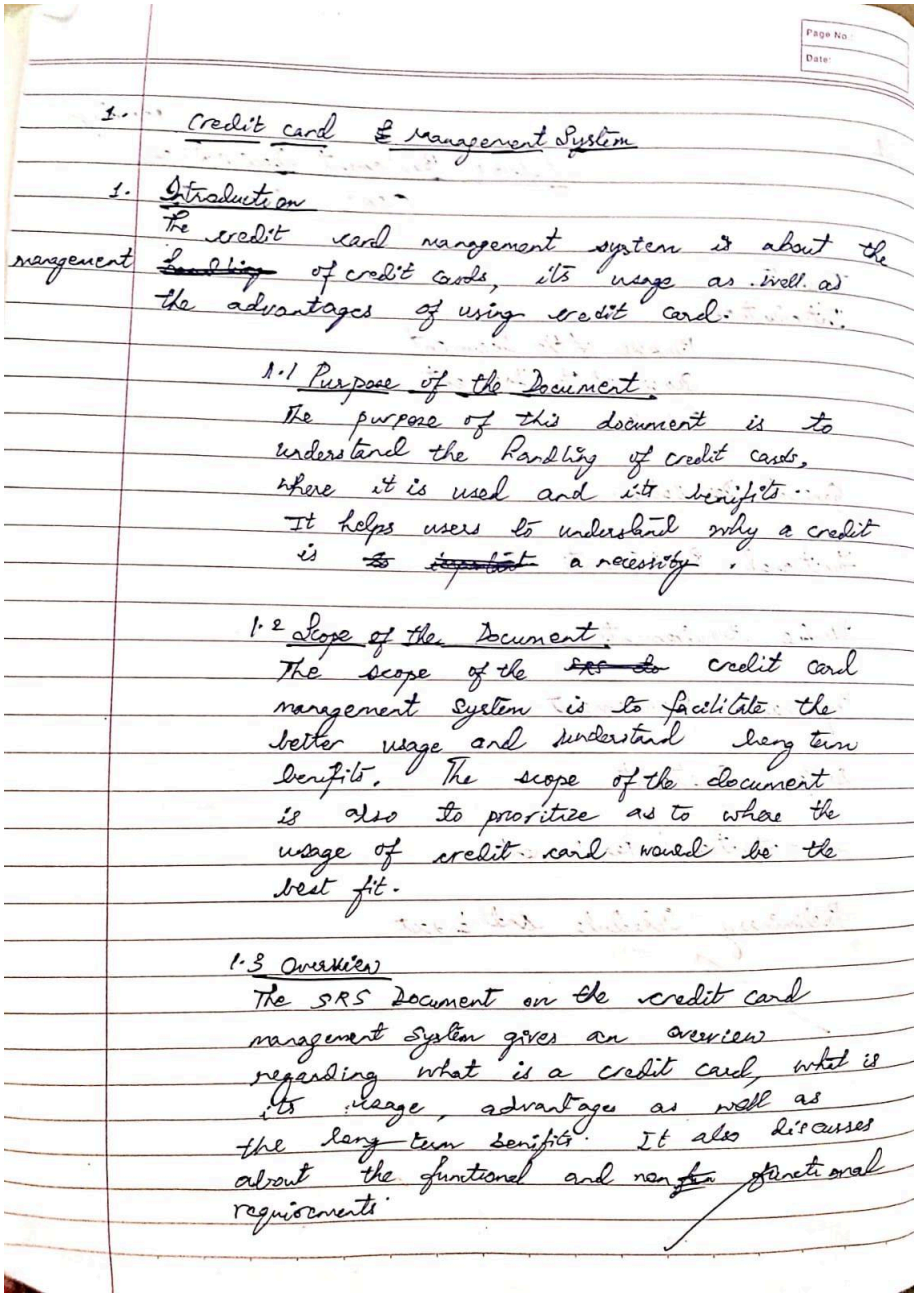
- Check if the booking exists.
- Verify payment status before assigning the room.

This diagram provides a clear understanding of the workflow for check-ins.

2.CREDIT CARD PROCESSING SYSTEM

2.1.Problem Statement

The Credit Card Management System (CCMS) is designed to simplify and automate credit card-related processes. It aims to manage user accounts, process applications, track transactions, generate statements, calculate rewards, ensure payment compliance, and prevent fraudulent activities. The system provides a secure, user-friendly interface for customers and staff, ensuring efficient management of credit card operations.



1. Credit card ~~&~~ management System

1. Introduction

The credit card management system is about the management ~~handling~~ of credit cards, its usage as well as the advantages of using credit card.

1.1 Purpose of the Document

The purpose of this document is to understand the handling of credit cards, where it is used and its benifits. It helps users to understand why a credit is ~~to important~~ a necessity.

1.2 Scope of the Document

The scope of the ~~SRS to~~ credit card management system is to facilitate the better usage and understand long term benifits. The scope of the document is also to prioritize as to where the usage of credit card would be the best fit.

1.3 Overview

The SRS Document on the credit card management system gives an overview regarding what is a credit card, what is its usage, advantages as well as the long term benifits. It also discusses about the funtional and non ~~fun~~ functional requirements.

Page No.:
Date:

2. General Description

The general description of the SRS Document explains about the meaning of credit card, usage, benifits and much more. It gives a broad perspective on the ~~case~~ ~~of~~ credit cards.

3. Functional Requirements.

The credit card should help to do transactions at a much faster rate.

Credit cards should be registered to the name of the user. The credit card user must keep a pin for the credit card and only on entering the ~~card~~ pin of the credit card, transactions can take place ↳ further

The credit card must be having some balance amount to do transactions otherwise the transactions are not possible. There could also be display of pin, credit card details.

4. Interface Requirements

The interface requirements for a credit card management system would be medium through which the credit card transactions would take place. It could be a proper internet connection to enable faster transactions. It could also be computer or some machinery in case the credit card that is in use requires that.

5. Performance Requirements

The performance requirements of a credit-card ~~case~~ management system would be the speed at which the transaction can take place in all possible situations. It also can be measured on how long ~~the~~ it would

take to complete the transaction. The credit card system should be efficient and it should be errorless transactions.
The credit card management system should be secure and must require its authentication before the usage.

6. Design Constraints.
The design constraints for the credit card management system are there be swipe or tap feature for the credit card. ~~There must be~~ The design constraints are on top functionality, no-internet service, contact-less transactions.

7. Non Functional Requirements
The non functional requirements are performance, speed, reliability, faster, efficient transactions, no error transactions, easy to use, convinience.

8. Preliminary Schedule and Budget.
The project should approximately take 1 year or 1.5 years to complete starting from initial date. The budget should be around $150.

20/9

2.3.Class Diagram

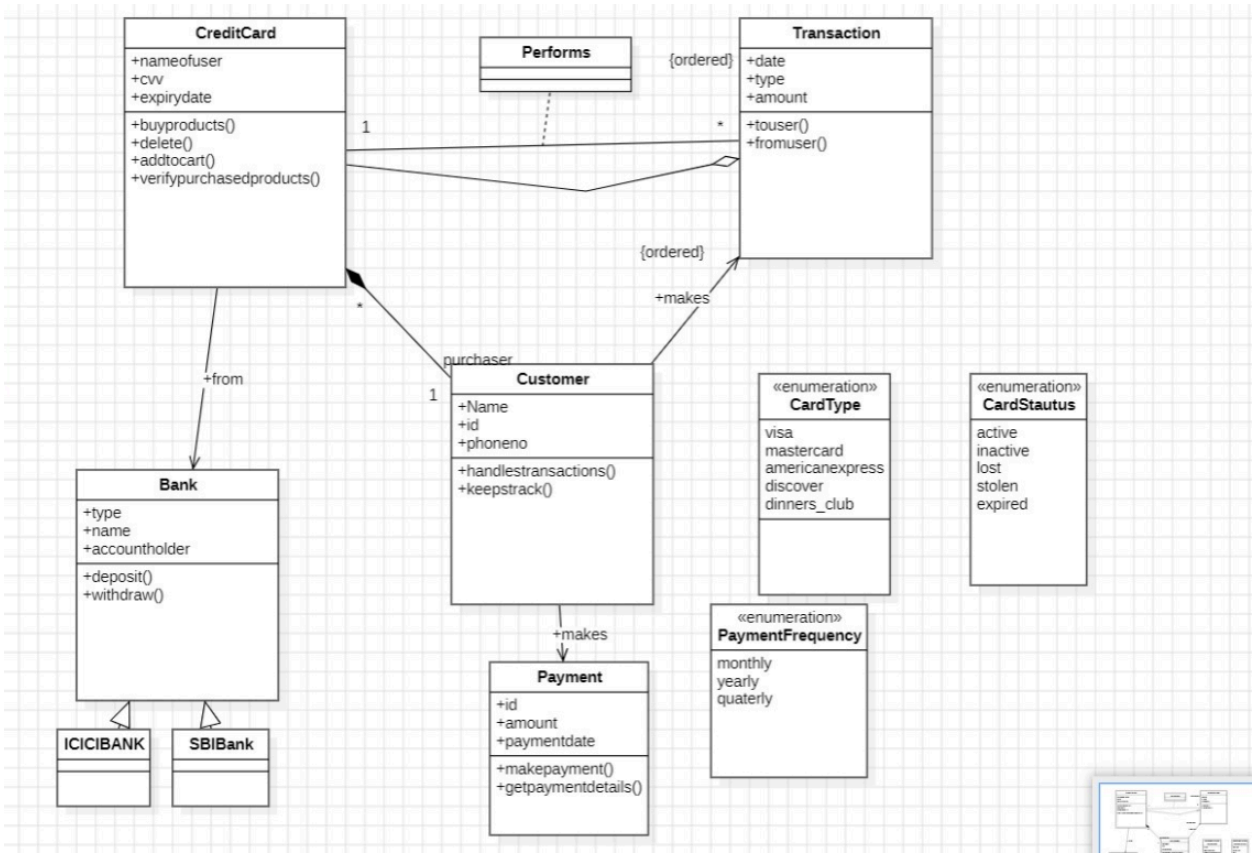


Fig.2.3.1

Brief Description:

- CreditCard: Used by customers to make purchases, linked to transactions and issued by banks.
- Customer: Handles transactions and payments using credit cards.
- Transaction: Tracks payment details like date, type, and amount.
- Bank: Issues credit cards and handles deposits/withdrawals (e.g., ICICIBank, SBIBank).
- Payment: Manages payment processing and details.
- Enumerations:
    - CardType: Types of cards (e.g., Visa, Mastercard).
    - CardStatus: Status of cards (e.g., Active, Lost).
    - PaymentFrequency: Payment intervals (e.g., Monthly, Yearly).

This system maps customer, credit card, and payment interactions efficiently.

2.4.State Diagram

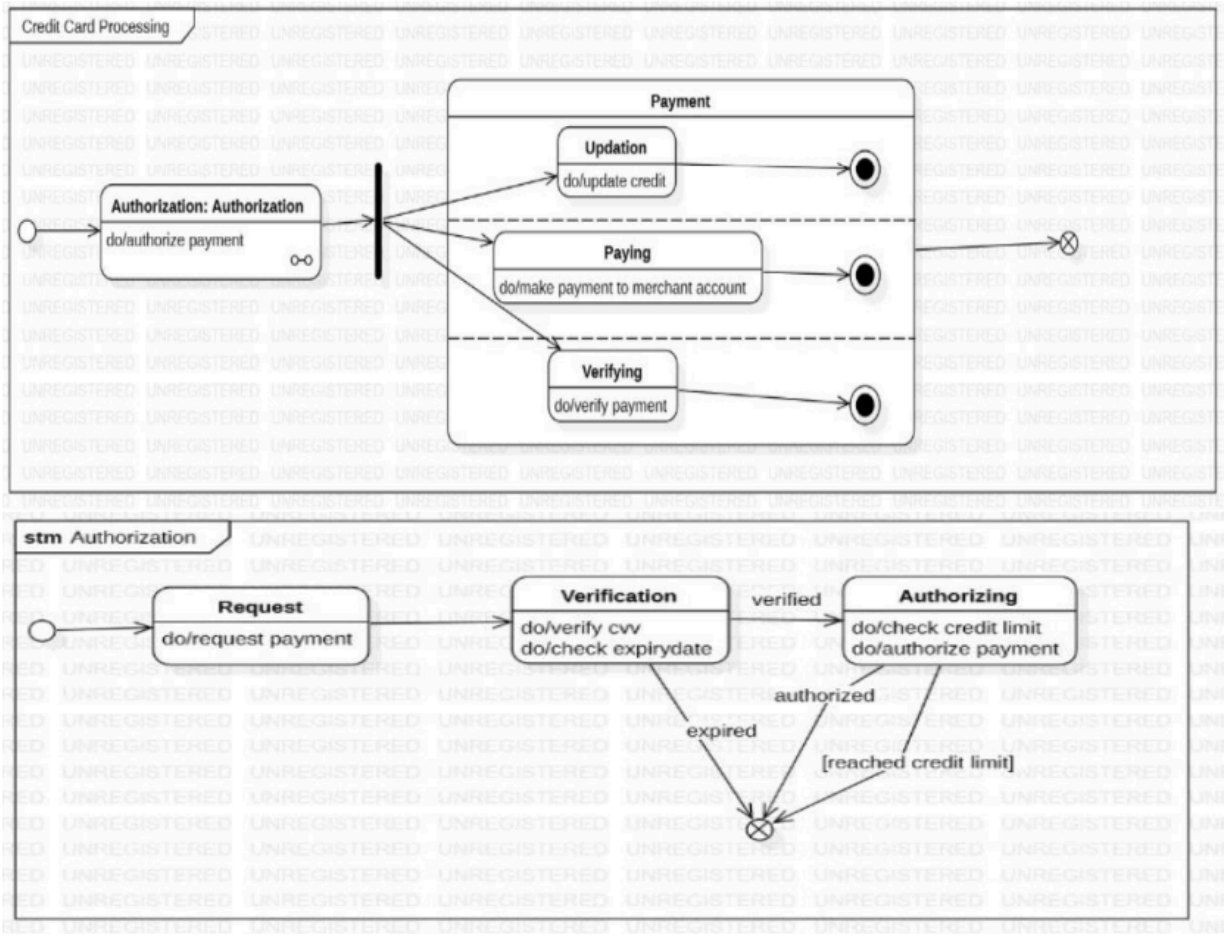


Fig.2.4.1

Brief Description

The state diagram illustrates the lifecycle of credit cards and payments:

1. Credit Card States:
    - Application Submitted → Under Review → Approved → Activated → Blocked (if flagged for fraud).
2. Payment States:
    - Payment Pending → Processing → Successful or Failed.

- Failed payments can retry or result in late fee calculations.

2.5.Use Case Diagram

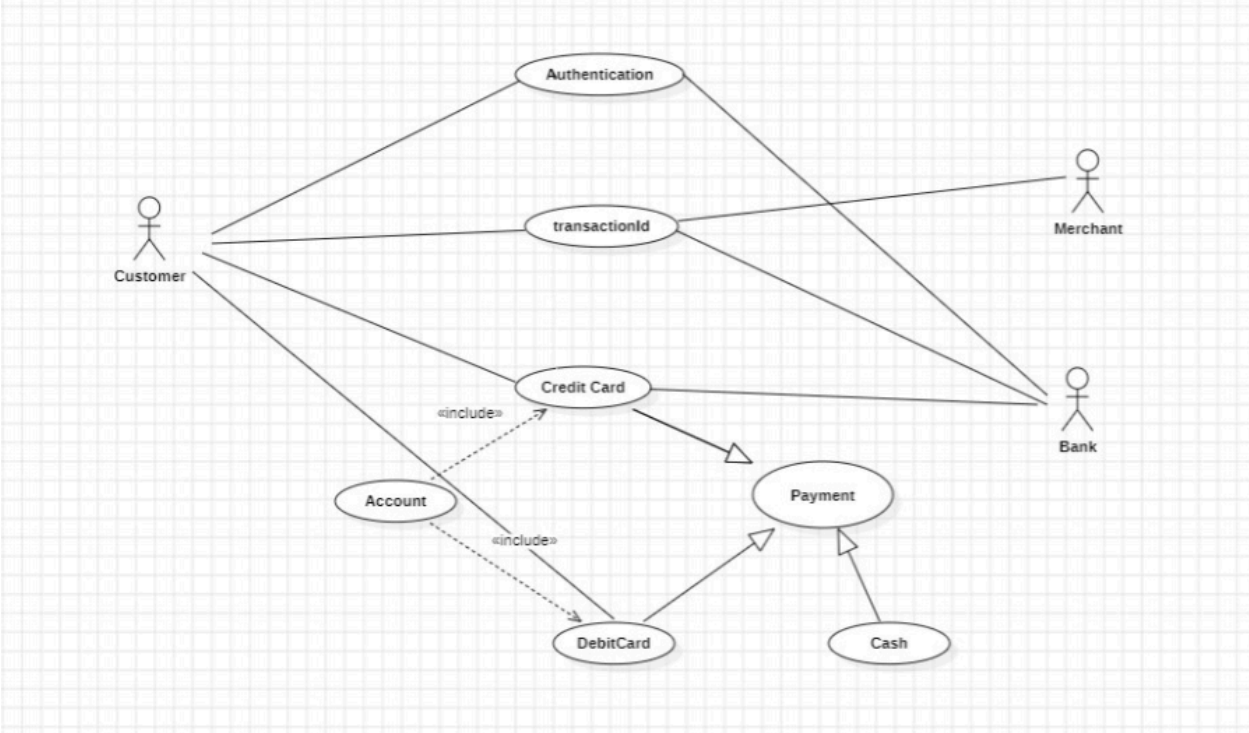


Fig.2.5.1

Brief Description

Actors:

The diagram contains five actors: Customer, Merchant, Bank, Receptionist, and Admin.

Use Cases:

The diagram contains five use cases: Authentication, transactionId, Credit Card, Account, and Payment.

Relationships:

- The Customer actor is associated with Authentication, Credit Card, and Account use cases.

- The Merchant actor is associated with Authentication and transactionId use cases.
- The Bank actor is associated with Authentication and Payment use cases.
- The Receptionist and Admin actors are associated with the Payment use case.
- The Credit Card use case includes the Account use case.
- The DebitCard and Cash use cases generalize the Payment use case.

Overall, the diagram represents the use cases involved in managing payments and transactions within a hotel management system.

2.6.Sequence Diagram

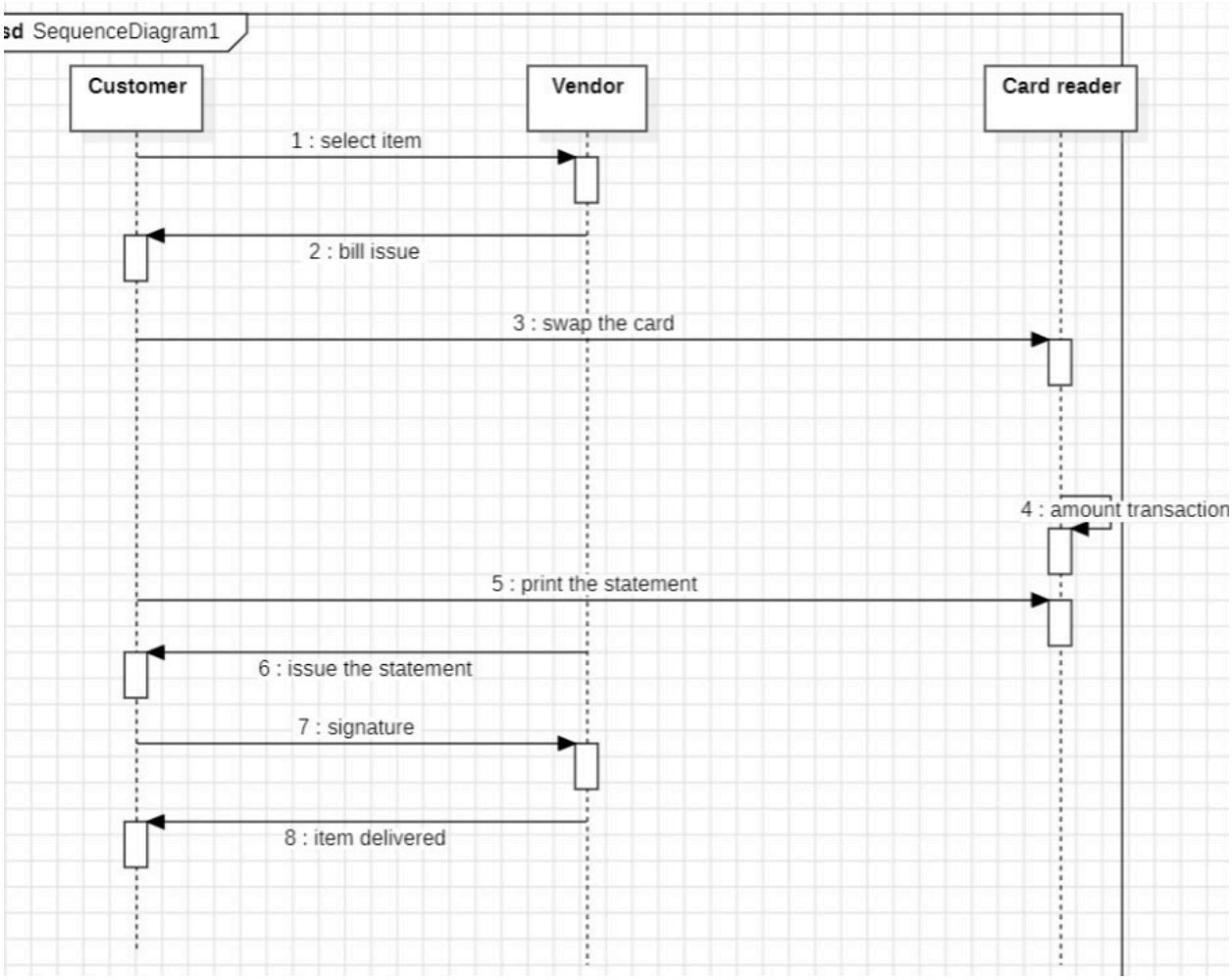


Fig 2.6.1

Brief Description

- Customer: The customer selects an item and requests a bill.
- Vendor: The vendor issues a bill to the customer.
- Customer: The customer hands their credit card to the vendor.

- Vendor: The vendor uses the card reader to swap the card.
- Card Reader: The card reader performs the transaction.
- Card Reader: The card reader prints the statement for the customer.
- Vendor: The vendor issues the statement to the customer.
- Customer: The customer signs the statement.
- Vendor: The vendor delivers the purchased item to the customer.

This sequence accurately captures the steps involved in the interaction.

2.7.Activity Diagram

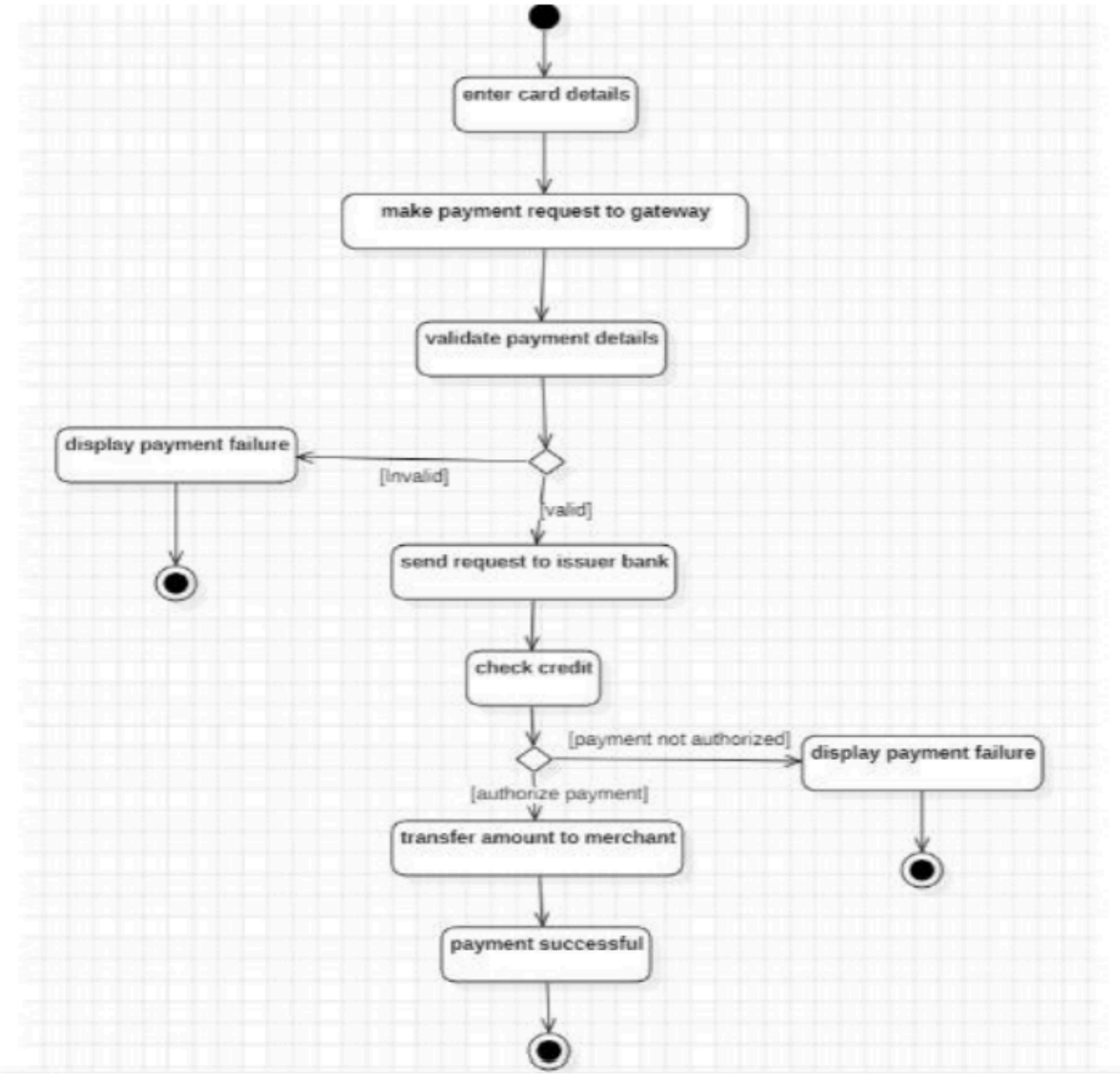


Fig.2.7.1

Brief Description

Example:Fraud Detection Process:

1. Start monitoring transactions.
2. Detect suspicious patterns (e.g., high-value transactions or unusual locations).
3. Trigger fraud alert.
4. Admin investigates and blocks the card if necessary.
5. Notify the customer and request confirmation.

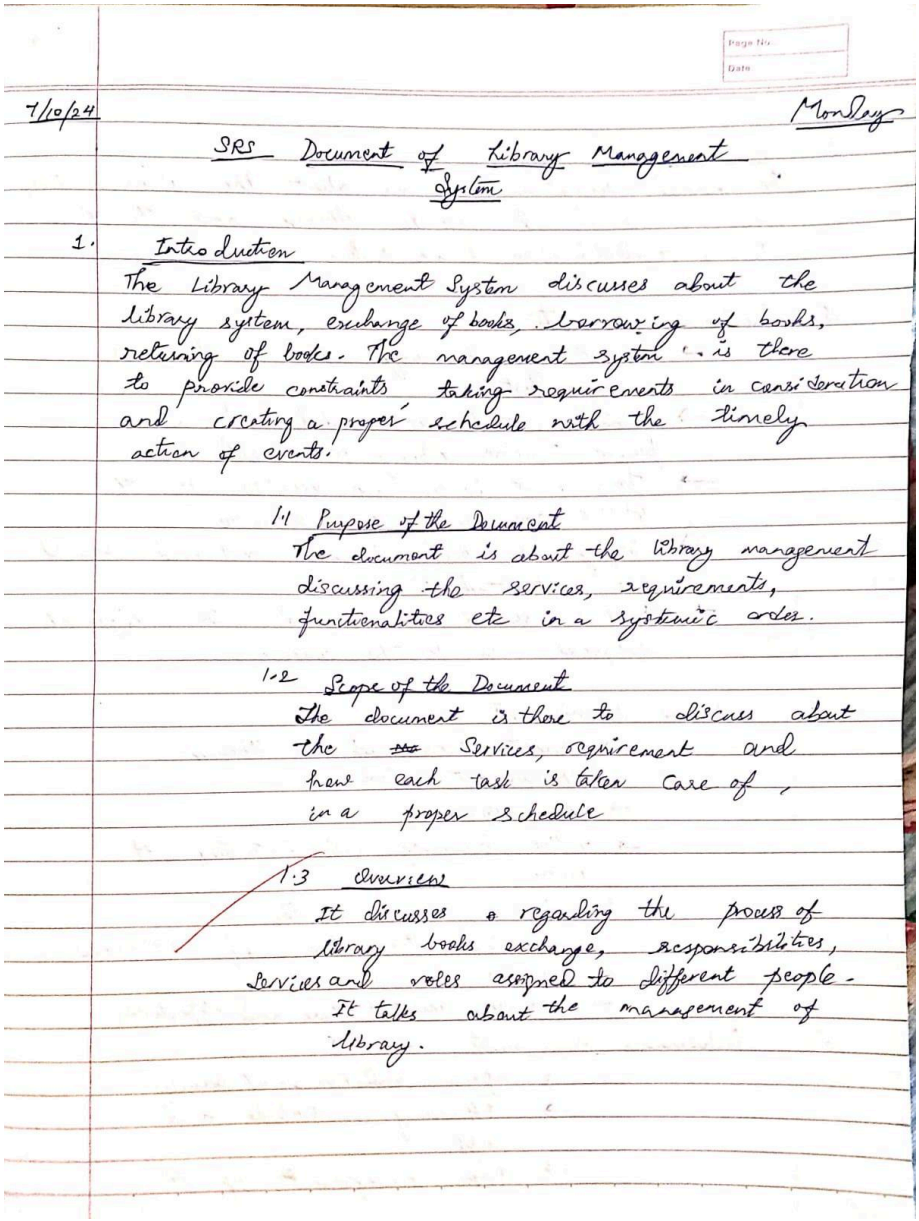
This structure clearly outlines the flow of activities in the fraud detection process.

3.LIBRARY MANAGEMENT SYSTEM

3.1.Problem Statement

The Library Management System (LMS) is designed to streamline the management of library resources and operations. It enables users to search for books, issue and return them, track borrowing history, and manage fines, while also providing administrators with tools for cataloging, inventory management, and user account handling.

3.2.SRS-Software Requirements Specification



7/10/24 Monday

SRS Document of Library Management System

1. Introduction

The Library Management System discusses about the library system, exchange of books, borrowing of books, returning of books. The management system is there to provide constraints taking requirements in consideration and creating a proper schedule with the timely action of events.

1.1 Purpose of the Document

The document is about the library management discussing the services, requirements, functionalities etc in a systematic order.

1.2 Scope of the Document

The document is there to discuss about the ~~xxx~~ Services, requirement and how each task is taken care of, in a proper schedule

1.3 Overview

It discusses a regarding the process of library books exchange, responsibilities, services and roles assigned to different people. It talks about the management of library.

Page No.:
Date:

2. General Description

The general description discusses about the representatives, roles, services provided by the library and the timely scheduled action of each task.

3. Functional Requirements

→ Speed of ~~topics~~ borrowing, exchange of books should be fast
→ Login credentials are needed for a person (online library management)
→ There must be an app version of the website and an online website.
→ Display of books borrowed, returned should be kept track of.
→ Books should be organised in different categories as per the genre.

4. Interface Requirements

→ Website version of the library management System
→ App version
→ Online borrowing and returning of books.
→ Efficiency, speed on the website for any transaction (if returned late).
→ RDBMS must be there and effective

5. Performance Requirements

→ speed goodDelivery of services
→ Efficiency in website and app.
→ Proper management of the

different services, roles and responsibilities.
→ No bugs/errors on website

6. Design Constraints
→ fully functional website
→ High speed, proper delivery of services
→ Conversion of website should be there.
→ Should work regardless of whether there is an internet connection or no.
→ Proper efficiency.

7. Non-Functional Requirements.
→ Security : It should not be hacked by other users.
→ Performance : The website must be opened quickly and
→ Speed : High speed when browsing show all the necessary things
→ Reliability : when used by multiple users, it should respond well and if error or returning a book. occurs it should
→ Scalability : It should reach large users. be able to
→ Efficiency : It must be efficient in all the operations required. recover itself

8. Preliminary Schedule and Budget :
The project (library management system) should take approximately 1-2 years to complete from the start date with a budget of $ 4000.

3.3.Class Diagram

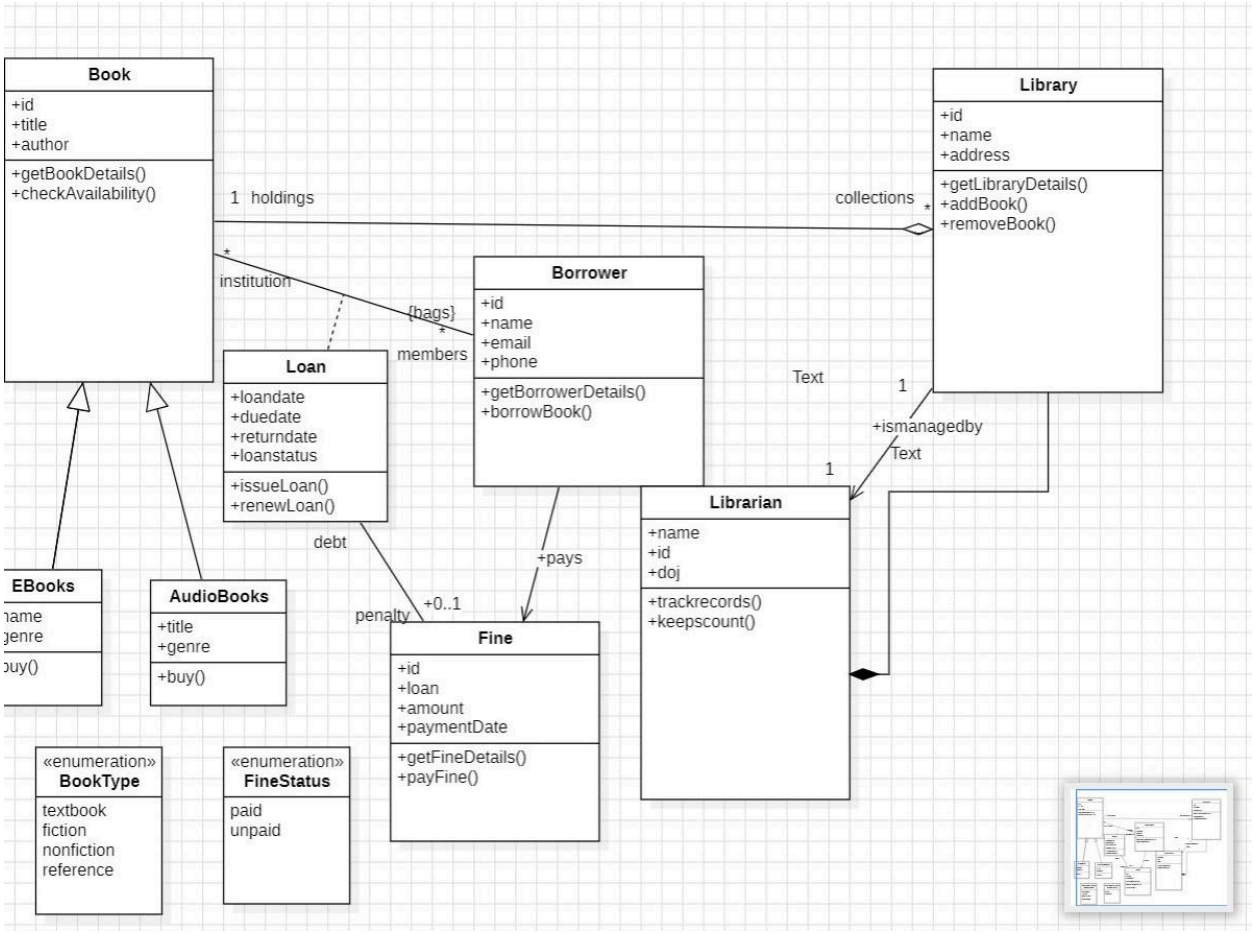


Fig.3.3.1

Brief Description

Entities:

- Book: Represents a book.
- EBooks: Represents electronic books.
- AudioBooks: Represents audio books.
- Borrower: Represents a library member.
- Librarian: Manages the library.
- Loan: Represents a loan taken by a borrower.
- Library: Where books are stored.

- BookType: Enum for book categories (e.g., textbook, fiction).
- Fine: Represents overdue fines.
- FineStatus: Enum for fine status (paid/unpaid).

Relationships:

- Association: General relationships (e.g., a book can have multiple loans).
- Aggregation: "Has-a" relationship (e.g., a library has books).
- Composition: Strong "has-a" relationship (e.g., a loan includes a fine).

Methods:

- Each class has specific methods, like checkAvailability() in the Book class.

3.4.State Diagram

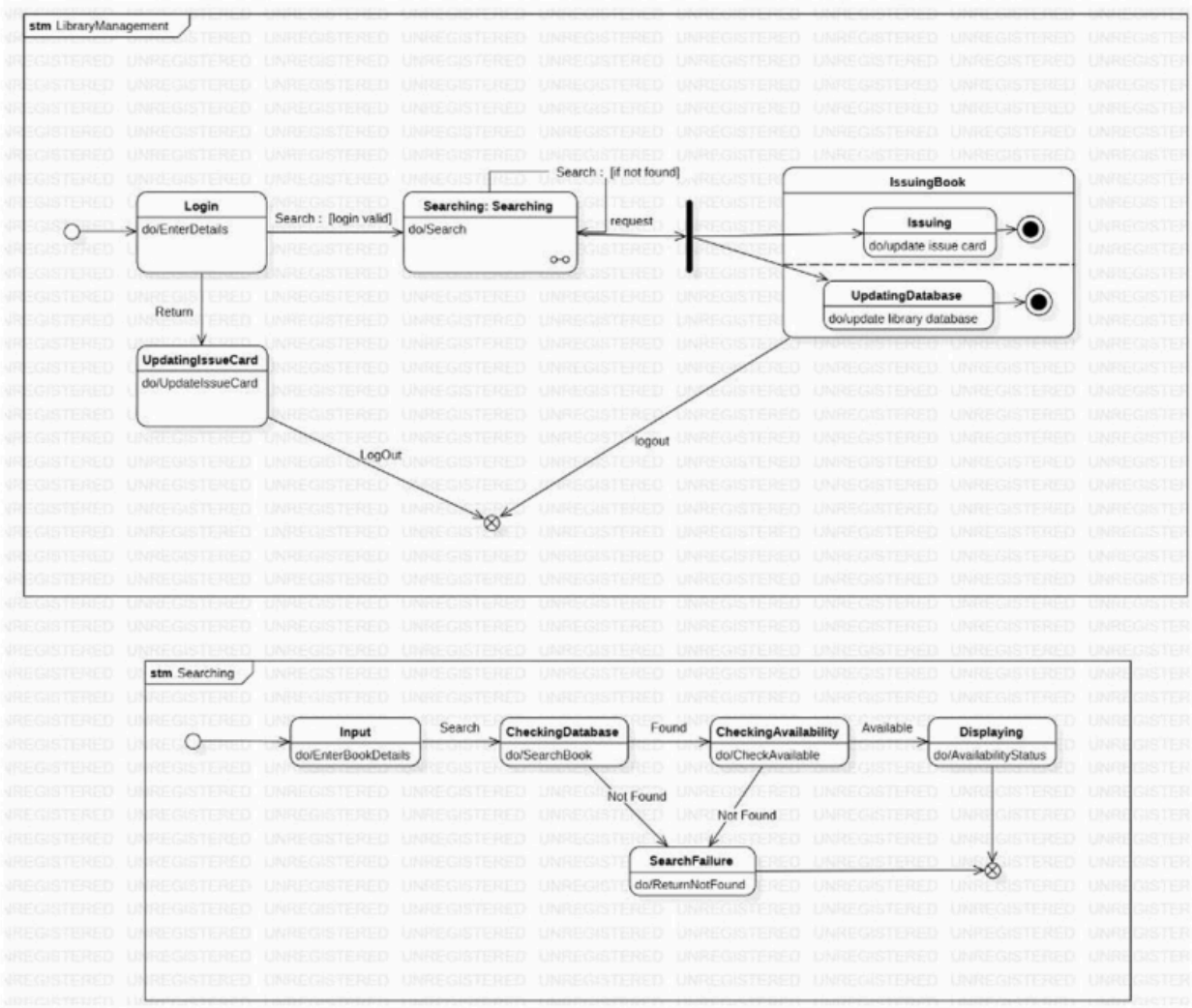


Fig.3.4.1

Brief Description

1. Book Available: The book is in the library's inventory.
2. Book Issued: The book is borrowed by a user.
3. Book Returned: The book is returned and the inventory is updated.
4. Overdue Fine Imposed: A fine is applied if the book is not returned on time.

3.5.Use Case Diagram

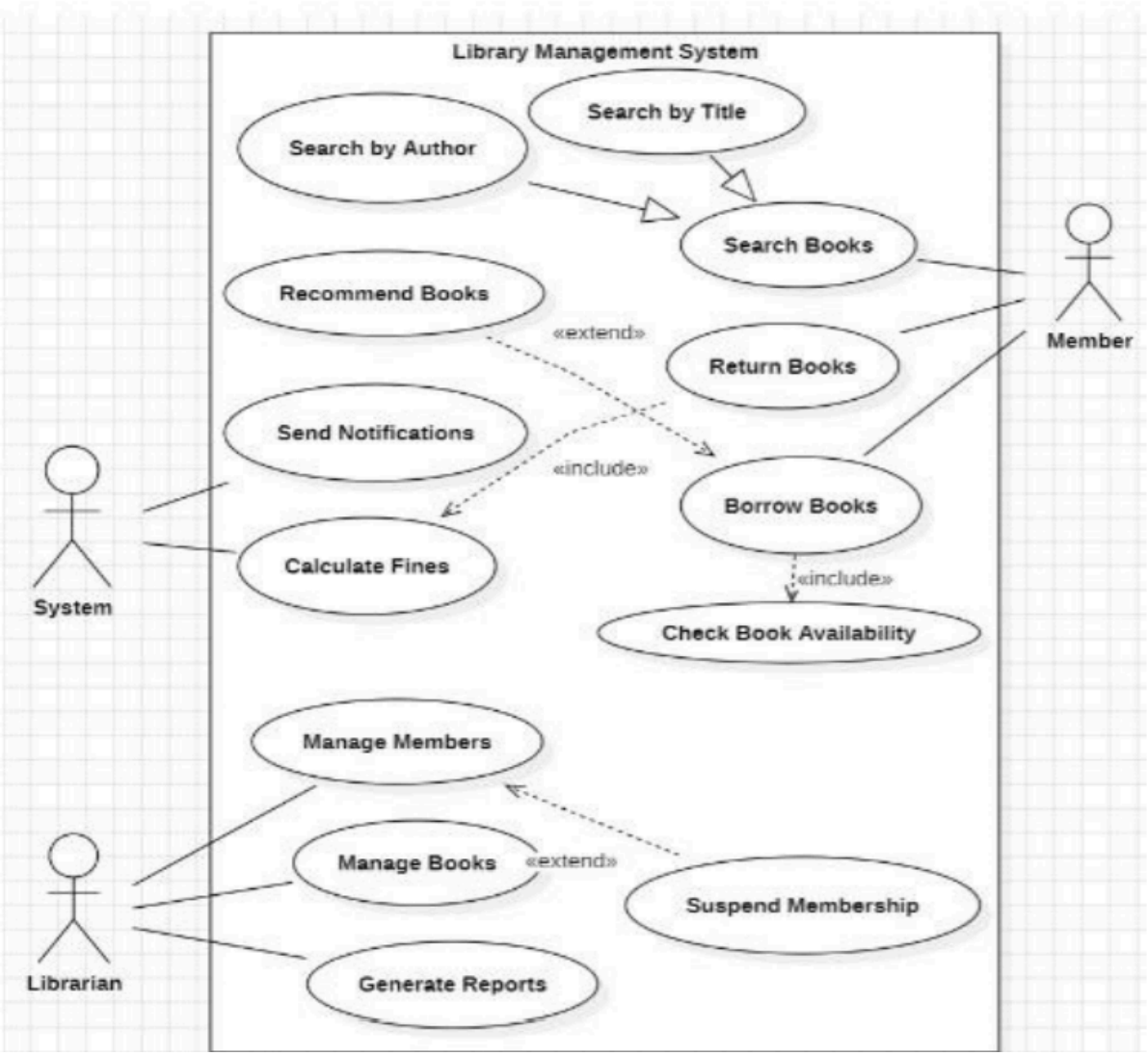


Fig.3.5.1

Brief Description

Actors:

- Users: Library members who interact with the system.
- Administrators: Manage library operations.

Use Cases:

- Search Book: Users search for available books.
- Borrow Book: Users borrow books from the library.
- Return Book: Users return borrowed books.
- Pay Fine: Users pay fines for overdue books.
- Add/Remove Book: Administrators update the library inventory.
- Generate Reports: Administrators create reports on library activities.

3.6.Sequence Diagram

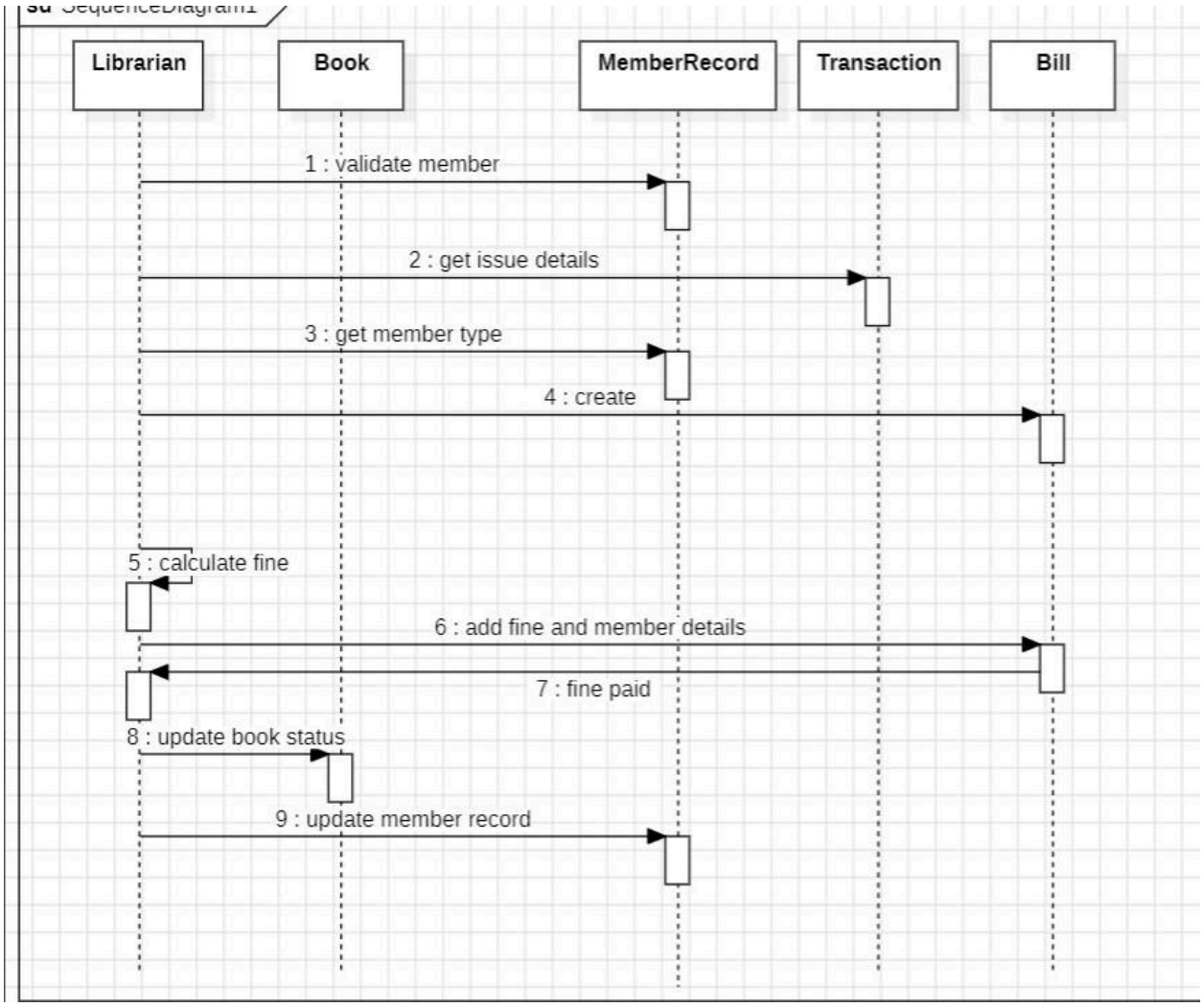


Fig.3.6.1

Brief Description

- The interaction begins with the librarian validating the member.
- The librarian retrieves issue details, member type, and calculates the fine.
- The fine and member details are added to the bill.
- The librarian marks the book as returned.
- The librarian updates the member record.

Components Involved:

- Librarian
- Book
- MemberRecord
- Transaction
- Bill

3.7.Activity Diagram

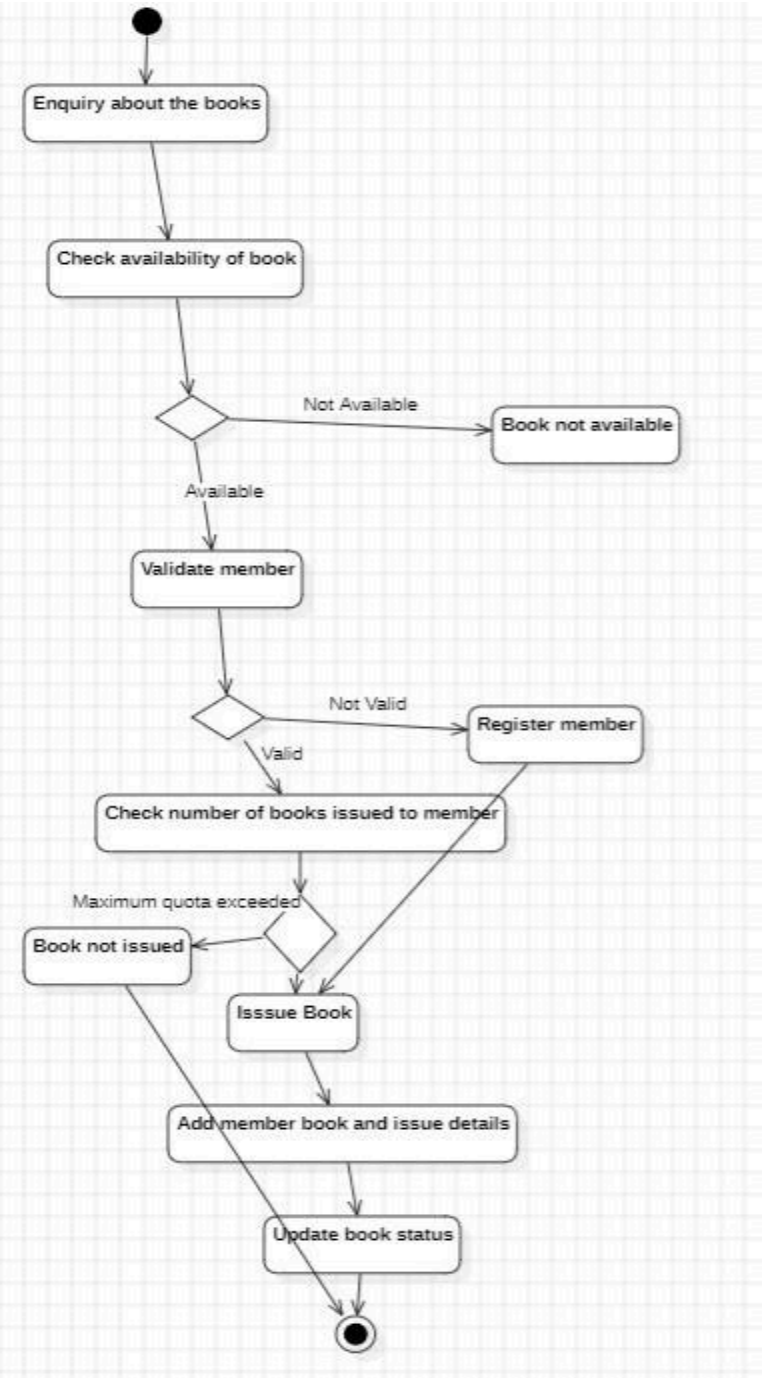


Fig.3.7.1

Brief Description

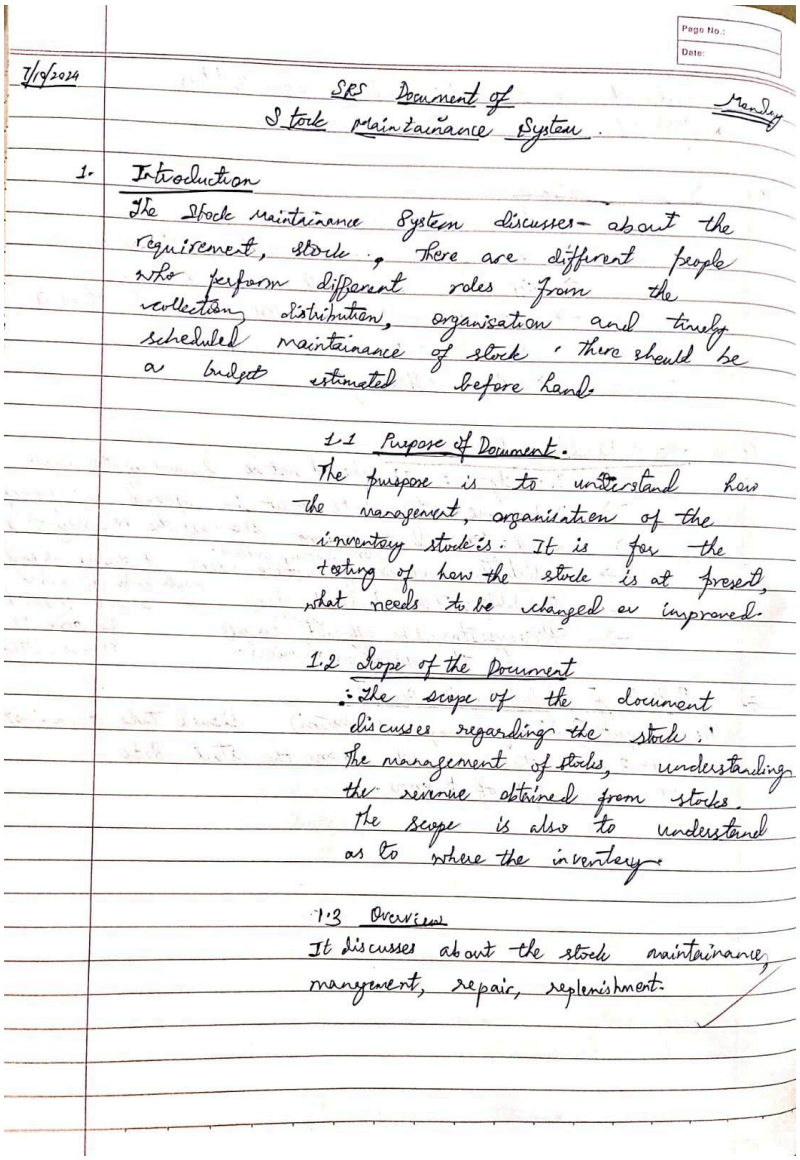
1. User returns a borrowed book.
2. System checks the return date.
3. If overdue:
    - Calculate the fine.
    - Notify the user.
4. Update the book status to "Available".
5. Update the user's borrowing history.

4.STOCK MAINTENANCE SYSTEM

4.1.Problem Statement

The Stock Maintenance System (SMS) is designed to efficiently track and manage inventory for businesses. It ensures real-time updates on stock availability, monitors stock levels, automates reorder processes, and provides detailed reporting to minimize inventory-related inefficiencies.

4.2.SRS-Software Requirements Specification



7/9/2024

SRS Document of
Stock maintainance System.

Monday

1. Introduction

The Stock maintainance System discusses- about the requirement, stock, There are different people who perform different roles from the collection, distribution, organisation and timely scheduled maintainance of stock, there should be a budget estimated before hand.

1.1 Purpose of Document.

The purpose is to understand how the management, organisation of the inventory stock is. It is for the testing of how the stock is at present, what needs to be changed or improved.

1.2 Scope of the Document

The scope of the document discusses regarding the stock, the management of stocks, understanding the revenue obtained from stocks. The scope is also to understand as to where the inventory.

1.3 Overview

It discusses about the stock maintainance, management, repair, replenishment.

2. General Description.

The general description discusses about the responsibilities, roles, services, repair, maintainance, procurement of items in the Stock maintainance System.

3. Functional Requirement.

-> Stock maintainance website which should get updated on a timely basis.
-> The Stock maintainance ~~Todo~~ ~~app~~ app should be created.

4. Interface Requirements

-> Website for Stock maintainance.
-> App version of the website which is efficient, functional, ~~fast~~.
-> BOBAS requirement

5. Performance Requirement.

-> High speed
-> efficiency
-> timely delivery of services
-> Proper management of the different services, roles and responsibilities.
-> No errors on website

6. Design Constraints

-> fully functional website
-> high speed, proper delivery of services
-> app version of website should be there.
-> should get updated regardless of whether there is an internet connection or no-
-> High efficiency.

7. Non-Functional Requirements

→ Security: The website / application should be secure and not be hacked by any person.

→ Performance: When there is high procurement then the website / app should handle it well.

→ Speed: The app / website page should be responsive and the required details about procurement to be updated.

→ The person operating the website and updating the details should be authenticated.

→ The time factor also comes here. It should take maximum 20-30 seconds to update the necessary details on the website. If it takes more than that, then it needs to be worked on.

8. Preliminary Schedule & Budget:

The project should take approximately 1-2 years to complete. The estimated budget for the same is ~~$85~~ $8500.

4.3.Class Diagram

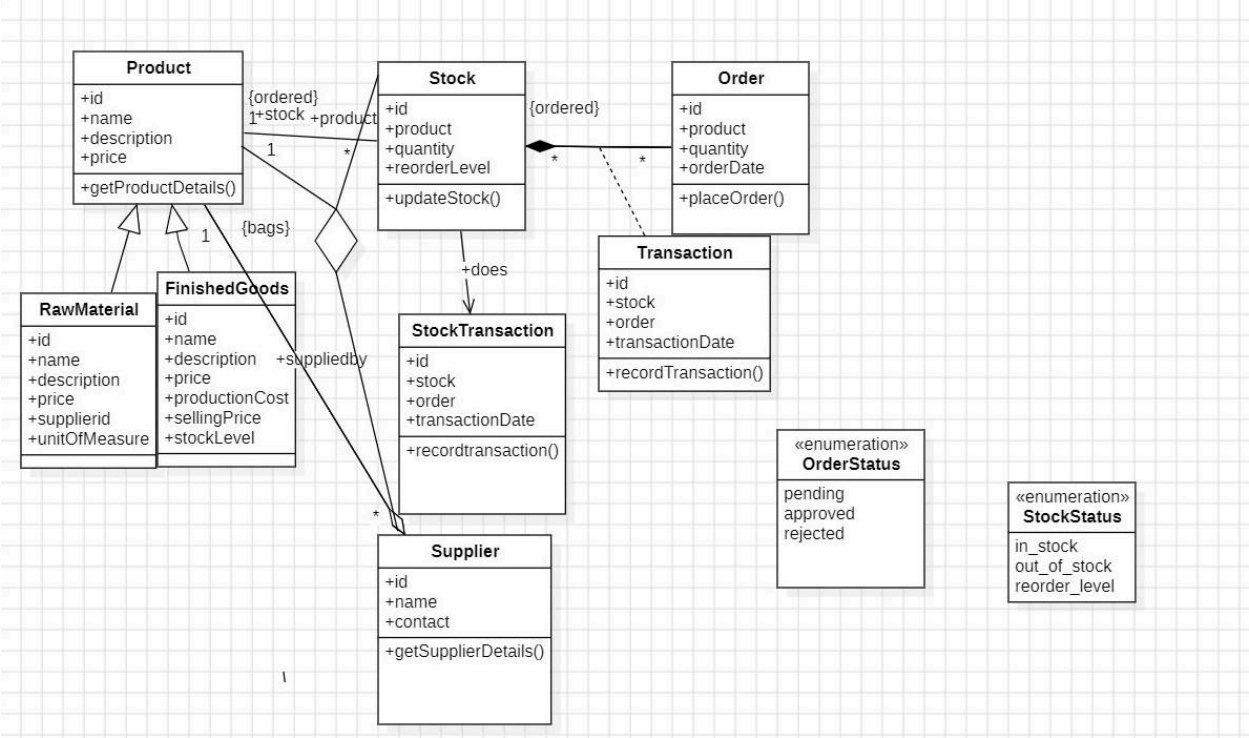


Fig.4.3.1

Brief Description

1. Entities:

- Product: Contains details like name, description, and price.
- RawMaterial: Represents raw materials with supplier info, unit of measure, and price.
- FinishedGoods: Represents finished products with stock levels, production cost, and selling price.
- Stock: Tracks product stock levels, quantities, and reorder points.
- Order: Manages customer orders with product details and quantity.
- Supplier: Stores supplier details like name and contact.
- StockTransaction and Transaction: Record stock changes and order-related transactions.

2. Key Relationships:

- A Product can also be a FinishedGood (1:1).
- A Product is linked to multiple Stock records (1:many).
- A Supplier provides multiple RawMaterials (1:many).
- A FinishedGood is composed of multiple RawMaterials (many:many).
- Each Order has multiple StockTransactions (1:many).
- A Transaction is tied to one StockTransaction (1:1).

3. Enumerations:

- OrderStatus: Tracks order states (pending, approved, rejected).
- StockStatus: Indicates stock levels (in stock, out of stock, reorder level).

4.4.State diagram

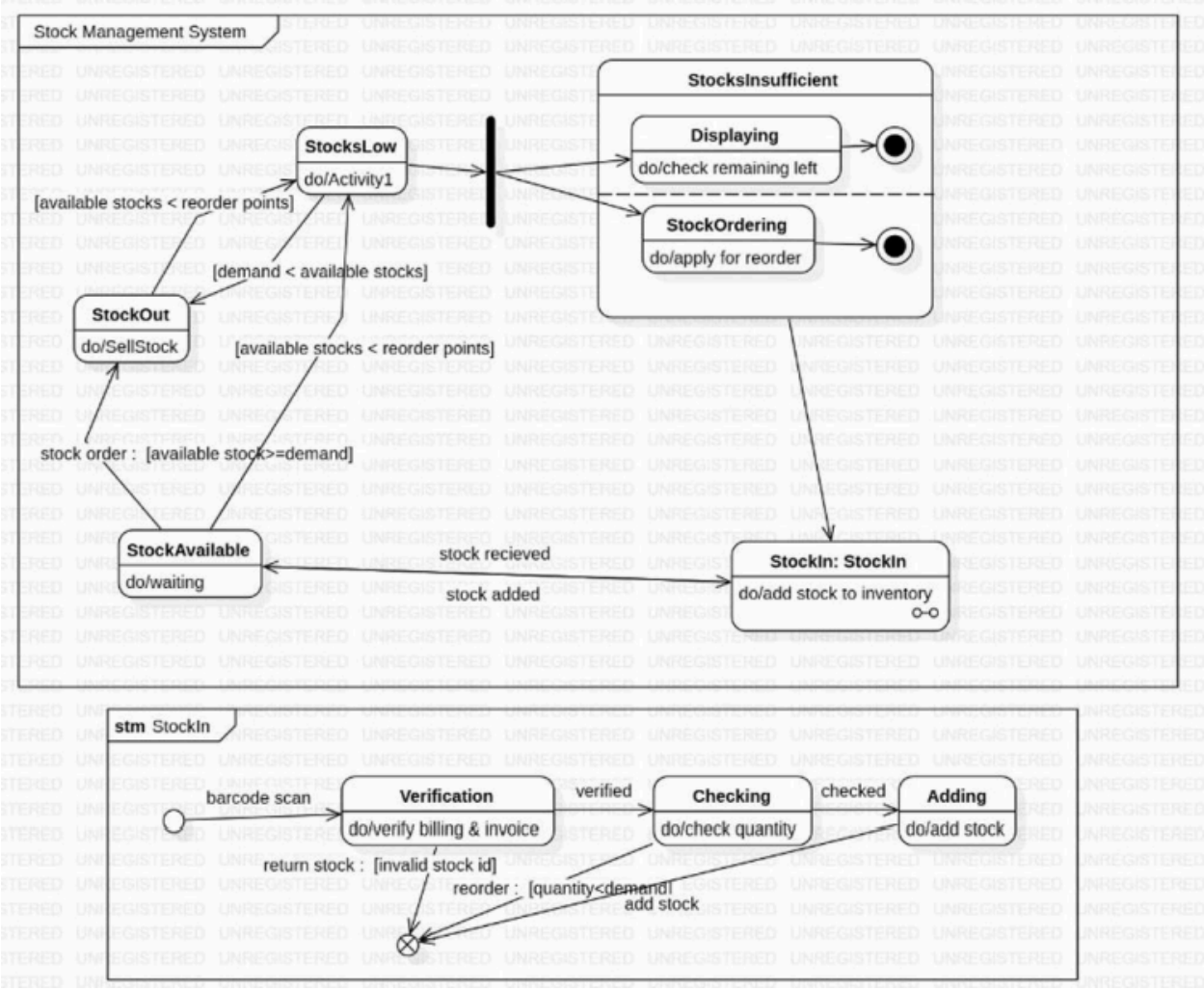


Fig.4.4.1

Brief Description

States:

1. Stock Added:
    - Represents the initial state when new stock is recorded in the inventory.
2. Stock Updated:
    - Occurs when stock levels change due to sales, returns, or manual adjustments.
3. Stock Low:
    - Triggered when stock levels fall below the reorder threshold, signaling the need for replenishment.
4. Reorder Processed:
    - Represents the completion of the reorder process where new stock is ordered, received, and added to the inventory.

Transitions:

- From Stock Added → Stock Updated: After an initial stock addition, levels are adjusted due to sales or returns.
- From Stock Updated → Stock Low: If updates cause stock to drop below the reorder threshold.
- From Stock Low → Reorder Processed: When a reorder is initiated and stock is replenished.
- From Reorder Processed → Stock Updated: Once the reorder is complete, stock levels are adjusted.

4.5.Use Case Diagram

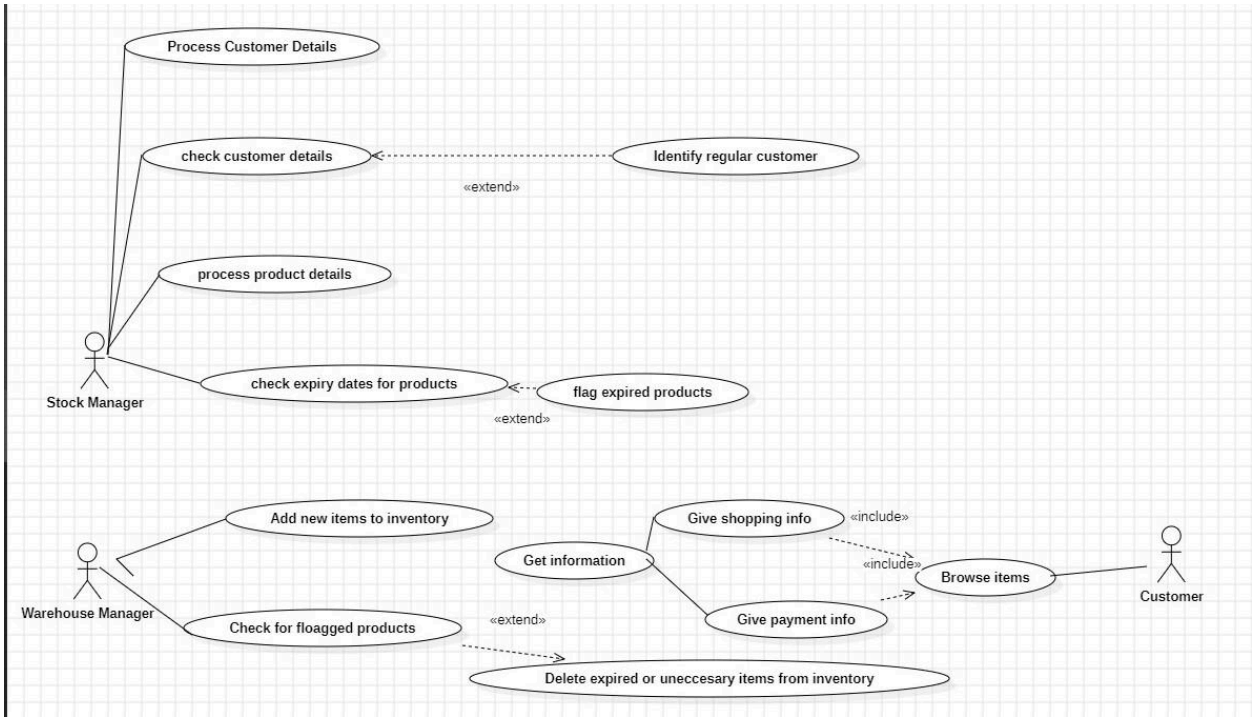


Fig.4.5.1

Brief Description

1. Process Customer Details:
    - Manages and stores customer information.
    - Extends: *Identify Regular Customer*.
2. Check Expiry Dates for Products:
    - Monitors product expiration dates.
    - Extends: *Flag Expired Products*.
3. Flag Expired Products:
    - Marks expired items in the inventory for removal.
4. Delete Expired or Unnecessary Items from Inventory:
    - Shared responsibility with the Warehouse Manager.

Warehouse Manager Responsibilities

1. Add New Items to Inventory:
    - Updates inventory with newly received items.
2. Check for Flagged Products:
    - Reviews products flagged as expired or unnecessary.
3. Delete Expired or Unnecessary Items from Inventory:
    - Shared responsibility with the Stock Manager.

Customer Responsibilities

1. Give Shopping Info:
    - Provides information about shopping preferences and orders.
2. Get Information:
    - Accesses product details and availability.
3. Browse Items:
    - Searches and explores available products.
    - Included In: *Give Payment Info*.
4. Give Payment Info:
    - Completes transactions and payments.

Relationships

- Shared Use Case:
  Delete Expired or Unnecessary Item*s from Inventory* is used by both the Stock Manager and Warehouse Manager.
- Include:
  Browse Items is included in Give Payment Info.
- Extend:
    - Identify Regular Customer extends Process Customer Details.
    - Flag Expired Products extends Check Expiry Dates for Products.

4.6.Sequence diagram

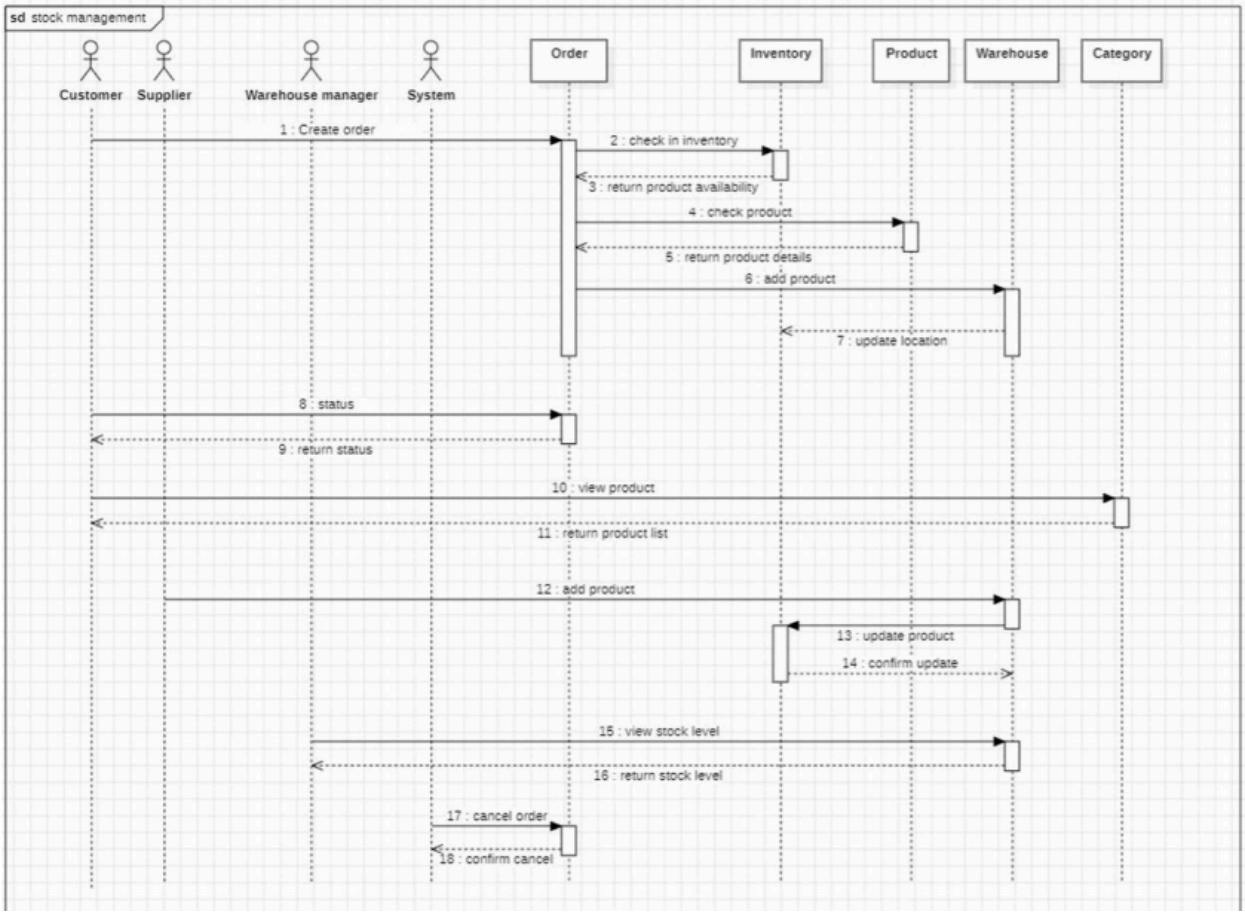


Fig.4.6.1

Brief Description

The sequence diagram illustrates the process of recording a sale:

1. Staff scans the product barcode.

2. The system fetches product details from the database.

3. Staff enters the quantity sold.

4. The system deducts the quantity from stock and updates the database.

5. The system generates a transaction record and updates sales data.

4.7.Activity Diagram

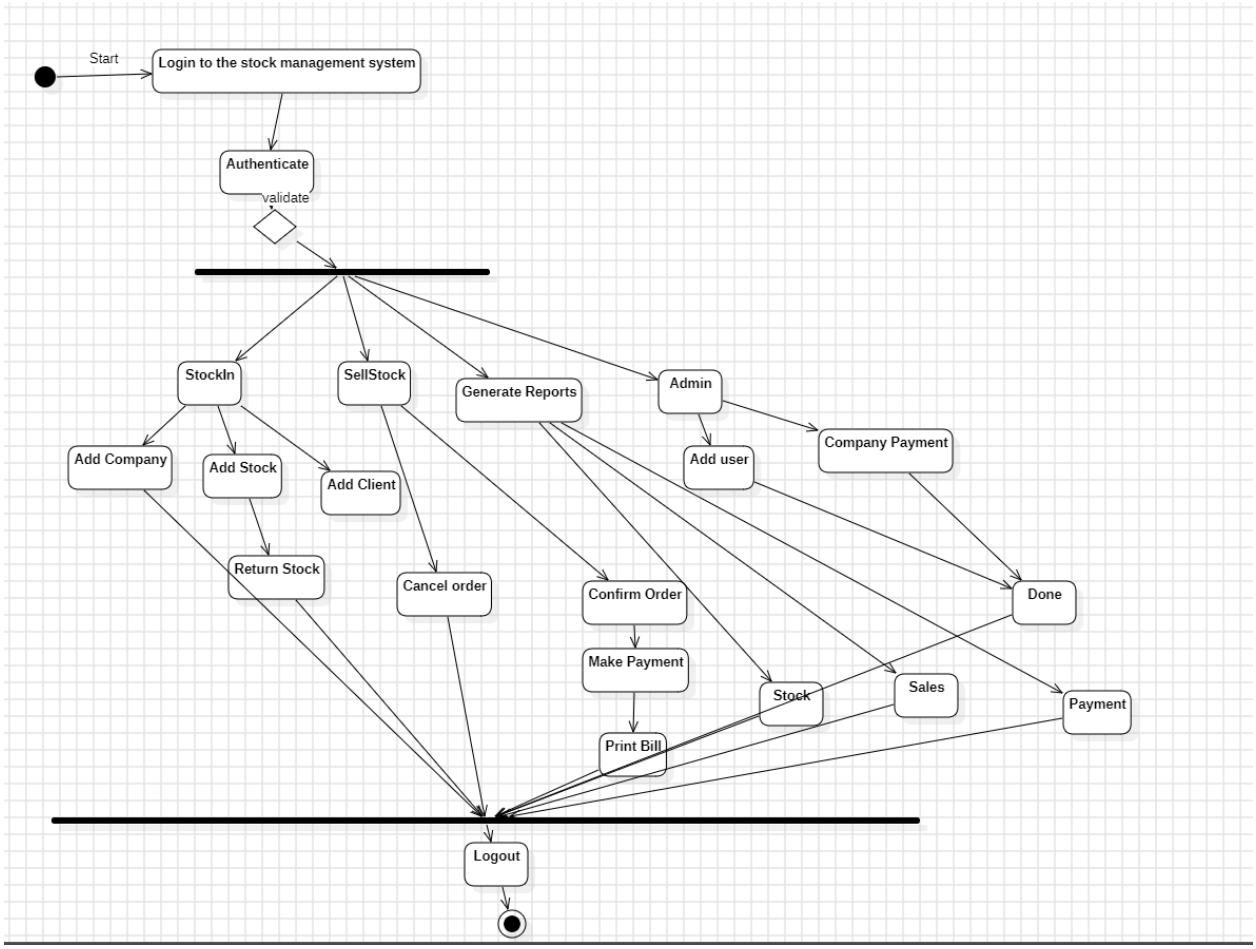


Fig.4.7.1

Brief Description

1.Login:

- The system starts with user authentication.

2.Stock Management:

- Includes adding stock, selling stock, and returning stock.

3.Company and Client Management:

- Allows adding and managing companies and clients.

4.Order and Payment Processing:

- Supports order confirmation, payment handling, and reporting.

5.Administrative and Logout Functions:

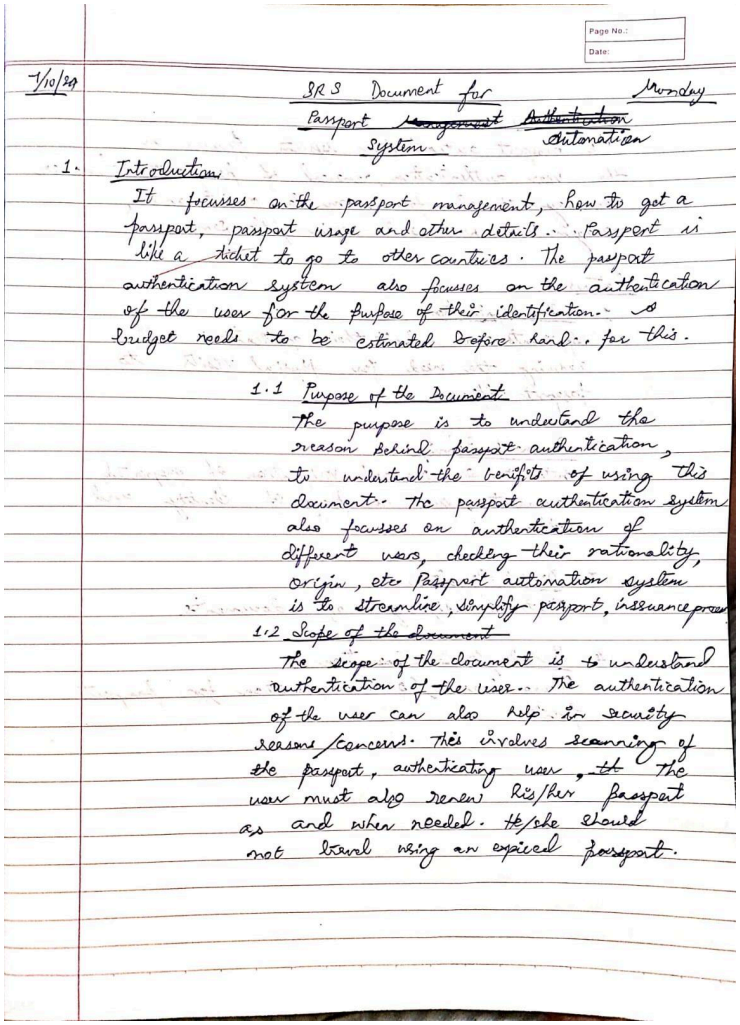
- Includes adding users, managing company payments, and logging out.

5.PASSPORT AUTOMATION SYSTEM

5.1.Problem Statement

The Passport Automation System (PAS) is designed to simplify and streamline the process of passport application, verification, issuance, and renewal. It provides an online platform for applicants to submit their details, track application progress, and receive updates, while enabling officials to manage and verify applications efficiently.

5.2.SRS-Software Requirements Specification



7/10/24 — SRS Document for Passport ~~Management~~ ~~Authentication~~ Automation System — Monday

1. Introduction

It focusses on the passport management, how to get a passport, passport usage and other details. Passport is like a ticket to go to other countries. The passport authentication system also focusses on the authentication of the user for the purpose of their identification. A budget needs to be estimated before hand. for this.

1.1 Purpose of the Document

The purpose is to understand the reason behind passport authentication, to understand the benifits of using this document. the passport authentication system also focusses on authentication of different users, checking their nationality, origin, etc Passport automation system is to streamline, simplify passport, issuance pro

1.2 Scope of the document

The scope of the document is to understand authentication of the user. The authentication of the user can also help in security reasons/concerns. This involves scanning of the passport, authenticating user, the user must also renew his/her passport as and when needed. He/she should not travel using an expired passport.

1.3 Overview :-

The passport authentication system focuses on the user authentication, renewal of passport whenever necessary. It also focusses on the maintenance, management as well as the repair of the same.

Features :-

1. Online Application
Applicants can submit their applications online, reducing the need for physical visits to passport offices.

2. Document Scanning
Automated scanning and verification of supported documents, such as proof of identity and address.

3. Digital Signature
Electronic signing of documents.

4. Barcode / QR Code Generation :-
Secure barcode / QR Code generation for passport printing

Page No.:
Date:

2. General Description

It is a computerized platform designed to streamline and automate the passports application, processing, and issuance process. It integrates various technologies to provide a secure, efficient, and user-friendly experience for citizens, passport authorities, and other stakeholders.

3. Functional Requirements

→ Passport automation website should get updated on a timely basis.

→ A passport ~~autho~~ automation fully functional application must be created.

4. Interface Requirements

→ A website for Passport Automation System.

→ App version of the website that is fully functional, as well as efficient and fast.

→ RDBMS requirement

5. Performance Requirements

→ User authentication: < 2 seconds

→ Application processing: 1000+/day, 90% auto verified

→ Biometric data capture: fingerprints <10 seconds; facial recognition <5 seconds

→ Document Scanning/verification: 10+/minute, <50 seconds.

→ Payment processing: <2 seconds

→ Reporting/analytics: report generation < 1 min

6. Design Constraints

→ fully functional website
→ High speed, proper delivery of services
→ Should work regardless of whether there is an internet connection or not.
→ App version of the website must be there.
→ High efficiency, no errors in the website.

7. Non-Functional Requirements.

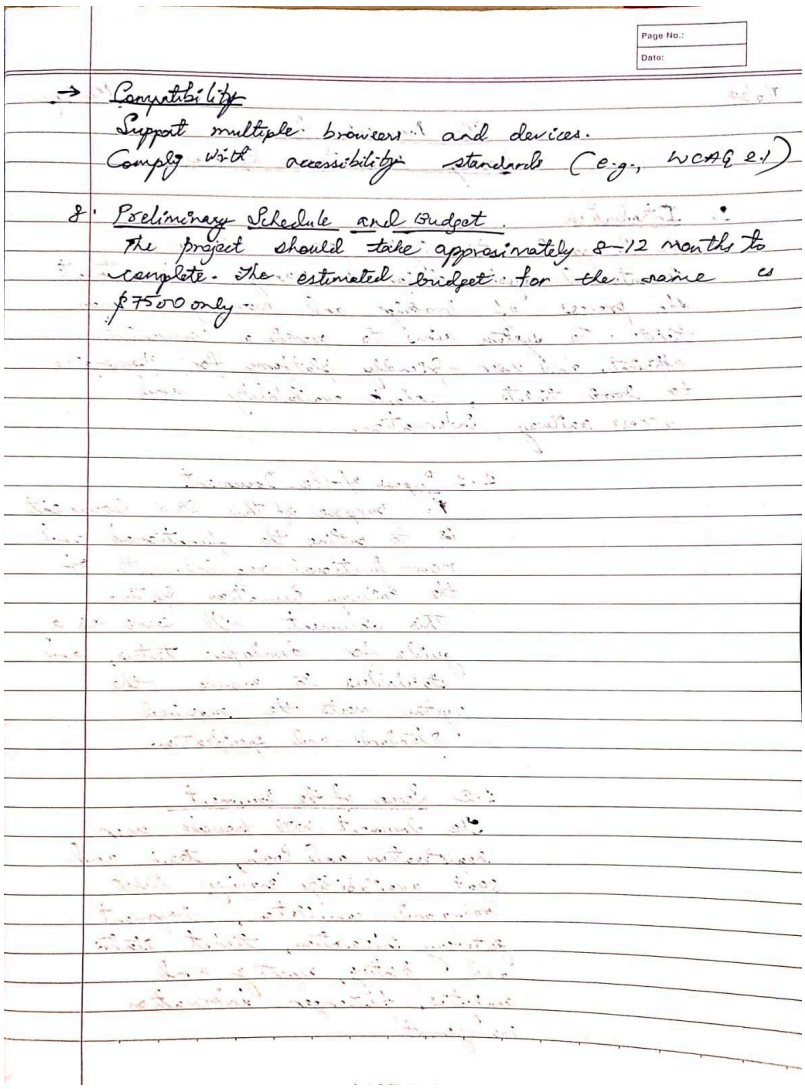
→ Security:
Encrypt sensitive data (eg: biometrics, personal info)
Comply with international security standards. (e.g., ISO 27001)

→ Scalability
Handle 10,000 + concurrent users.
Scale up/down based on demand.

→ Availability
- 99.9% uptime
- 24/7 operation

→ Performance
Average response time < 2 seconds
Peak load handling: 500 + requests/second

→ Usability
User friendly interfaces
intuitive navigation


Page No.:
Date:

→ Compatibility
Support multiple browsers and devices.
Comply with accessibility standards (e.g., WCAG 2.1)

8. Preliminary Schedule and Budget
The project should take approximately 8–12 months to complete. The estimated budget for the same is $7500 only.

5.3.Class Diagram

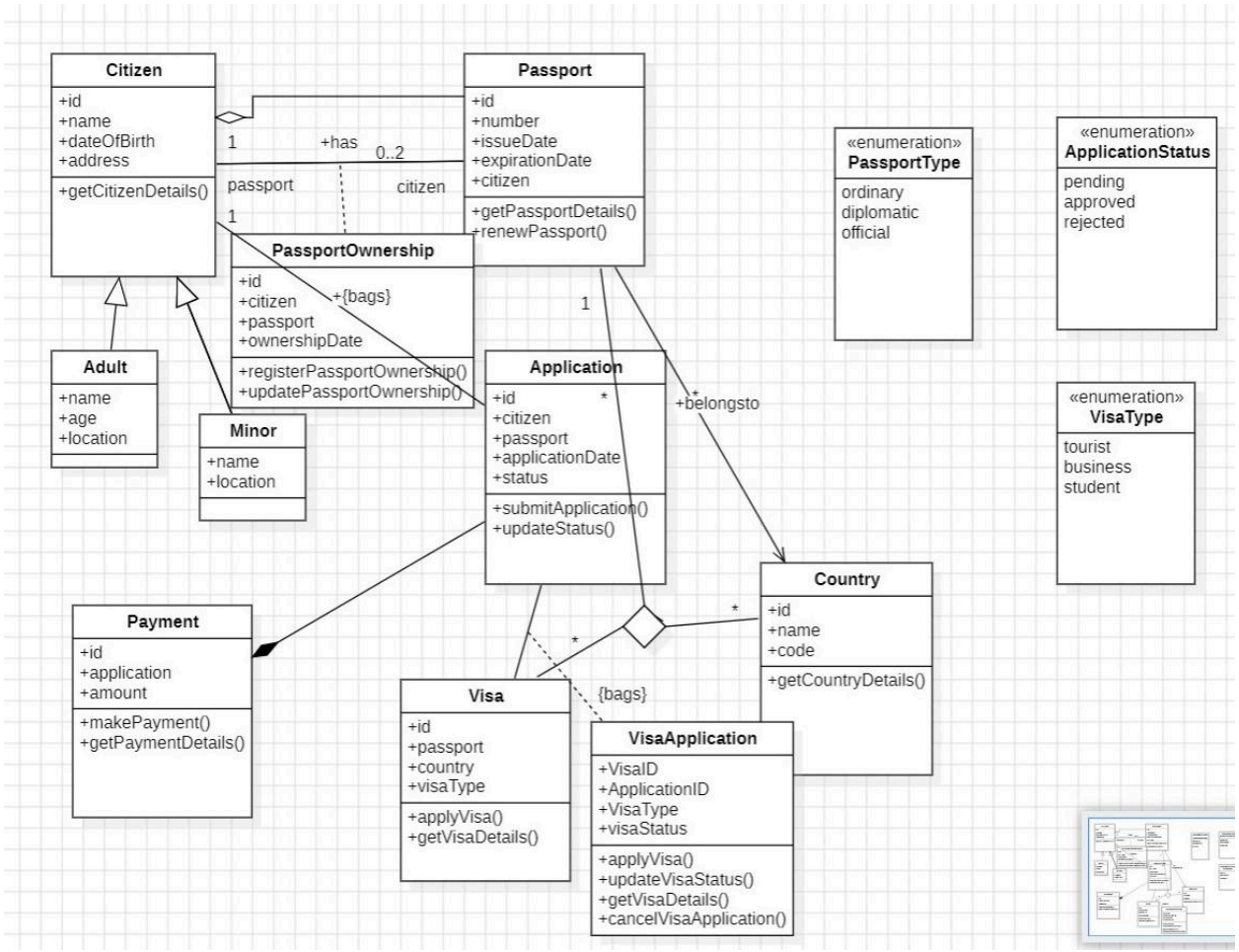


Fig.5.3.1

Brief Description

- Entities: The diagram represents various entities such as Citizens, Passports, Applications, Visas, and Countries.

- Relationships: It illustrates relationships between these entities, such as a Citizen having multiple Passports, an Application belonging to a Citizen, and a Visa being associated with a Country.

- Attributes: Each entity has specific attributes, like a Citizen having a name and address, and a Passport having a number and expiration date.

- Enumerations: The diagram uses enumerations to define sets of values for specific attributes, such as the types of Passports and the statuses of Applications.

- Operations: It outlines potential operations that can be performed on the data, such as registering a PassportOwnership, submitting an Application, and renewing a Passport.

5.4.State Diagram

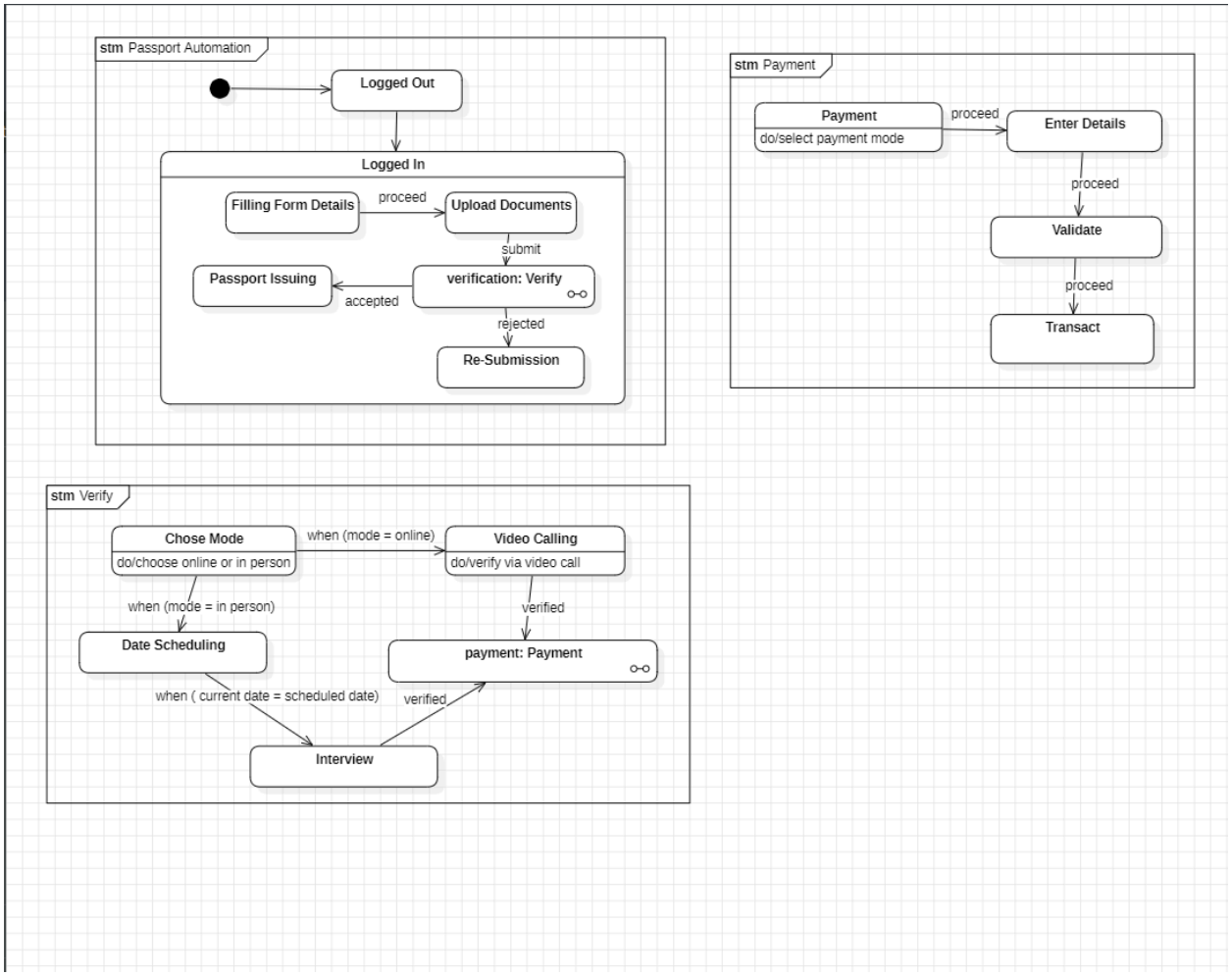


Fig.5.4.1

Brief Description

- User Journey: The system guides users through the passport application process, starting with login or registration and progressing through form filling, document upload, verification, and payment.

- Verification Modes: Users can choose between online or in-person verification. Online verification involves video calling with an officer, while in-person verification requires scheduling an interview.

- Payment Integration: The system includes a payment module where users can select their preferred payment method and proceed with the transaction.

- State Transitions: The diagram illustrates the various states the application can be in, such as "Logged Out," "Logged In," "Filling Form Details," "Upload Documents," "Verification," "Payment," and "Interview." Transitions between these states are triggered by user actions or system processes.

- Error Handling: The system includes mechanisms for handling errors, such as rejected applications, which can be resubmitted after addressing the issues.

5.5.Use Case Diagram

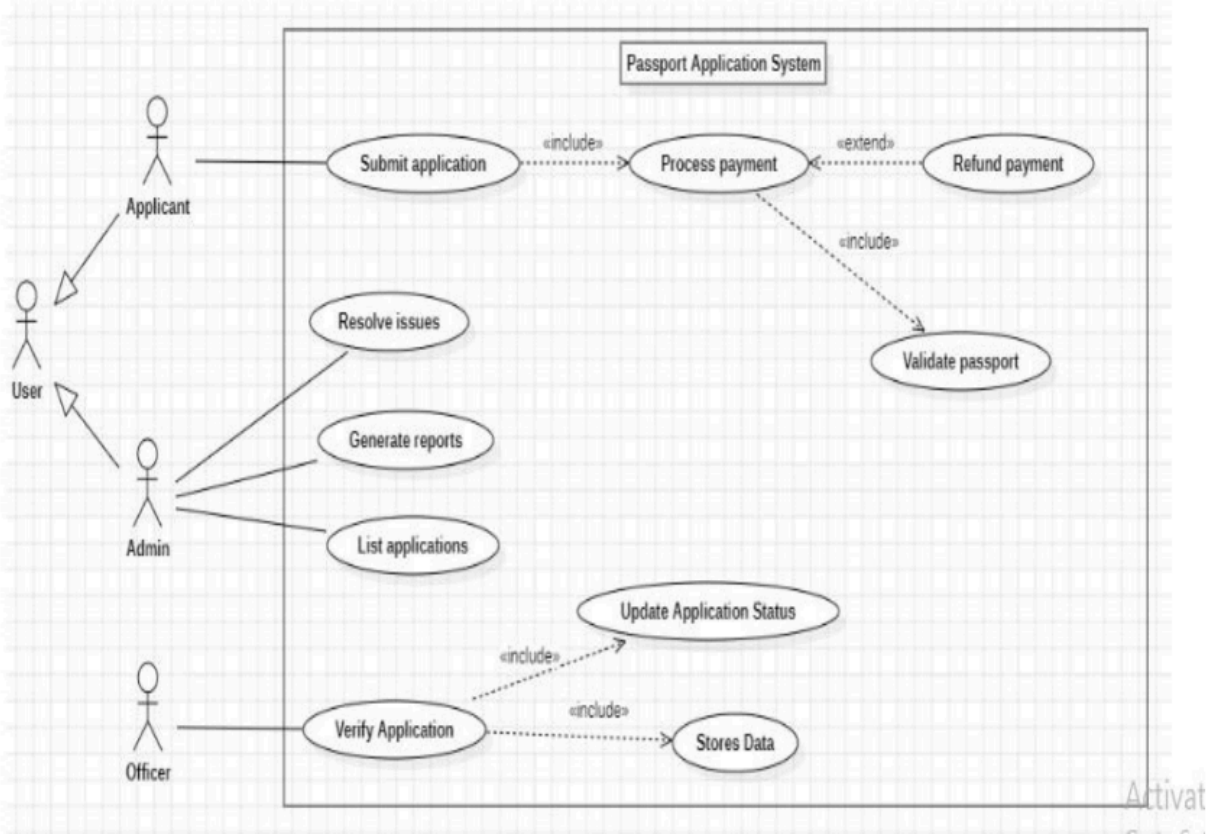


Fig.5.5.1

Brief description

Actors:

- **Applicants**
- **Officials**
- **Police Department**

Use Cases:

1. Submit Application
    - Actor: Applicants
2. Upload Documents

    - Actor: Applicants
3. Pay Fees
    - Actor: Applicants
4. Track Status
    - Actor: Applicants
5. Verify Documents
    - Actor: Officials
6. Manage Police Clearance
    - Actor: Police Department
7. Issue Passport
    - Actor: Officials

5.6.Sequence Diagram

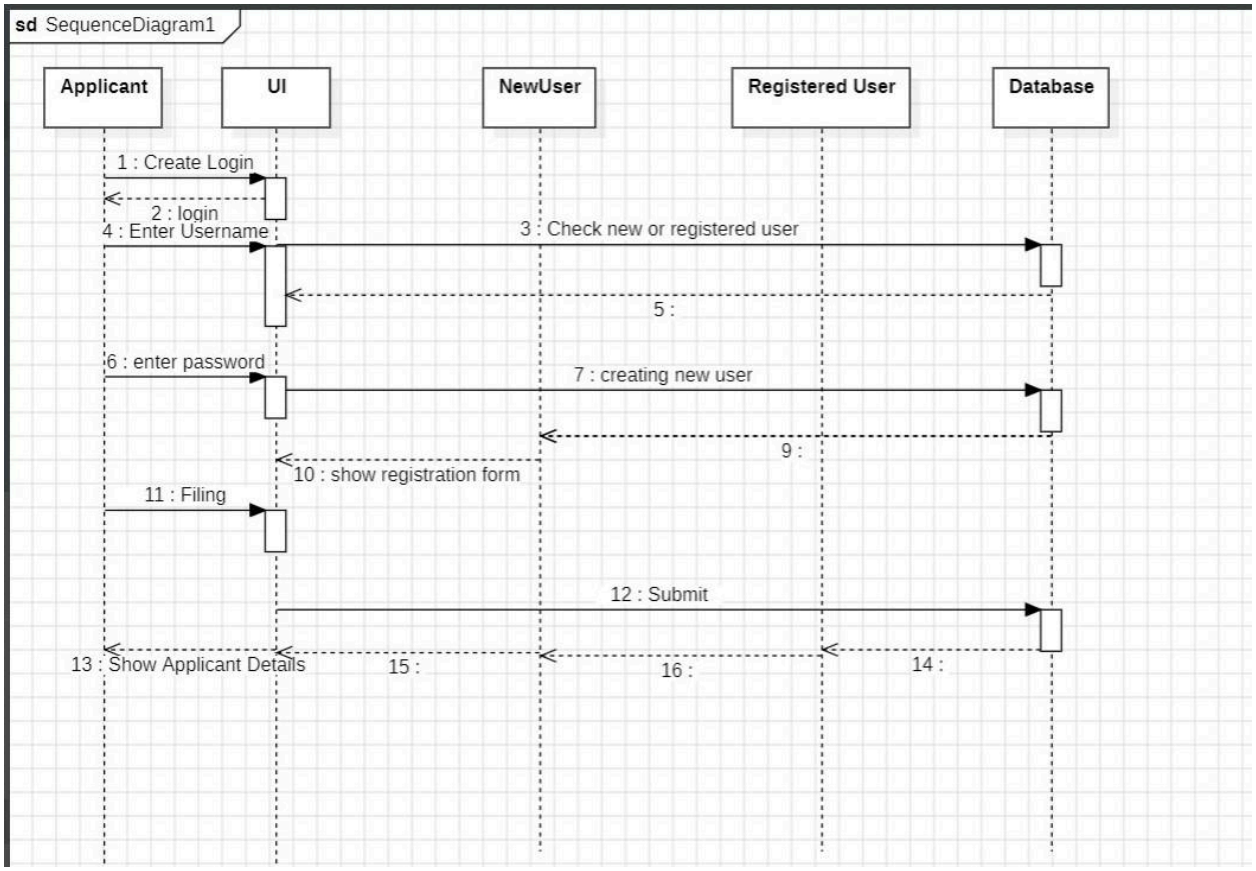


Fig.5.6.1

Brief Description

User Interaction: The sequence starts with the Applicant initiating the process by creating a login.

User Authentication: The UI checks if the user is new or registered. If new, the UI presents a registration form.

Registration: The NewUser submits registration details, which are stored in the Database.

Login & Data Retrieval: The Registered User logs in, and the UI retrieves their details from the Database.

Data Display: The UI then displays the Applicant's details to the user.

5.7.Activity Diagram

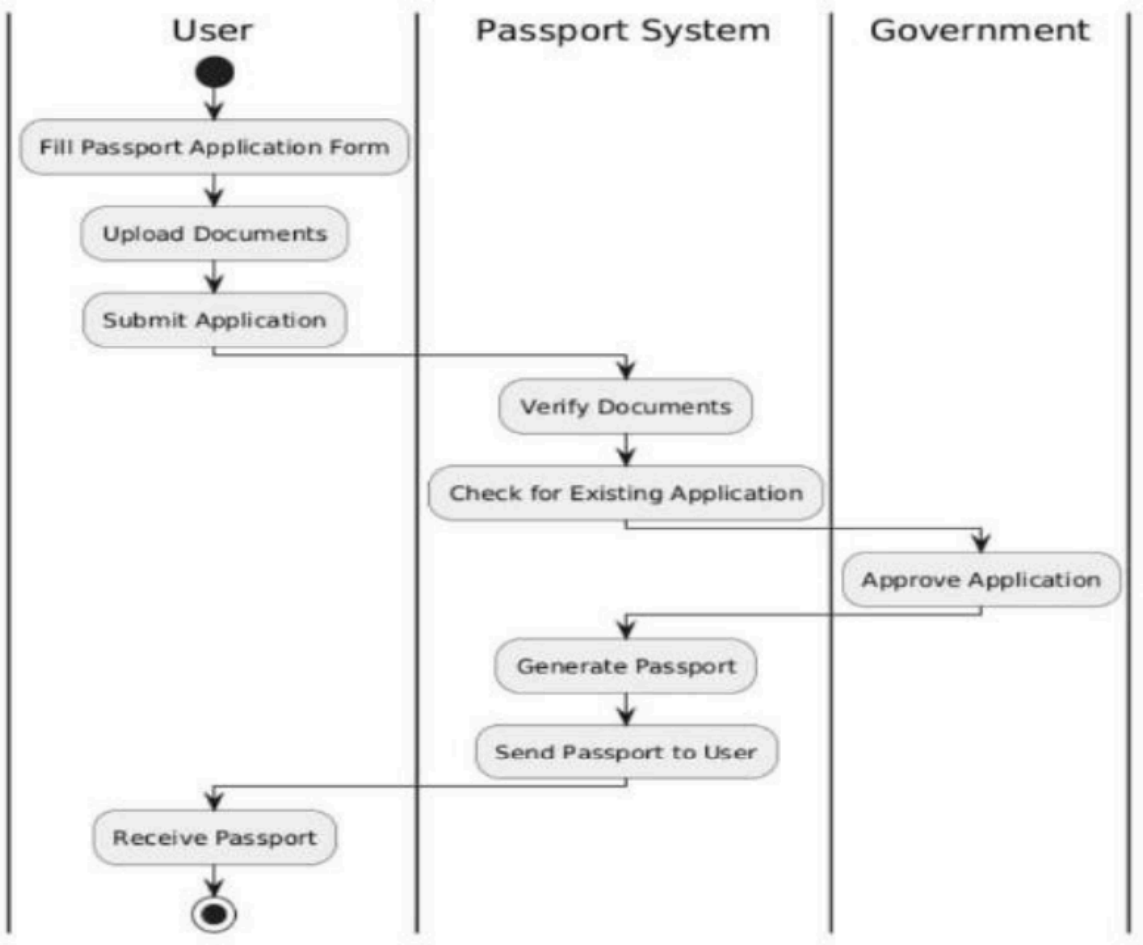


Fig.5.7.1

Brief Description

1.Applicant Submits Application

- The applicant submits the application through the system.

2.System Forwards Application

- The system forwards the submitted application to officials for document verification.

3.Document Verification

- Officials verify the documents and take one of the following actions:
  a. Approve Application
  b. Reject Application

4.Initiate Police Clearance

- For approved applications, the system initiates the police clearance process.

5.Police Clearance Status

- The police department sends the clearance status to the system.

6.Update Application Status

- Based on the police clearance, officials update the application status to "Approved".

7.Generate and Issue Passport

- The system generates and issues the passport for applications marked as "Approved".