

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



Lab Record

Object-Oriented Modeling And Design

Submitted in partial fulfillment for the 5th Semester Laboratory

Bachelor of Engineering
in
Computer Science and Engineering

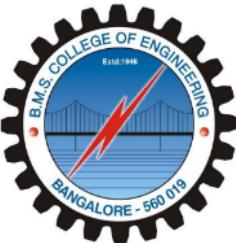
Submitted by:

Siri Sathish

1BM22CS280

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 **Siri Sathish (BM22CS280)** during the 5th Semester Oc t24-Jan2025.

Signature of the Faculty Incharge:
Prameetha Pai
Assistant Professor
Department of Computer Science and Engineering
B.M.S. College of Engineering, Bangalore

Table of Contents

Title
Hotel Management System
Credit Card Processing
Library Management System
Stock Management System
Passport Automation System

1. Hotel Management System

SRS-Software Requirements Specification Document

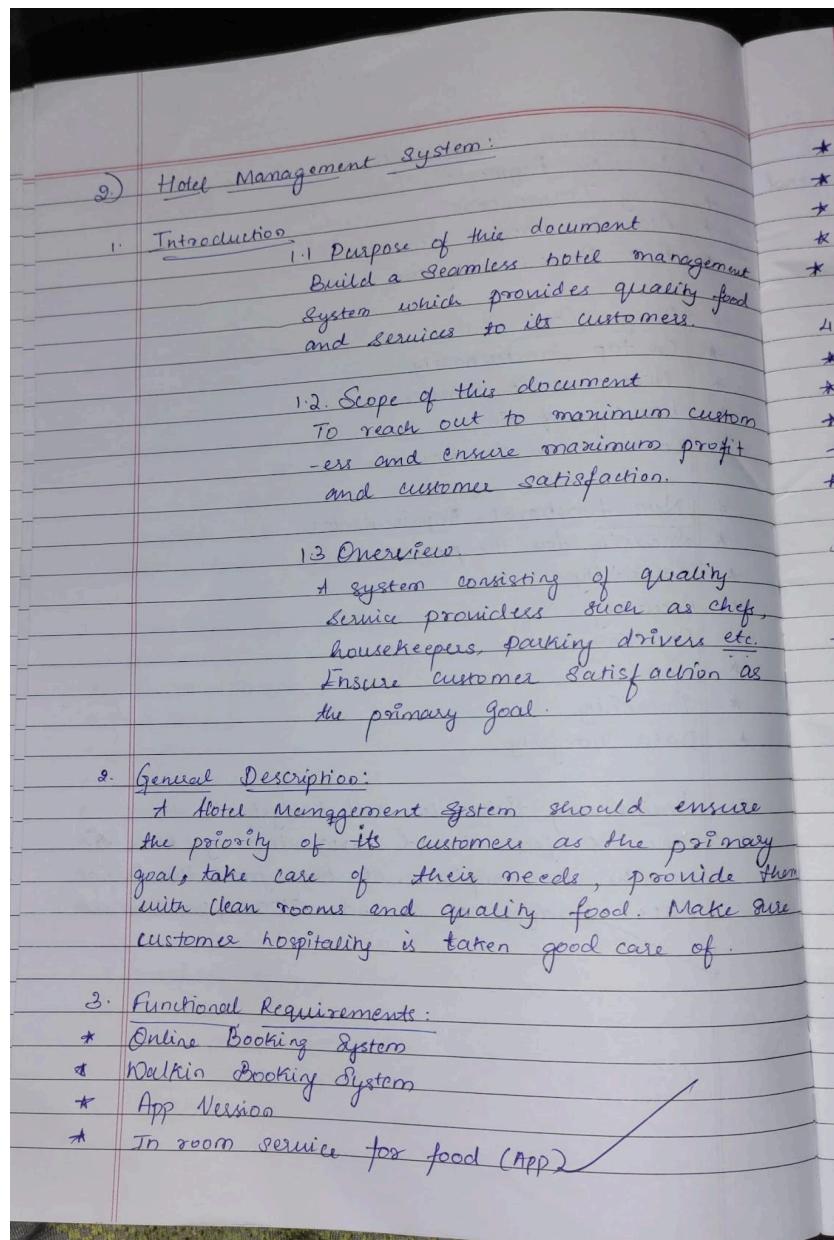


Figure 1.1

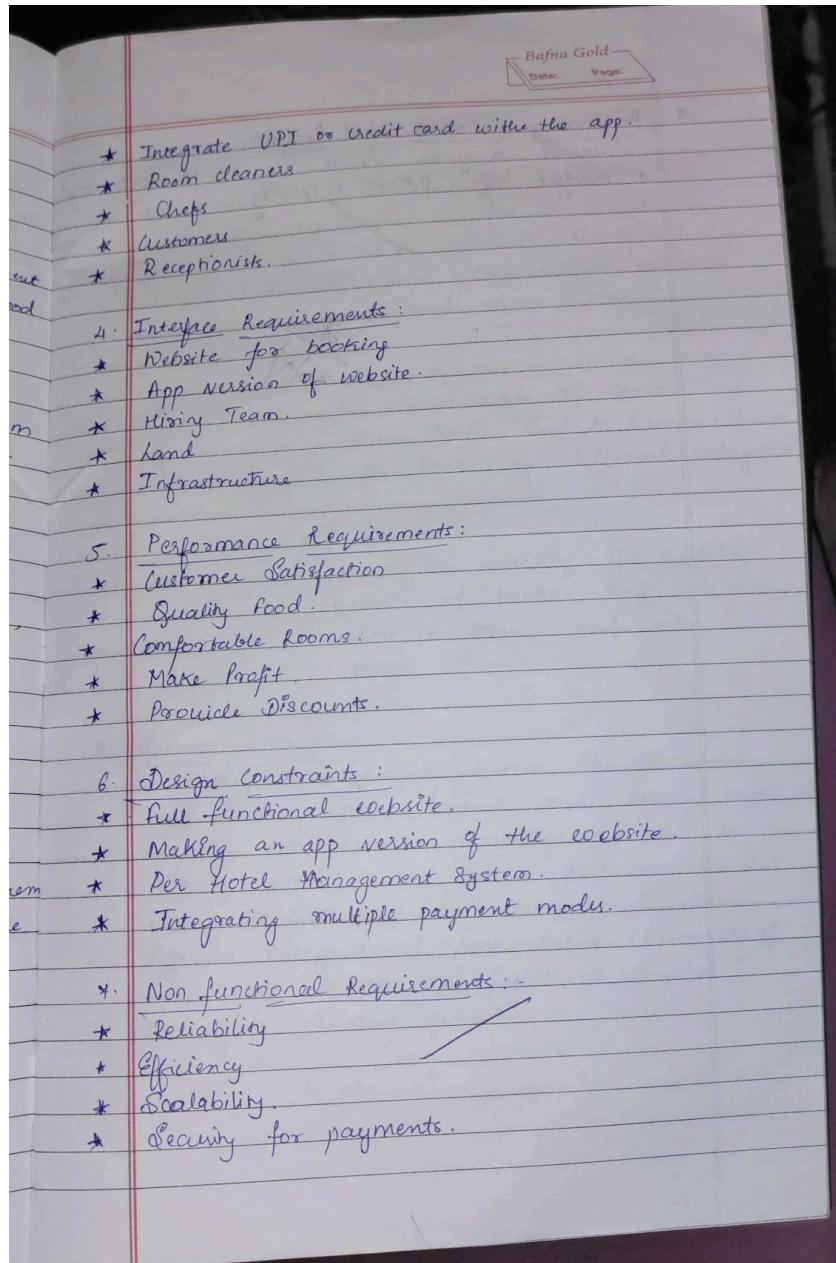


Figure 1.2

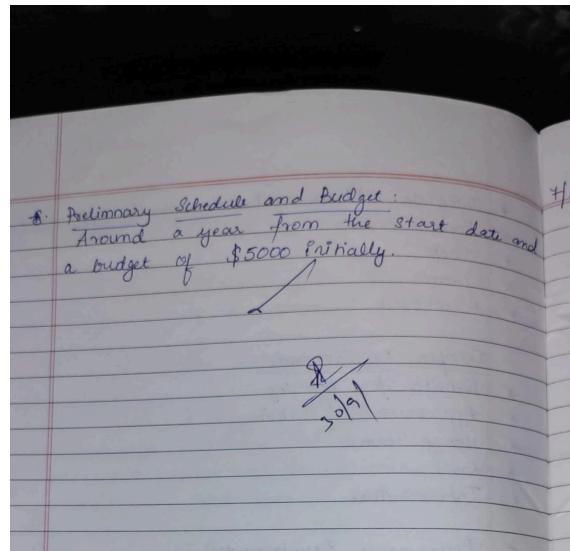


Figure 1.3

Class Diagram

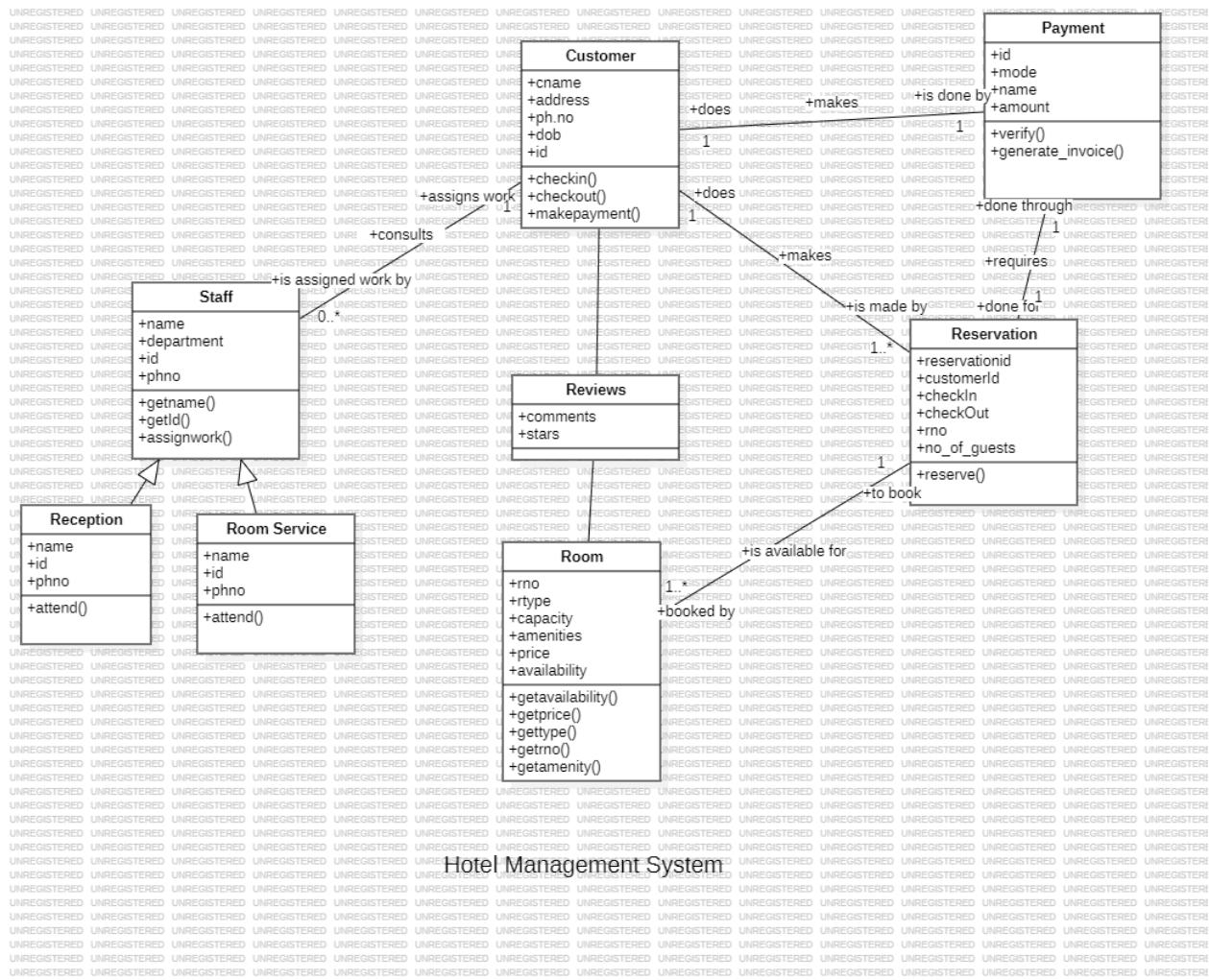


Figure 1.4

Description:

- Customer - Class that has attributes and functions indicative of the customer who books the hotel.
 - Room - Indicates the room that can be booked.
 - Payment - A class that has attributes and functions for a transaction.
 - Staff - Indicative of hotel staff.

- Reservation - Contains attributes and functions aiding the process of reservation.
- Generalization - Two classes Reception and Room Service inherit properties of the Staff class.
- Composition - The Offers class is a composition of Reservation i.e if reservation ceases to exist this class does not exist.

State Diagram

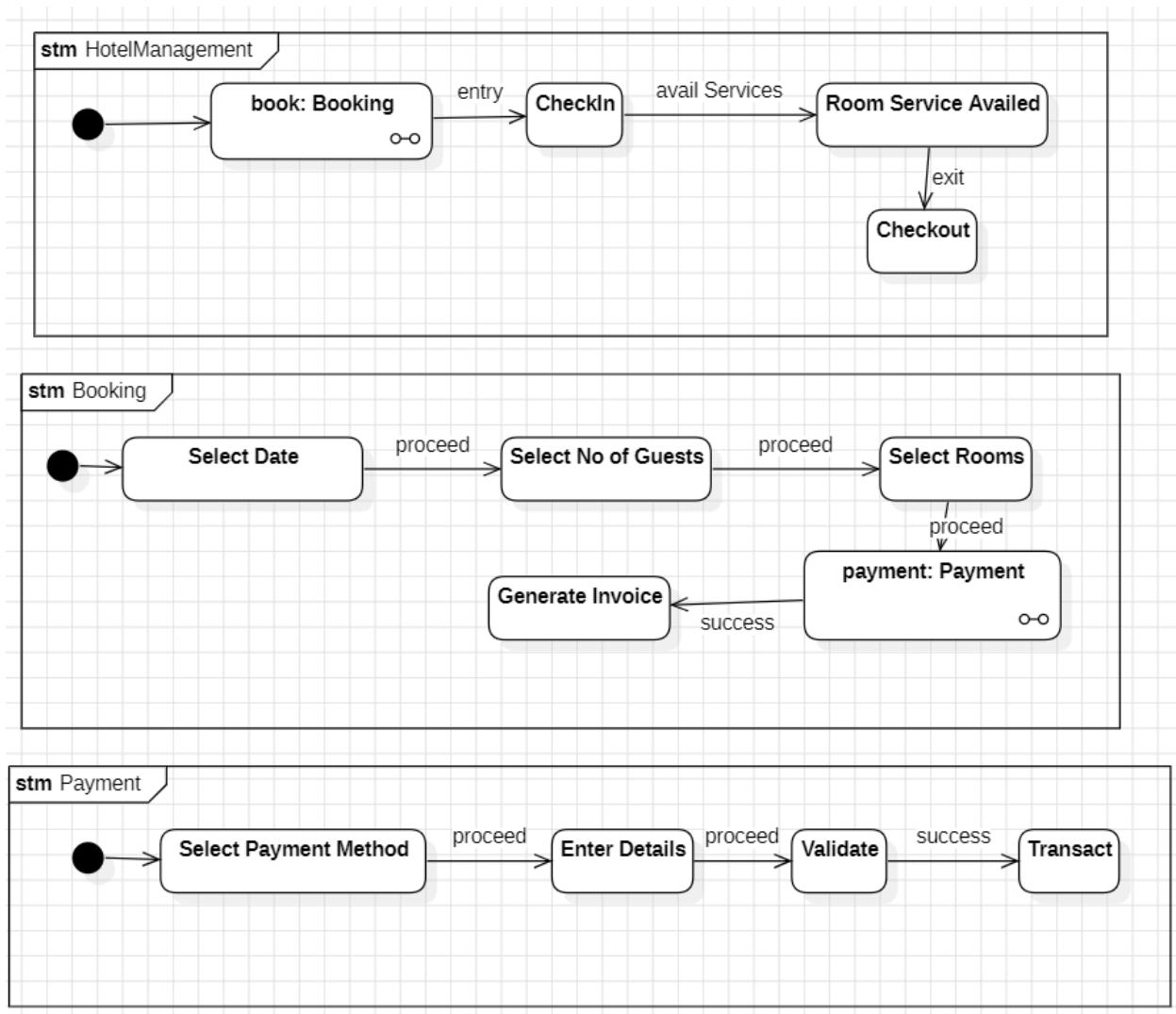


Figure 1.5

Description:

- There is a main Hotel Management state diagram and two submachine states called Booking and Payment.
- Hotel Management: Contains states such as Checkin - once the guest enters, Room Service Availed - if the guest opts for it and CheckOut on exit
- Booking: This submachine state has states to facilitate booking process i.e Select Date, Select no of guests, rooms, a submachine called payment and a state to generate invoice.
- Payment: This submachine has states for payment purpose such as Enter Details, Validate and Transact.

Use Case Diagram

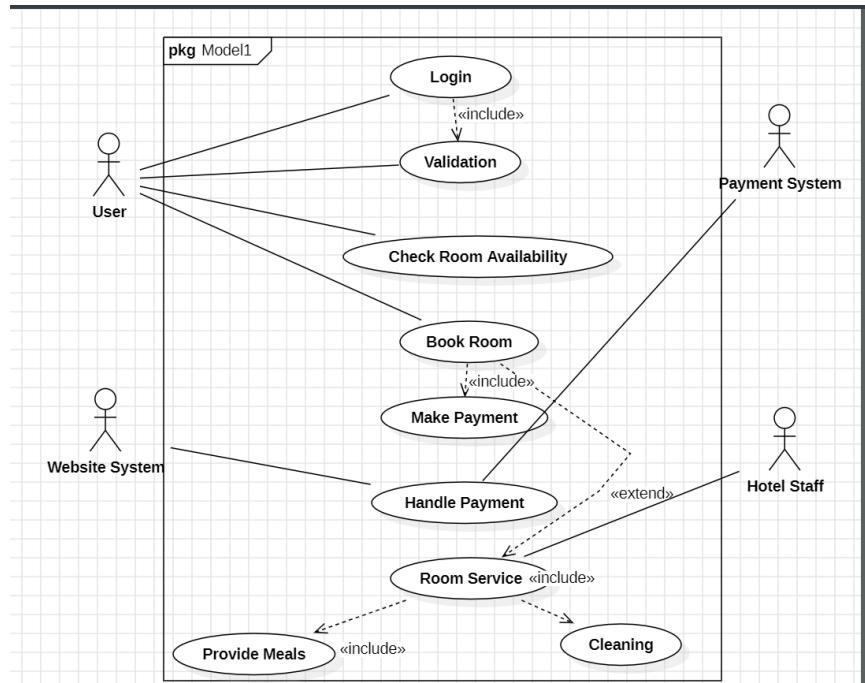


Figure 1.6

Description:

Actors involved: Hotel Staff, Website System, User, Payment System

User:

Login - The user can login to the website.

Check Room Availability - They can check if rooms are available.

Book room - They can make a booking. It includes making a payment and extends room service (choosing it is optional)

Payment System:

Handle payment - It handles payment.

Website System:

Validation - Validates the user from its database. (Error in diagram, line should go from website system to validation not user to validation)

Handle payment - Also handles payment process.

Hotel Staff:

Room Service - Provide room service to the guests. Includes providing meals and cleaning.

Sequence Diagram

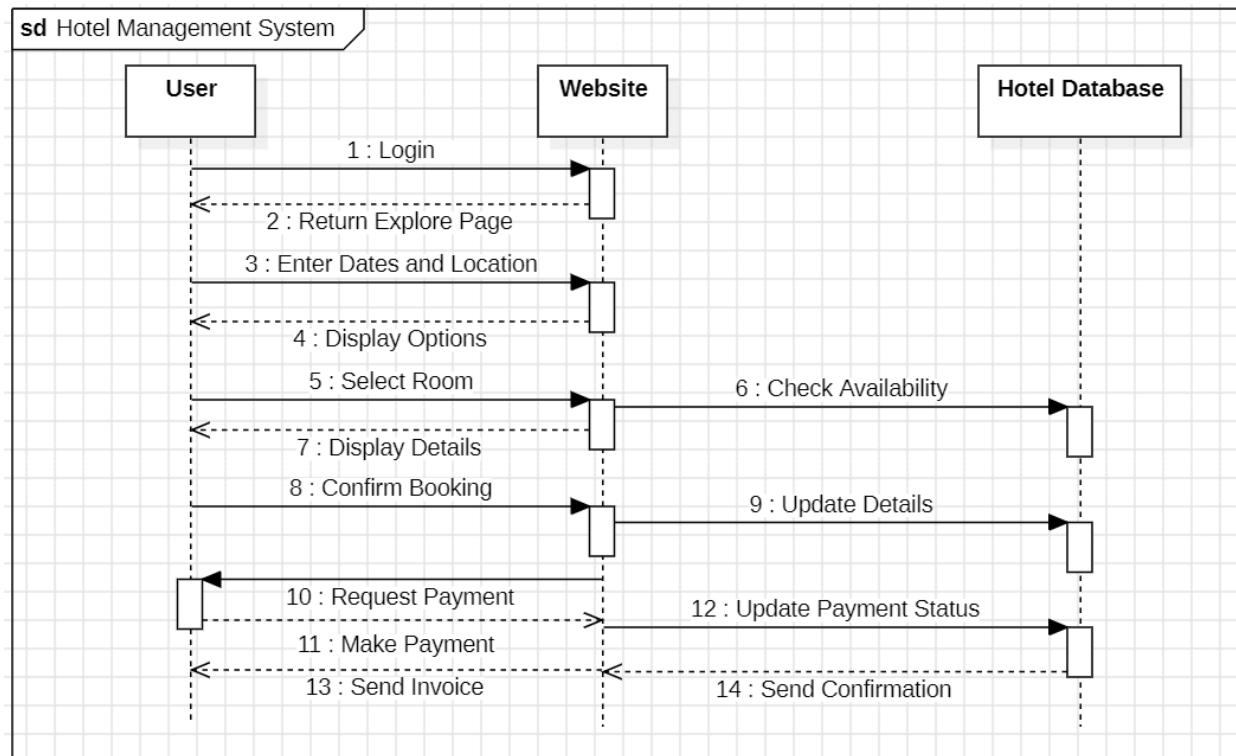


Figure 1.7

Description:

- The user logs in to the website. If the user is valid then the explore page is returned.
- The user enters the date and location. The website displays the available hotel options.
- Once the user selects the room of their choice the website enquires the hotel database about its availability.
- The website displays the details. The user confirms their booking, the user's details are updated in the hotel database.
- There is a payment request initiated by the website, the user makes the payment and this

is updated in the hotel database.

- On confirmation from hotel database an invoice is sent to the user.

Activity Diagram

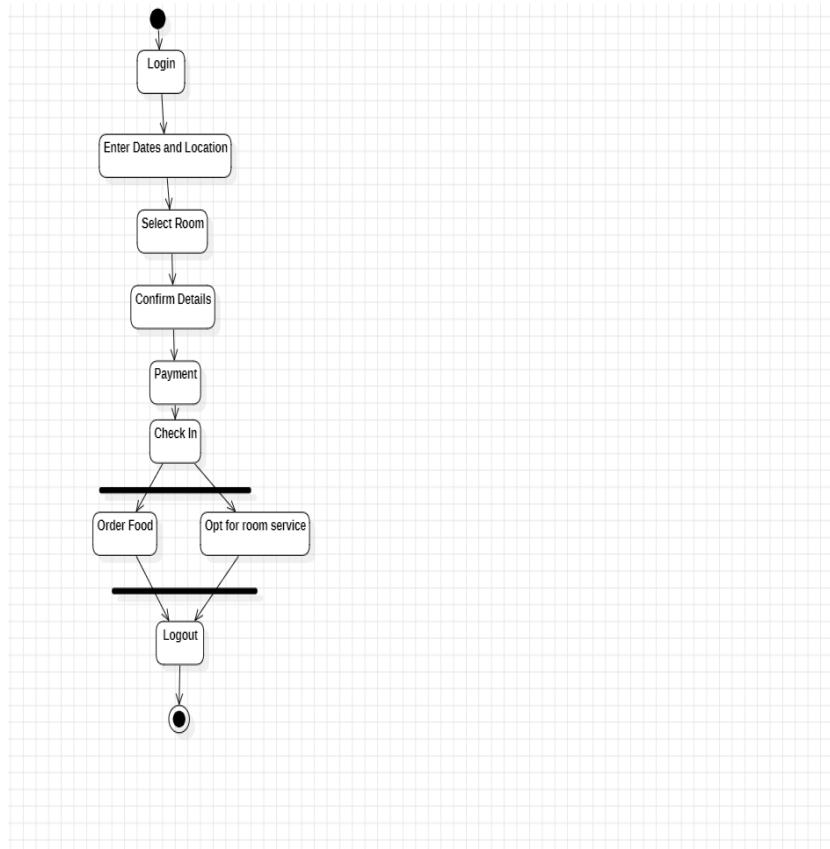


Figure 1.8

Description:

- The user logs in, enters the date and location in the website.
- Once the website displays the available options they select a room, confirm details and make payment.
- Post check in they may order food and opt for room service simultaneously.
- The user checks out. (Logout written wrongly in diagram)

2.Credit Card Processing System: SRS-Software Requirements Specification Document

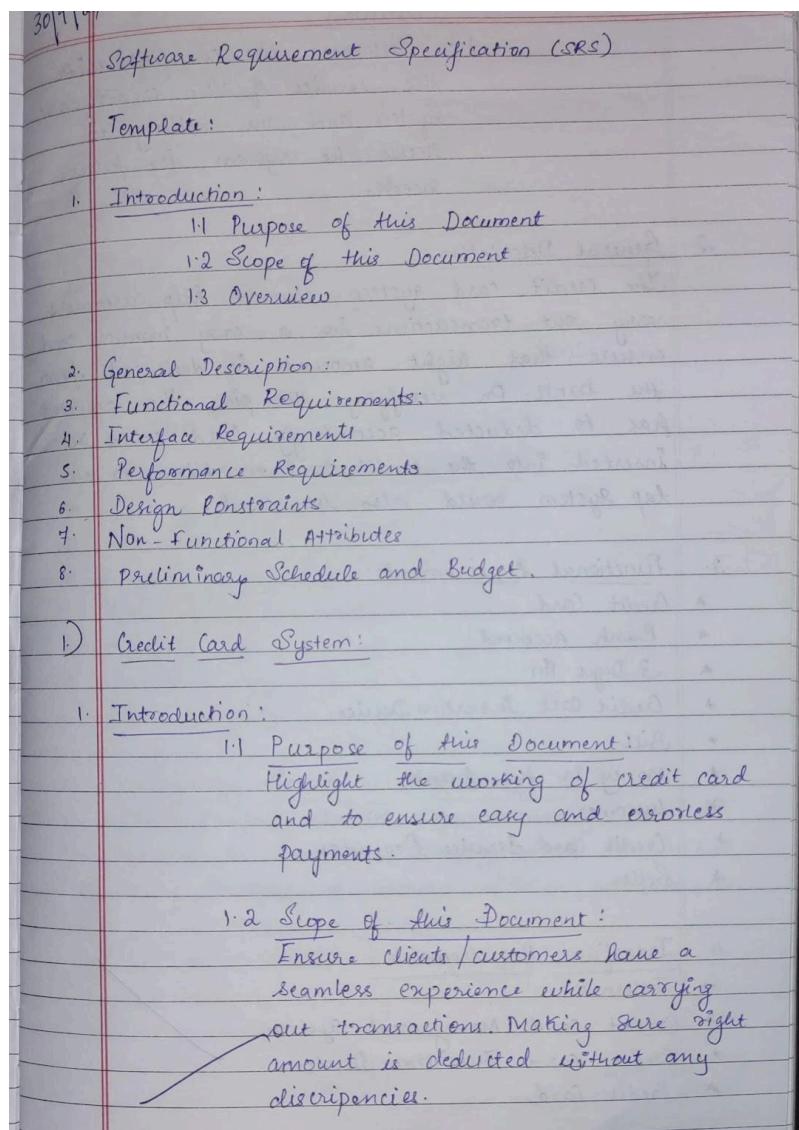


Figure 2.1

- Overview.
- 1.3 Customers are satisfied with the service of the credit card system and are willing to reuse the system for future needs.
2. General Description:
The credit card system should help customers carry out transactions in a easy manner and ensure that right amount is deducted from the bank. On verifying the pin, the amount has to deducted accordingly. Card could be inserted into the credit card machine or one tap system could also be used.
3. Functional Requirements:
- * Credit Card
 - * Bank Account
 - * 3 Digit Pin
 - * Credit Card Insertion Device.
 - * Bill
 - * Money in the bank
 - * Customer
 - * Credit card Service Provider
 - * Seller
4. Interface Requirements:
- * Internet
 - * Credit Card Management Systems
 - * Credit Card Insertion Device
 - * Credit Card

Figure 2.2

- Bafna Gold
Date: _____ Page: _____
5. Performance Requirements:
 - * Errorless Payments.
 - * Easy Transactions.
 - * Accurate amount deduction
 - * Fast Process Carry out
 6. Design Constraints:
 - * On tap functionality.
 - * No. - Internet Service
 - * Handling of pin.
 - * Contact-less transactions.
 7. Non-functional Requirements:
 - * Security for the pin
 - * Reliability
 - * Accuracy
 - * Precision
 - * Speed
 - * Portability
 - * Data Integrity
 8. Preliminary Schedule and Budget:
 1. The project should approximately take 6-months to complete, starting from the initial date. Budget of around \$1000 , per business.

Figure 2.3

Class Diagram

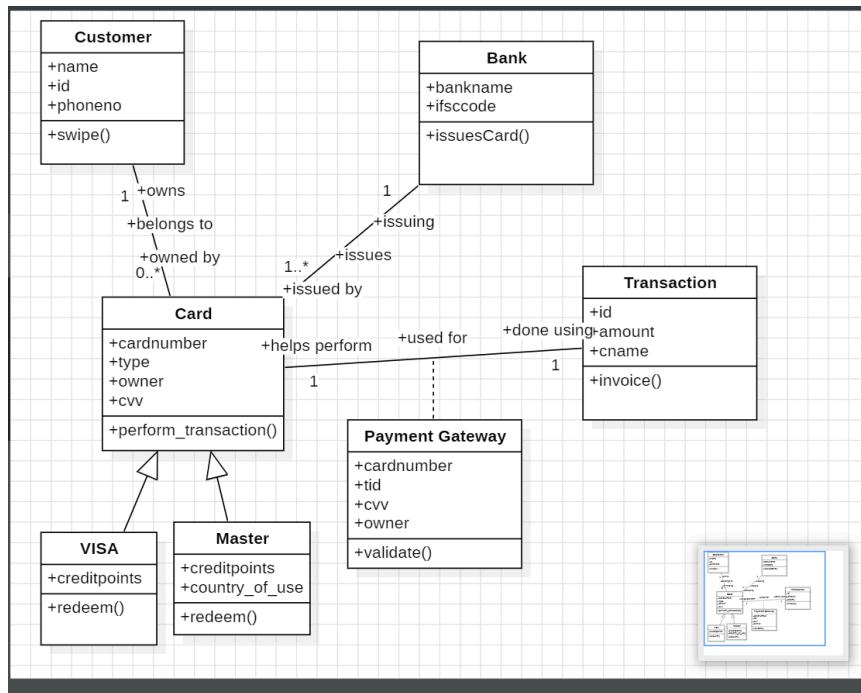


Figure 2.4

Description:

- Customer - Class indicative of customer using the card.
- Bank - Indicative of the bank of the credit card.
- Card - Indicative of the card used by the customer.
- Transaction - Indicative of the transaction that occurs on using credit card.
- Generalization - VISA, Master inherit properties from Card.
- Association Class - Payment Gateway is the association class associating Card and Transaction.

State Diagram

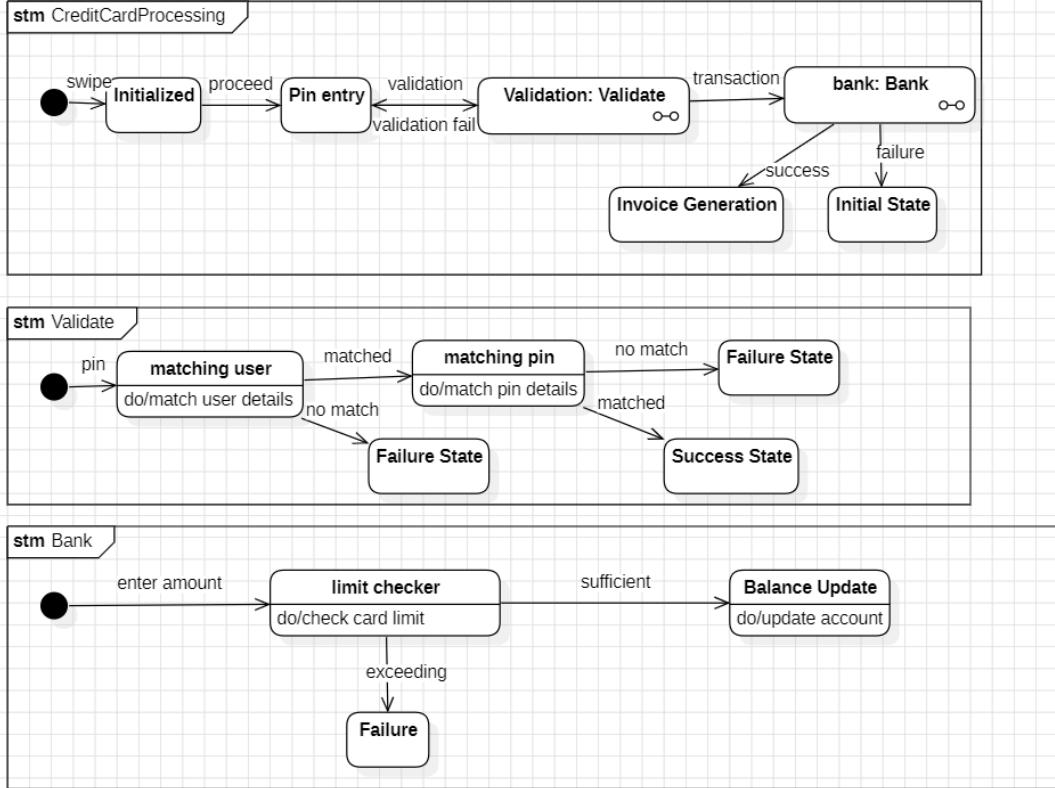


Figure 2.5

Description:

- The state diagram represents a CreditCardProcessing system with two submachines, Validate and Bank.
- The main system begins with the Initialized state (card swipe) and transitions to Pin Entry, followed by validation via the Validate submachine.
- If successful, it proceeds to the Bank submachine for transaction processing. The Validate submachine handles user and PIN matching, leading to success or failure.
- The Bank submachine checks transaction limits and updates balances, with transitions for

success or failure. Successful transactions lead to Invoice Generation, completing the process.

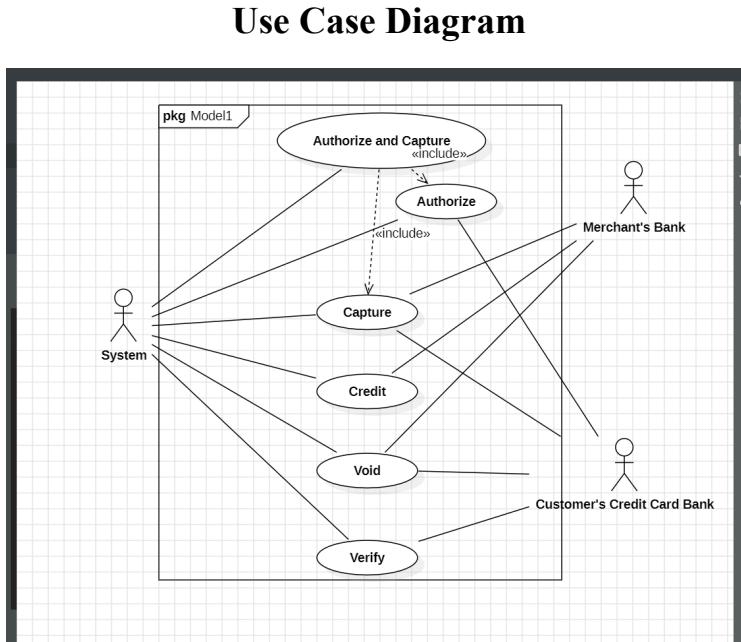


Figure 2.6

Description:

Actors Involved: System, Merchant's Bank, and Customer's Credit Card Bank.

System: Initiates and manages multiple use cases, including:

- Authorize and Capture: A composite process that includes Authorize and Capture operations.
- Authorize: Validates the payment with the Customer's Credit Card Bank.
- Capture: Completes the payment process by confirming the transaction with the Merchant's Bank.
- Credit: Handles refunding or crediting the customer's account.
- Void: Cancels an authorization or transaction.

- Verify: Verifies the cardholder's information.

Merchant's Bank: Processes authorization and captures funds from the transaction.

Sequence Diagram

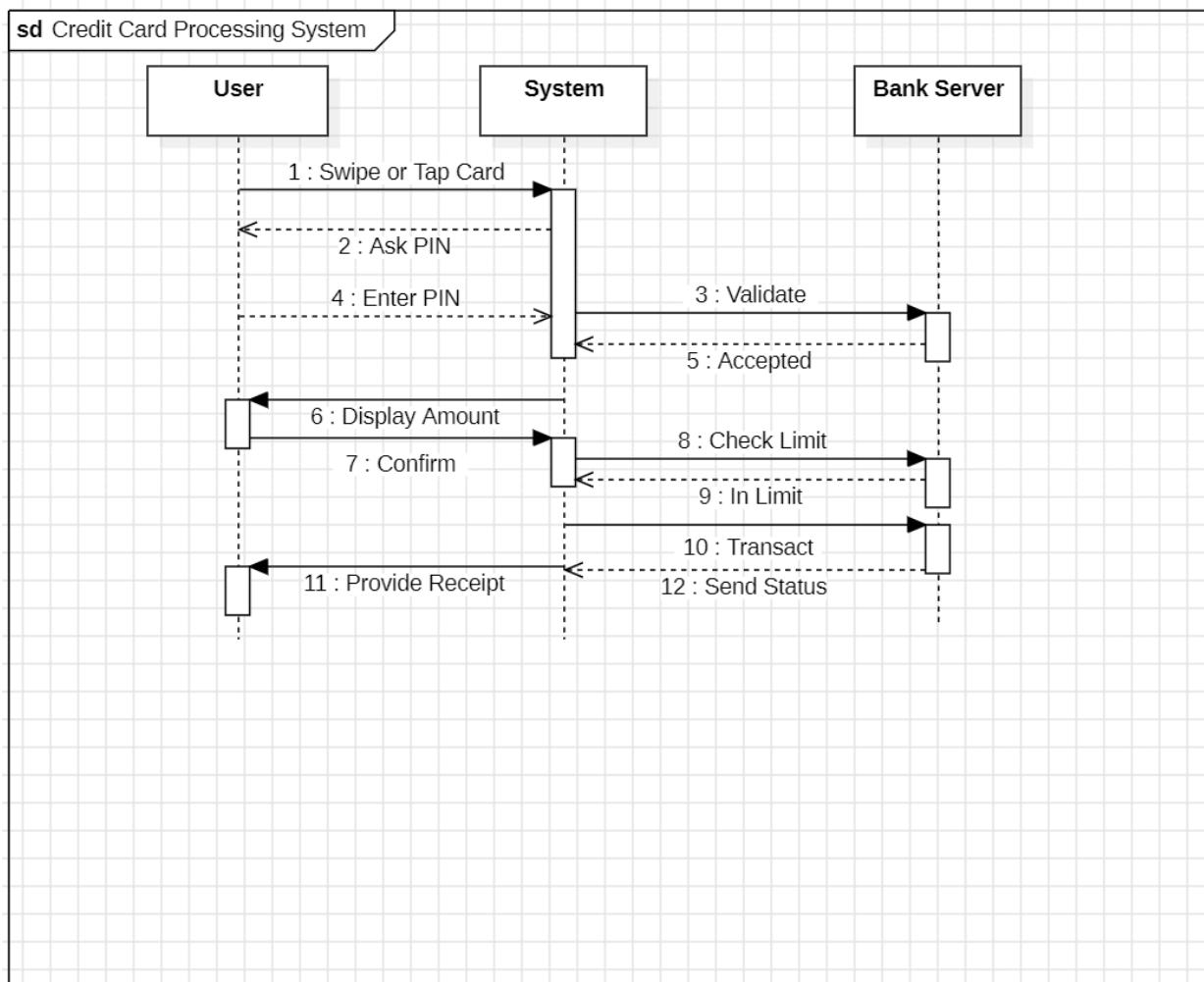


Figure 2.7

Description:

- The user swipes or taps their card.

- The system asks the user to enter the PIN.
- The user enters the PIN, and the system validates it with the bank server.
- The bank server responds with acceptance if the PIN is valid.
- The system displays the transaction amount to the user.
- The user confirms the amount.
- The system checks the credit limit with the bank server.
- If the limit is sufficient, the bank server processes the transaction.
- The transaction status is sent back to the system.
- The system provides a receipt to the user.

Activity Diagram

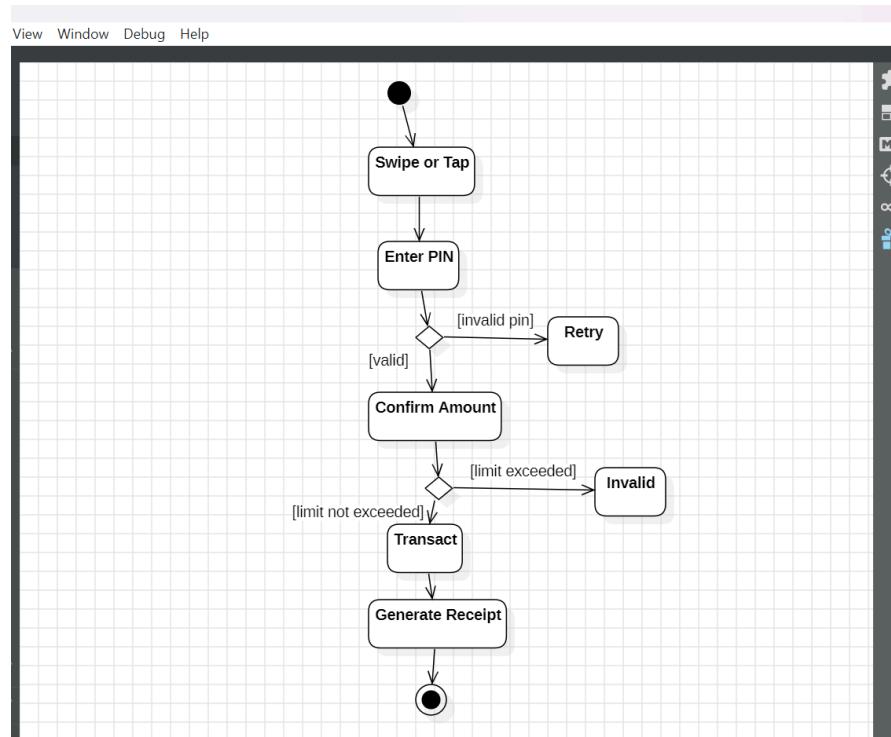


Figure 2.8

Description:

- The user swipes or taps the card.
- They enter the PIN. If the PIN is right the user is prompted to confirm if the amount displayed is right.
- Further if the limit has not been reached then the transaction happens and a receipt is generated else the transaction is invalid.

3. Library Management System

SRS-Software Requirements Specification Document

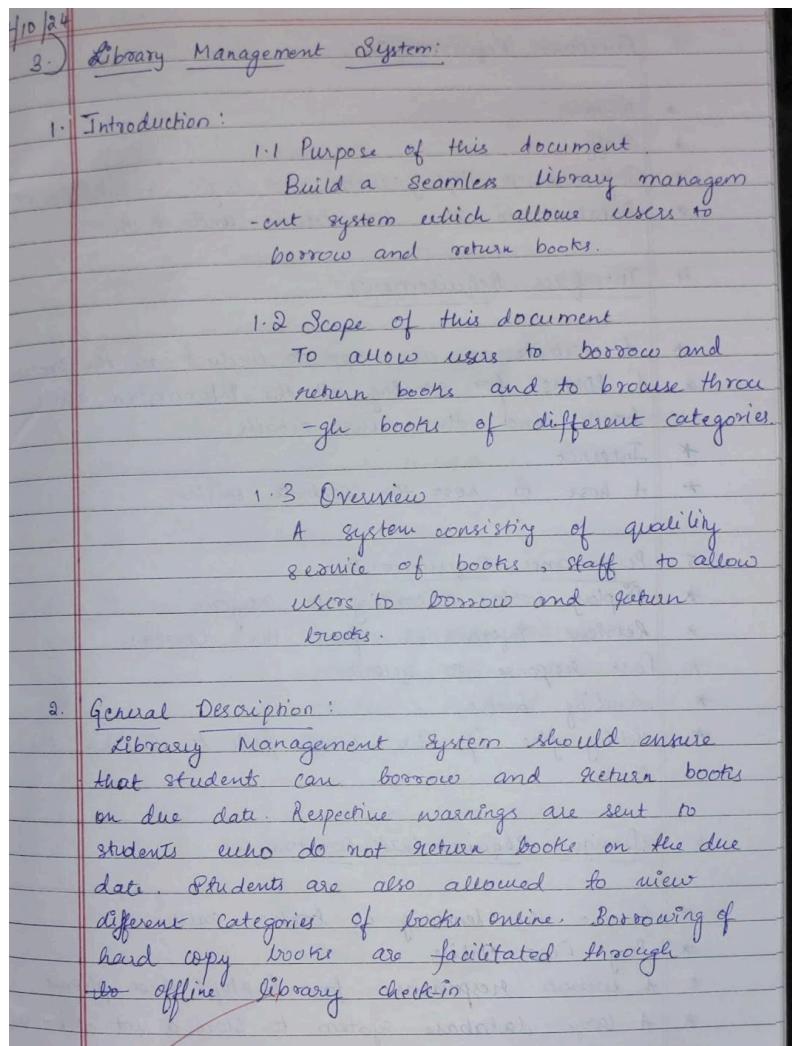


Figure 3.1

3. Functional Requirements:
- * Books
 - * Staff
 - * Software to display books and login credentials.
 - * Database to store user details, and books.
4. Interface Requirements:
- * A website or an app to conduct all the processes.
 - * A RDBMS for storing all the information about books and the user details.
 - * Internet
 - * A host to host the website online.
5. Performance Requirements:
- * Display books according to categories.
 - * Retrieve information from the database.
 - * Fast response to queries.
 - * Lending books.
 - * Setting fine if the user fails to return the book.
6. Design Requirements: Constraints:
- * Ensure the lending of books online.
 - * Pay fees online.
 - * Website responsive for laptops and phones.
 - * A large database system to store a lot of books.
 - * Provide audio forms of the books.

Figure 3.2

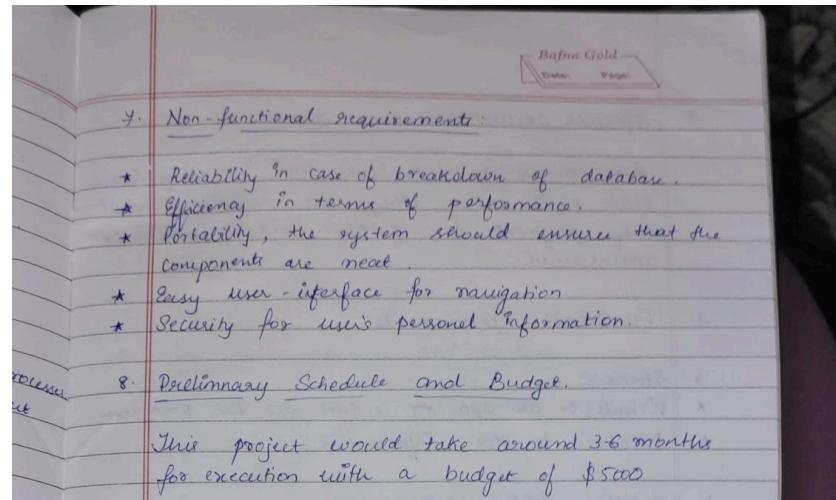


Figure 3.3

Class Diagram

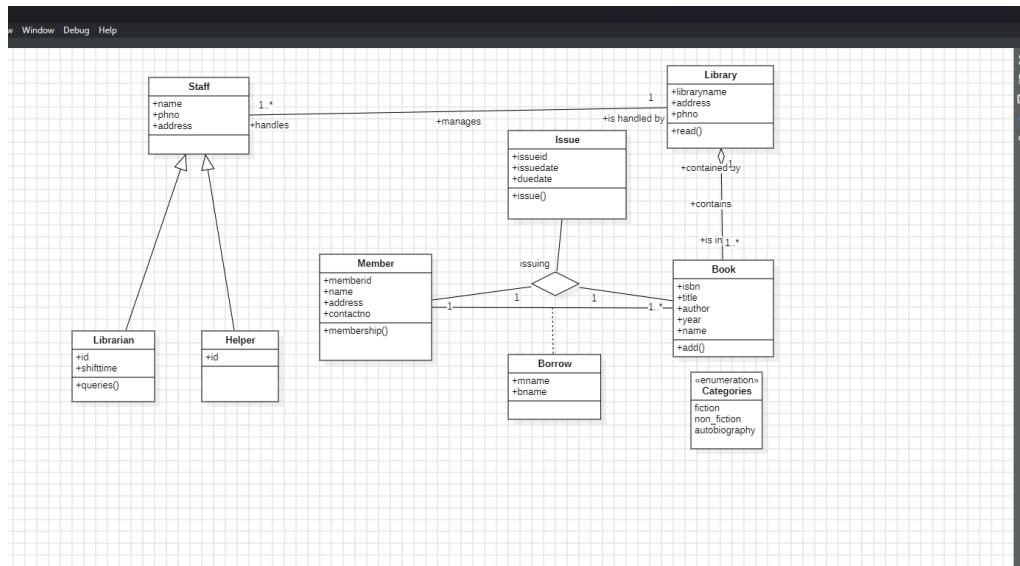


Figure 3.4

Description:

- Staff Class: Represents library staff with attributes such as name, phone number, and

address. It has generalization with subclasses Librarian (with shift time and queries functionality) and Helper, which inherit its properties.

- Library Class: Contains attributes like library name, address, and phone number. It is associated with the Staff class through aggregation, as the library is "handled by" one or more staff members.
- Book Class: Represents books with attributes such as ISBN, title, author, year, and category. It has a composition relationship with the Library class, as books are a part of the library and cannot exist independently of it.
- Member Class: Represents library members with attributes like member ID, name, address, and contact number. Members are associated with the Borrow class, which connects them to books.
- Issue Class: Manages the issuing of books with attributes like issue ID, issue date, and due date. It acts as an intermediary between Member and Book, forming an association.
- Borrow Class: Represents the act of borrowing, connecting Member and Book. It contains attributes for member name and book name.

State Diagram

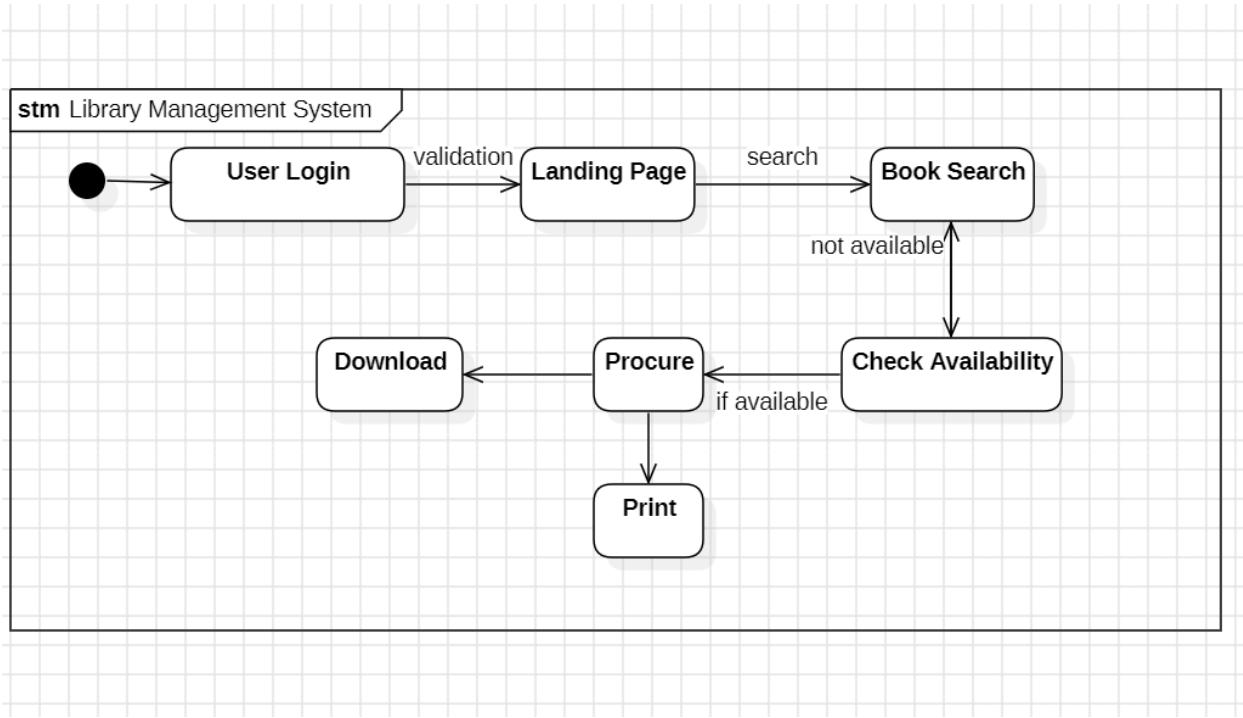


Figure 3.5

Description:

- The diagram represents a Library Management System with states for user interaction. The process starts at User Login, followed by validation and transition to the Landing Page.
- Users can search for books via the Book Search state, which leads to Check Availability. If the book is available, it transitions to Procure for obtaining the book, followed by options to Print or Download.
- If the book is not available, it loops back to Book Search for further searches.

Use Case Diagram

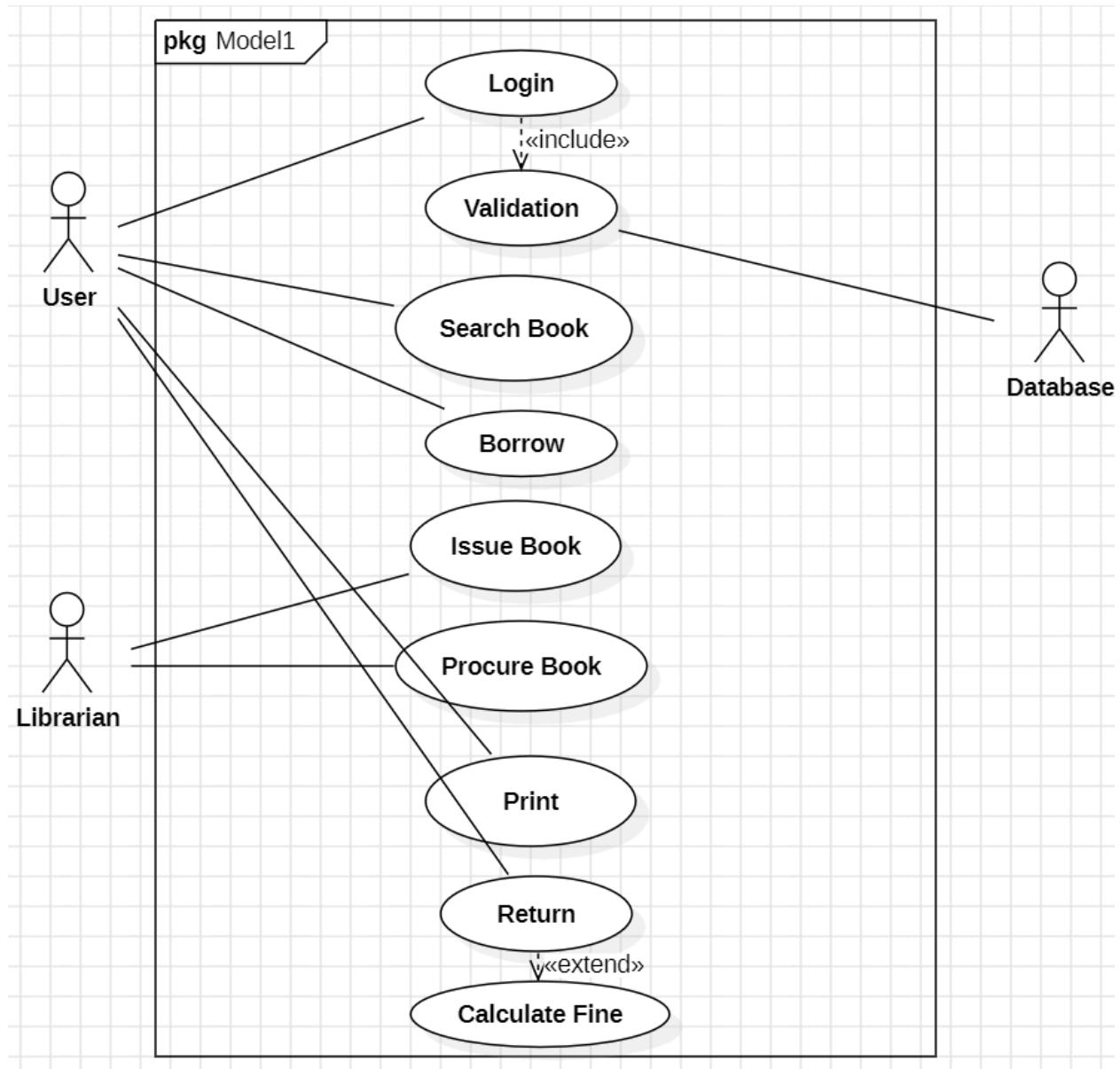


Figure 3.6

Description:

Actors Involved: User, Librarian, and Database.

User:

- Login: The user logs into the system, which includes Validation of their credentials against the Database.
- Search Book: Allows the user to search for books in the library's database.
- Borrow: Users can borrow books, which triggers the Issue Book use case managed by the Librarian.

- Return: Handles the process of returning borrowed books and extends to Calculate Fine in case of overdue returns.

Librarian:

- Issue Book: Oversees the issuance of books to users.
- Procure Book: Handles the procurement of new books for the library.
- Print: Prints reports or details as needed for administrative purposes.

Database: Supports all the system's use cases, such as storing user information, book details, and transaction records.

Sequence Diagram

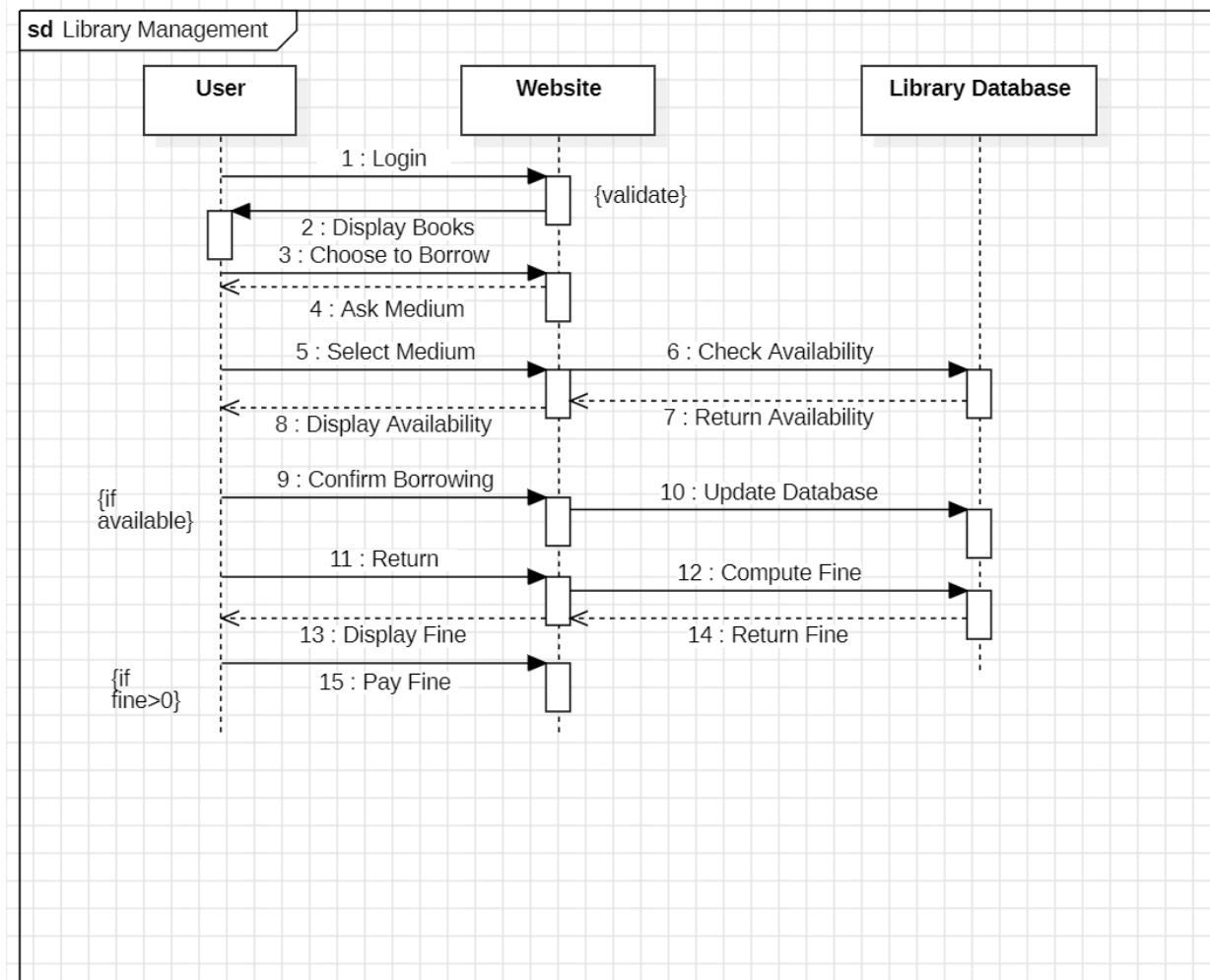


Figure 3.7

Description:

Login: The user logs into the library management system via the website, and the system validates the user credentials using the library database.

Display Books: After successful login, the website displays a list of books available for borrowing.

Choose to Borrow: The user selects a book they wish to borrow.

Ask Medium: The website prompts the user to select the borrowing medium (e.g., physical or

digital).

Select Medium: The user selects their preferred medium for borrowing.

Check Availability: The website checks the book's availability by querying the library database.

Return Availability: The library database returns the availability status of the selected book.

Display Availability: The website displays the availability information to the user.

Confirm Borrowing: If the book is available, the user confirms their intention to borrow it.

Update Database: The website updates the library database to reflect the borrowed status of the book.

Return: When the user is done with the book, they return it through the website.

Compute Fine: The website requests the library database to compute any late return fine.

Display Fine: If a fine exists, the website displays the fine amount to the user.

Pay Fine: If the fine is greater than zero, the user proceeds to pay it via the website.

Return Fine: The payment status is updated in the library database.

Activity Diagram

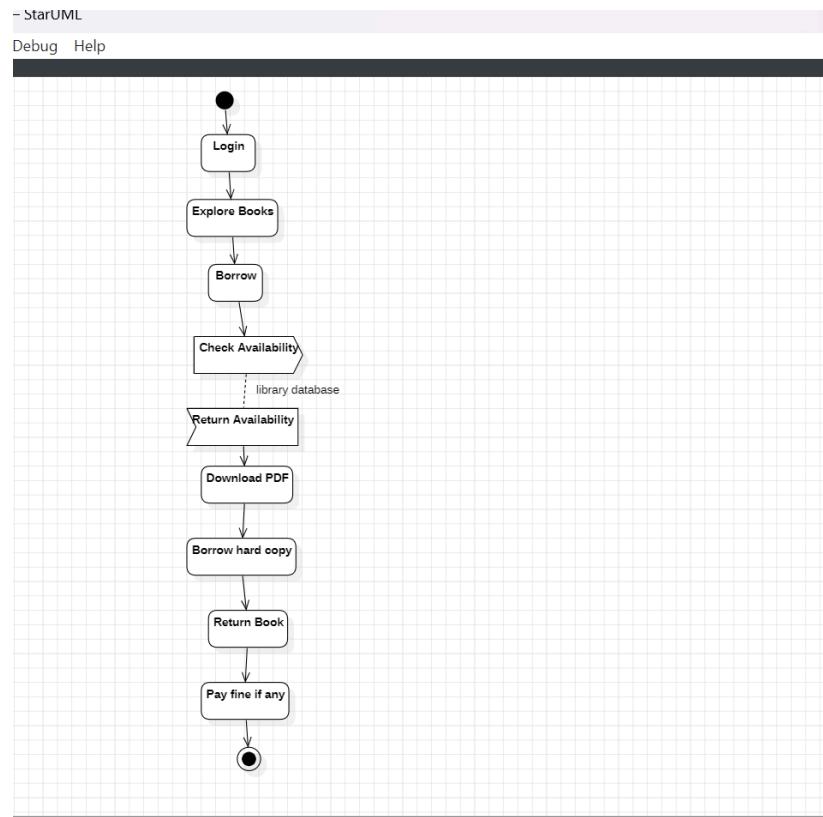


Figure 3.8

Description:

- The user logs in and explores the books.
- If they chose to borrow the availability is checked in the database and returned.
- If the book is available then the user may download a pdf and then borrow hardcopy.
- Once the user returns the book they may pay the fine if any.

4. Stock Maintenance System

SRS-Software Requirements Specification Document

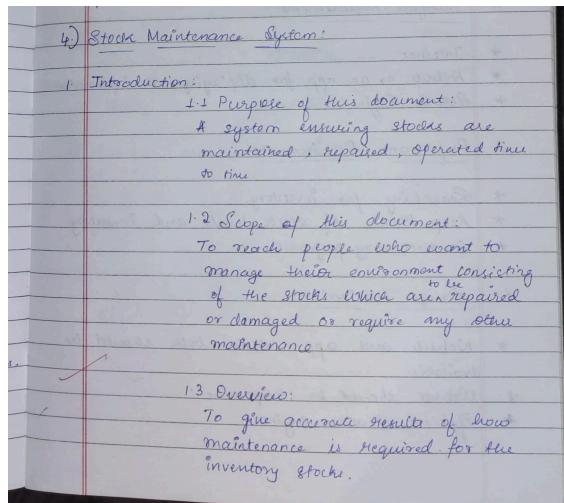


Figure 4.1

2. General Description:
keep track of stock levels and inventory. Locate
spare parts when they are missing. Conduct
easy payments for getting stocks repaired or
maintained.
3. Functional Requirements:
- * Stocks
 - * Website or an app to conduct all the processes
 - * Database to store information.
4. Interface Requirements:
- * Internet
 - * Website or an app for displaying all information.
 - * Search Engine.
5. Performance Requirements:
- * Searching for inventory.
 - * Keep track of stock levels and inventory.
 - * Conduct easy payments.
6. Design Constraints:
- * Website and app version both should be available.
 - * Website should be responsive.
 - * Efficient search engines.

Figure 4.2

Bajna Gold																				
	Date:	Page:																		
<u>7. Non-functional Requirements:</u>																				
<ul style="list-style-type: none"> * Security. * Reliability. * Portability. * Efficiency. * Easy graphical interface. 																				
<u>8. Preliminary Schedule and Budget:</u>																				
<table> <thead> <tr> <th></th> <th>Duration</th> <th>Cost</th> </tr> </thead> <tbody> <tr> <td>Requirements Gathering</td> <td>1 month</td> <td>\$50.</td> </tr> <tr> <td>Design</td> <td>1 month</td> <td>\$85</td> </tr> <tr> <td>Development</td> <td>2 months</td> <td>\$50</td> </tr> <tr> <td>Testing</td> <td>1 month</td> <td>\$10.</td> </tr> <tr> <td>Total:</td> <td>5 months</td> <td>\$135.</td> </tr> </tbody> </table>				Duration	Cost	Requirements Gathering	1 month	\$50.	Design	1 month	\$85	Development	2 months	\$50	Testing	1 month	\$10.	Total:	5 months	\$135.
	Duration	Cost																		
Requirements Gathering	1 month	\$50.																		
Design	1 month	\$85																		
Development	2 months	\$50																		
Testing	1 month	\$10.																		
Total:	5 months	\$135.																		
<p style="text-align: center;">8/10 Done</p>																				

Figure 4.3

Class Diagram

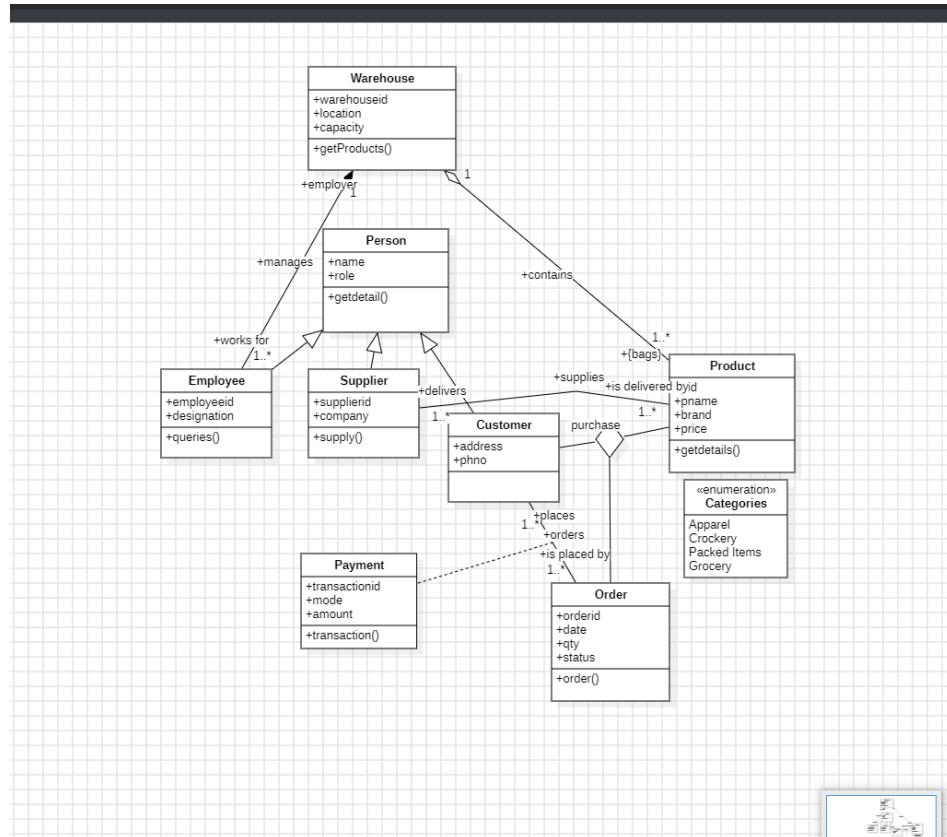


Figure 4.4

Description of Class Diagram:

- **Warehouse**: Represents storage facilities containing products (composition with Product).
- **Person**: General class (generalization) inherited by Employee, Supplier, and Customer.
- **Employee**: Represents staff working for the warehouse (aggregation with Warehouse).
- **Supplier**: Represents entities delivering products to the warehouse (association with Product).
- **Customer**: Represents buyers placing orders for products (association with Order).
- **Product**: Represents items stored in the warehouse, categorized by type (composition with Warehouse).

- Order: Represents customer purchases, linked to Product and Customer (aggregation with Customer).
- Payment: Captures transaction details for orders (association with Order).

State Diagram

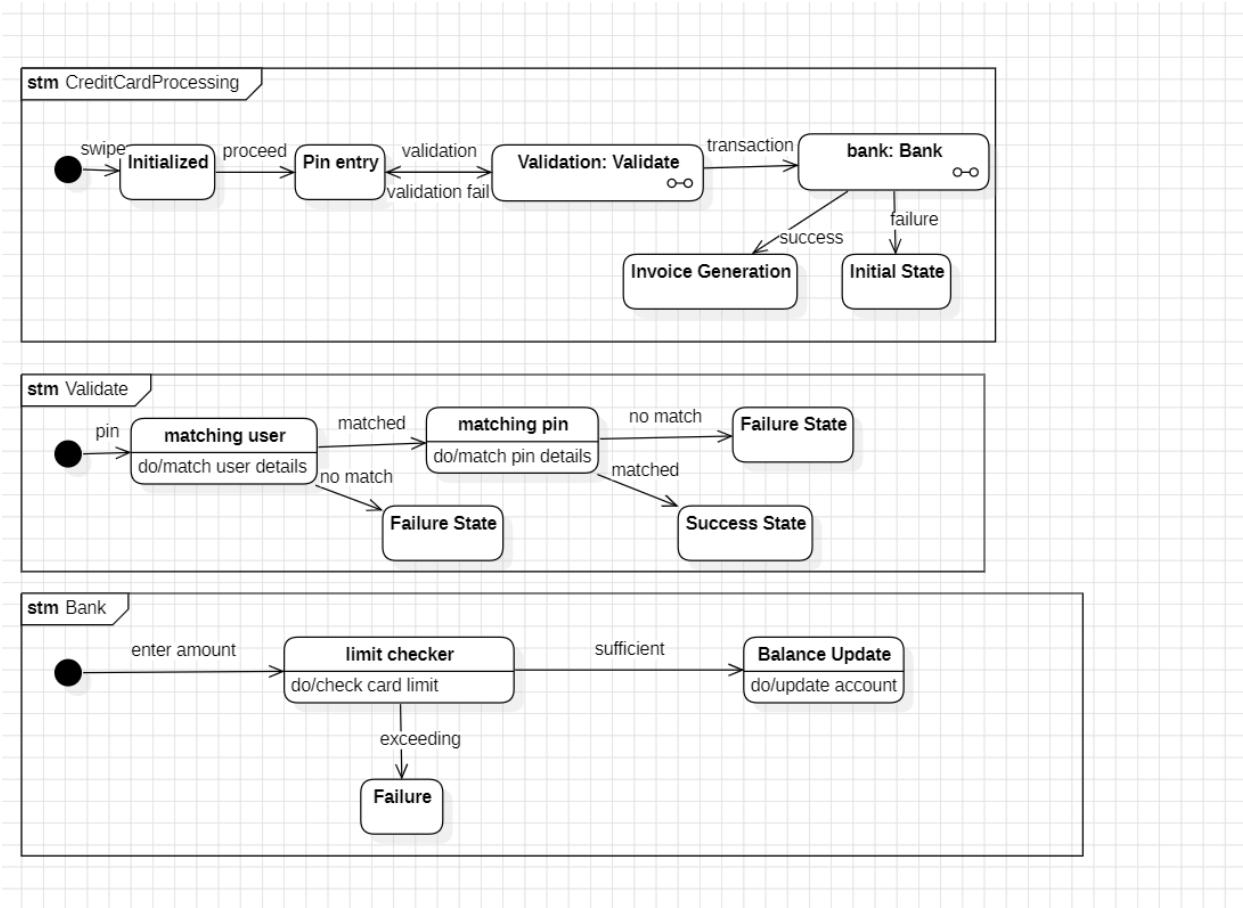


Figure 4.5

Description of State Diagram:

- The state diagram represents a Stock Maintenance system with states for managing inventory and deliveries.
- The system starts in the Locked state, operational only between 7 AM and 11 PM,

transitioning to Unlocked.

- In the Unlocked state, stock is checked in Item Count, and items are updated if necessary. Processes include Contacting Supplier, Placing Order, and a Payment submachine for transaction handling.
- Upon success, it proceeds to Fetching Order Details, Updating Warehouse, and Receiving Payment, finally leading to Schedule Delivery.
- The Payment submachine includes states for selecting a mode, entering details, validating, and completing the transaction.

Use Case

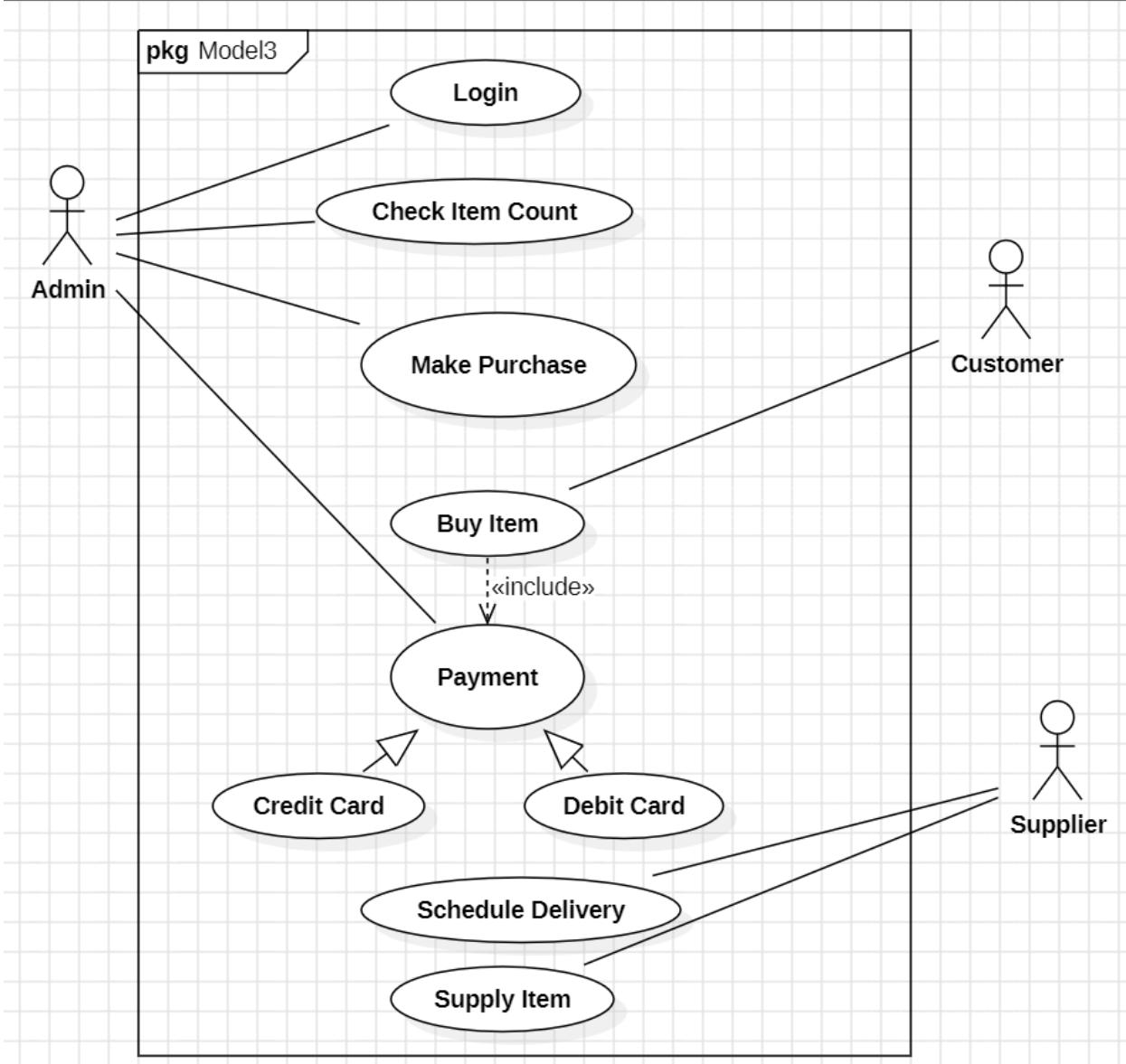


Figure 4.6

Description of Use Case Diagram:

Actors Involved: Admin, Customer, and Supplier.

Admin: Logs into the system, checks item count, and makes purchases by buying items.

Includes: The Buy Item use case includes the Payment process, which involves selecting either

credit or debit card as payment methods.

Supplier: Supplies items after the admin schedules delivery.

Extends: Delivery scheduling extends the purchase process to ensure timely item supply.

Sequence Diagram

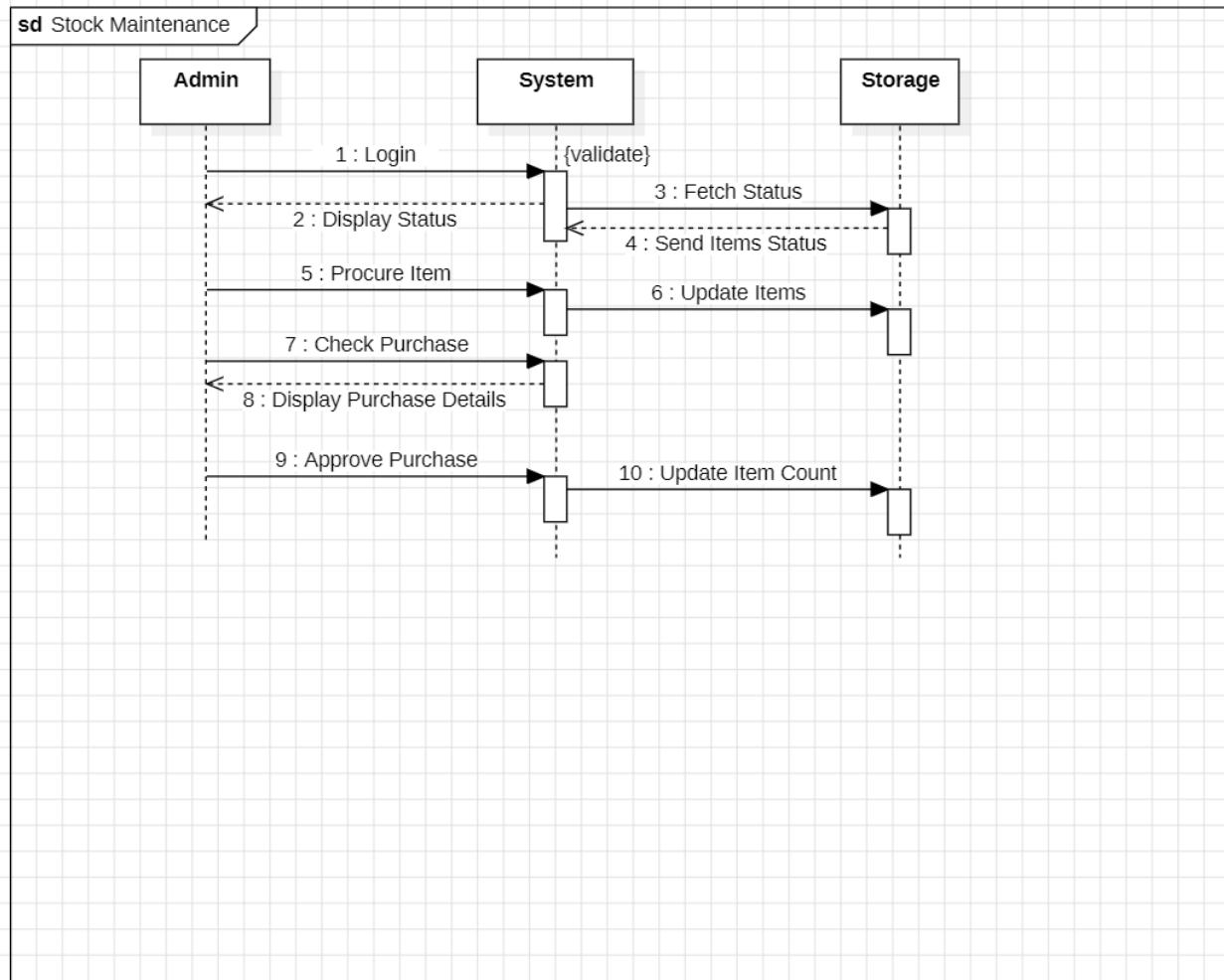


Figure 4.7

Description:

Login: The admin logs into the stock maintenance system through the interface, and the system validates the admin's credentials.

Display Status: The system displays the current stock status to the admin.

Fetch Status: The system retrieves the stock status by querying the storage system.

Send Items Status: The storage system sends the current items' status back to the system.

Procure Item: The admin initiates the process to procure an item based on the displayed stock

status.

Update Items: The system updates the stock items in the storage after procurement.

Check Purchase: The admin checks the purchase details to verify the procurement process.

Display Purchase Details: The system displays the details of the purchase to the admin.

Approve Purchase: If satisfied with the details, the admin approves the purchase.

Update Item Count: The system updates the item count in the storage system to reflect the approved purchase.

Activity Diagram

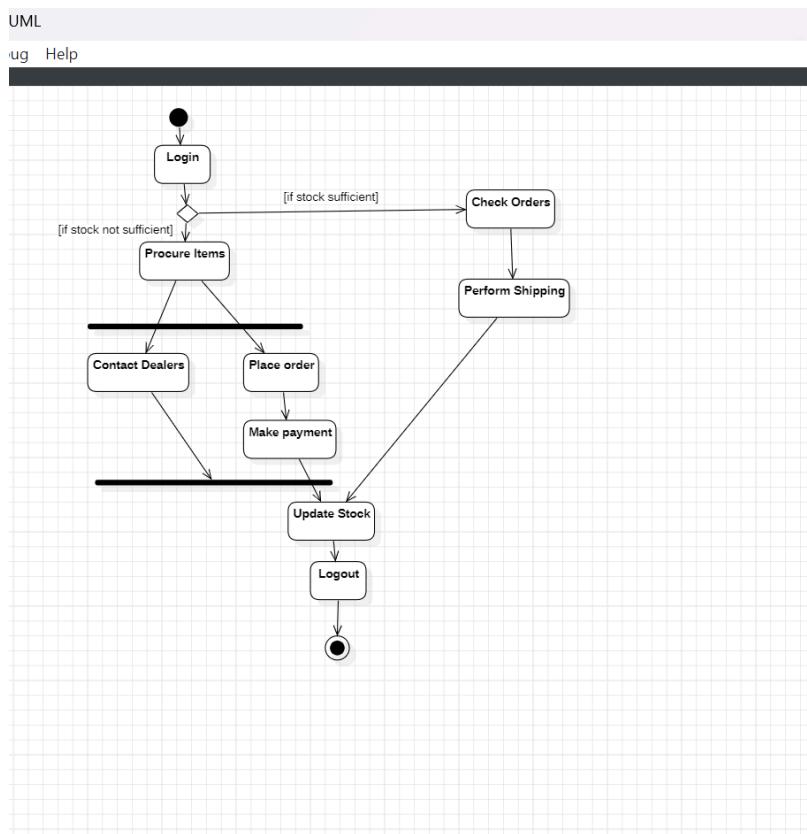


Figure 4.8

Description:

- The warehouse manager logs into the system.
- They check if there is sufficient stock. If there is, they proceed to check the orders and deliver those items.
- Otherwise they place the order by contacting the dealers and procuring the items by making payments.

5. Passport Automation System

SRS-Software Requirements Specification Document

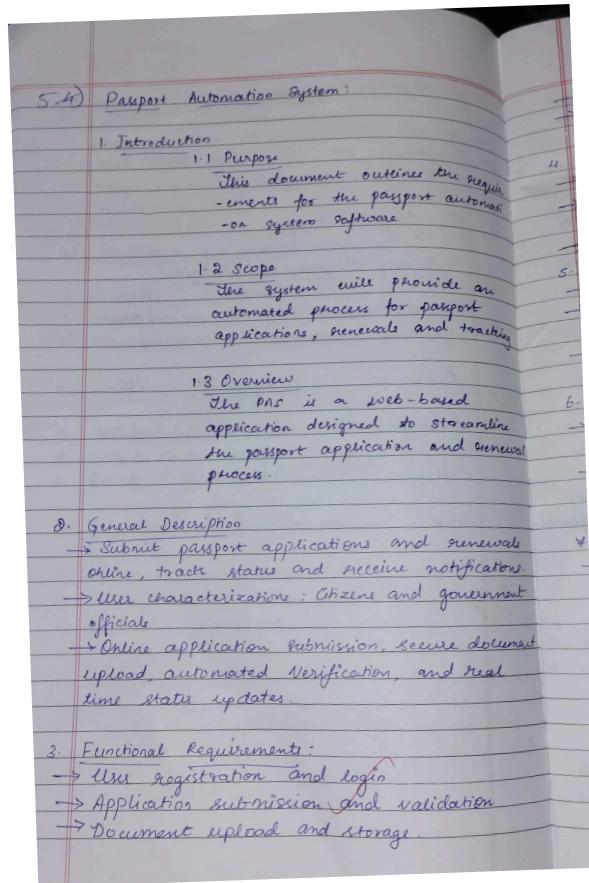


Figure 5.1

- Bafna Gold
Date: _____ Page: _____
- Application tracking and status updates.
 - Verification and processing.
4. Interface Requirements:
- User friendly interface.
 - Integration with existing government databases and systems.
5. Performance Requirements:
- Response time: 2 seconds.
 - Data storage and retrieval: efficient and secure.
 - Error rate: less than 1%.
6. Design Constraints:
- Security: ensure data integrity and comply with regulations.
 - Hardware and software limitations: compatible with existing infrastructure.
7. Non-functional attributes:
- Security: ensure data integrity and comply with regulations.
 - Scalability: accommodate increasing user traffic and data storage needs.
 - Reliability: available 24/7 with backups & disaster recover plan.
8. Preliminary Schedule & Budget:

Figure 5.2

Requirements Gathering	Duration 3 months	Budget \$ 100,000
Design	3 month	\$ 100,000
Development	3 months	\$ 100,000
Testing	3 months	\$ 100,000
Total Duration : 12 months		
Total Budget : \$ 400,000.		

Figure 5.3

Class Diagram

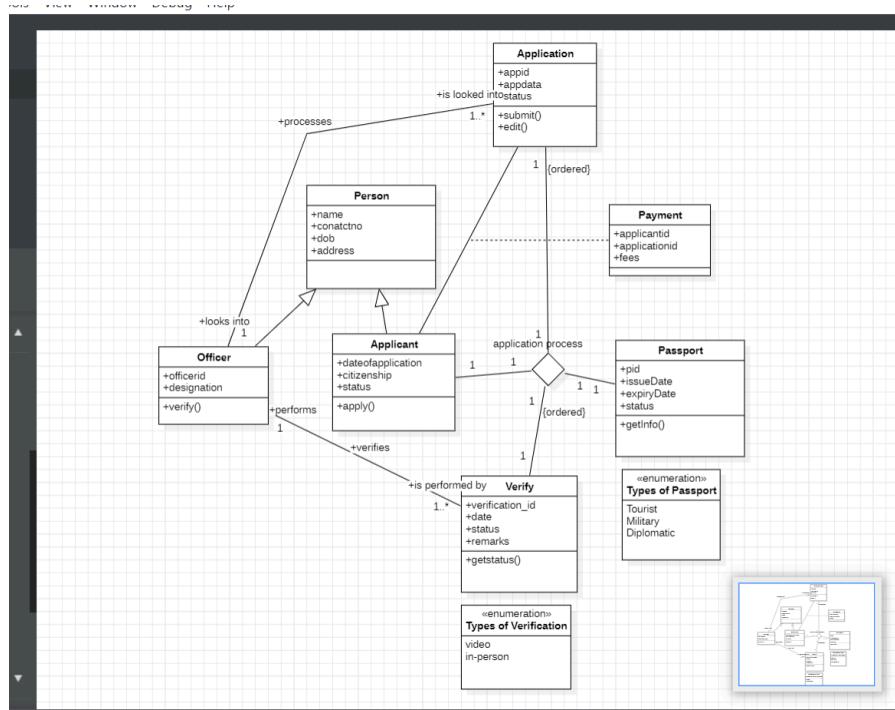


Figure 5.4

Description of Class Diagram

- Person is a general class associated with Applicant and Officer, representing entities involved in the system.
- Applicant applies for a passport, links to Application, and has attributes like citizenship and application status.
- Officer verifies applications, performing a verification process represented by the Verify class, which details the verification type (in-person or video).
- Application is associated with Payment for fees and leads to the generation of a Passport, categorized into different types (Tourist, Military, Diplomatic).
- The Passport includes essential attributes like issue and expiry dates, completing the process flow.

State Diagram

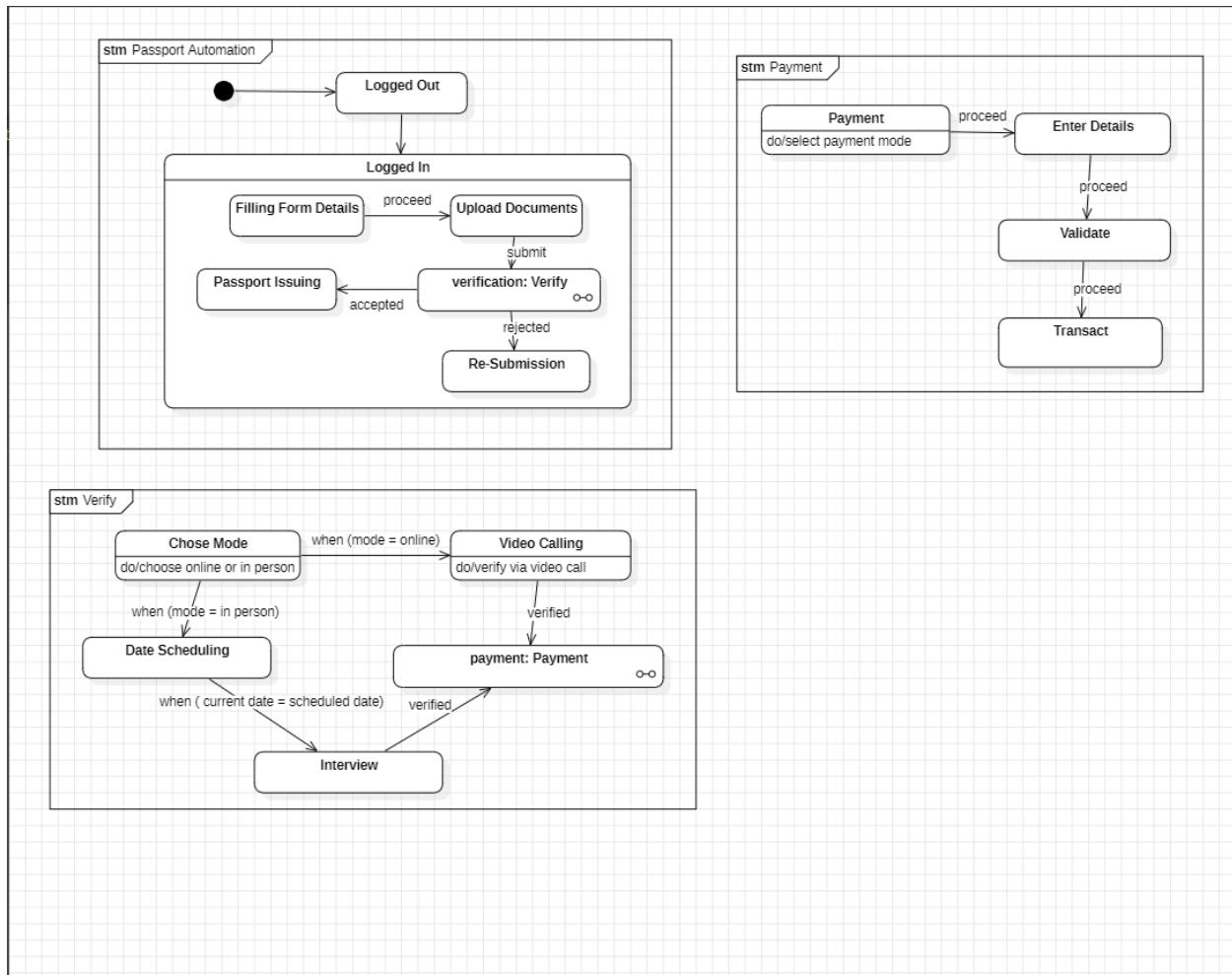


Figure 5.5

Description of State Diagram:

- This state diagram represents a Passport Automation system. It begins in the Logged Out state, transitioning to Logged In for processes such as Filling Form Details, Uploading Documents, and Passport Issuing.
- Verification involves choosing a mode (online or in-person) with subsequent video calling or scheduled interviews. Rejected applications allow re-submission.
- The Payment submachine handles mode selection, detail entry, validation, and transaction completion.

Use Case Diagram

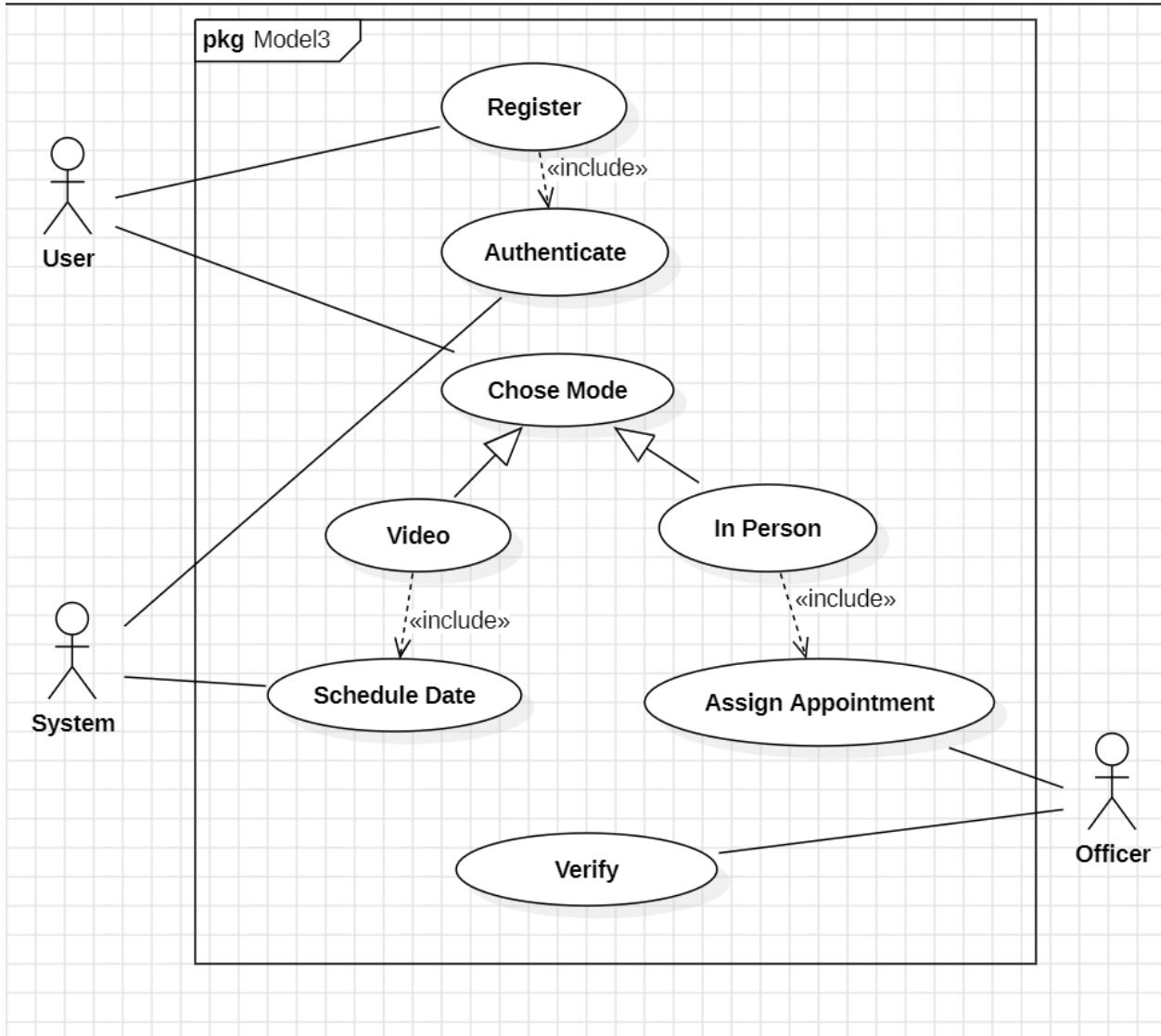


Figure 5.6

Description of Use Case Diagram:

Actors Involved: User, System, and Officer.

User: Registers and authenticates to access the system. They choose a mode—Video or In-Person.

System: Schedules dates for video mode or assigns appointments for in-person mode.

Includes: The Authenticate use case is included in Register, and Schedule Date and Assign Appointment include their respective mode-specific processes.

Extends: Verification extends to both appointment modes, ensuring successful completion by the Officer.

Sequence Diagram

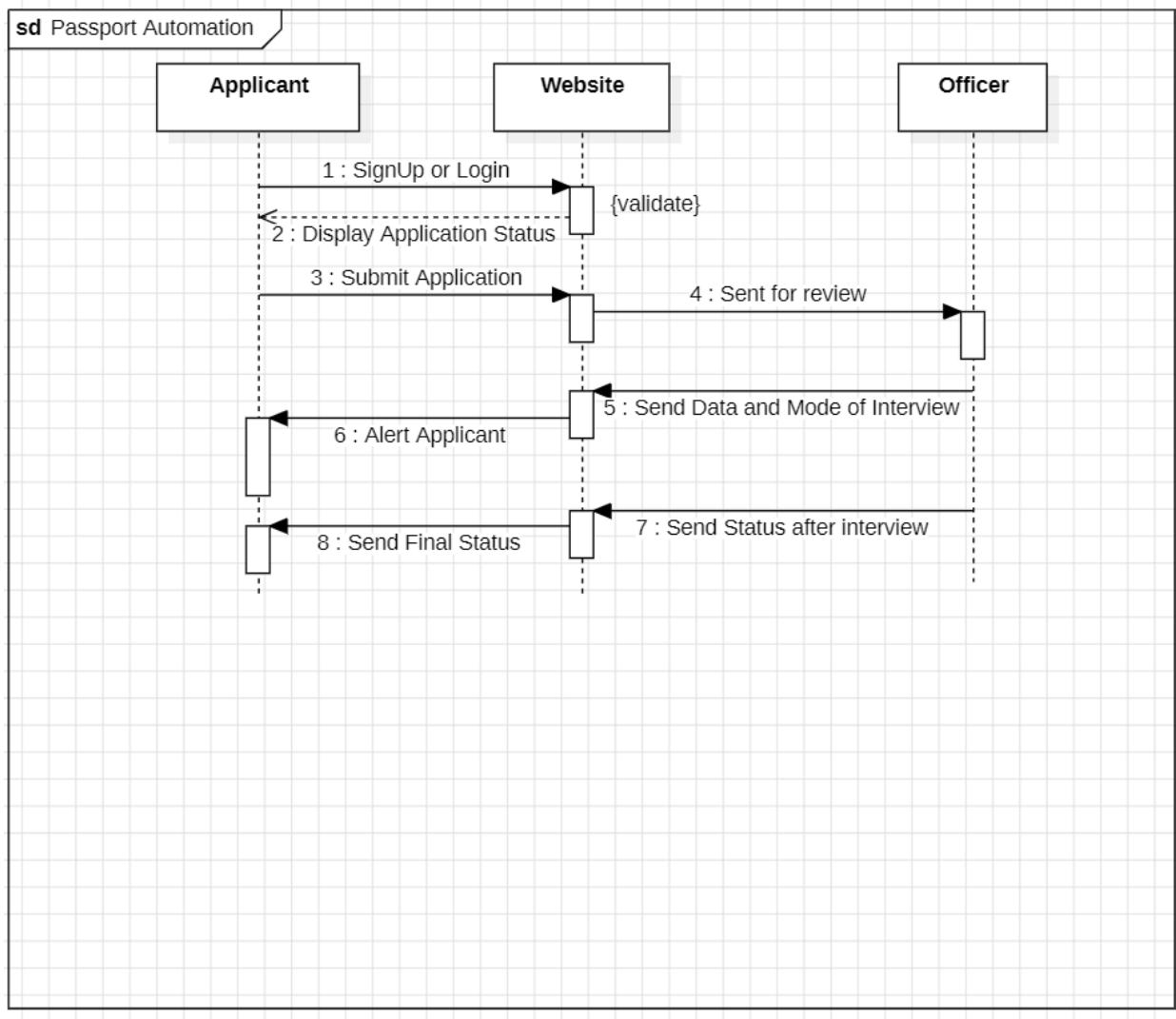


Figure 5.7

Description:

SignUp or Login: The applicant signs up or logs into the passport automation system through the website, and their credentials are validated.

Display Application Status: The website displays the current status of the applicant's passport application.

Submit Application: The applicant submits their passport application via the website.

Sent for Review: The website forwards the submitted application data to the officer for review.

Send Data and Mode of Interview: The officer sends back the reviewed data along with the mode and schedule of the interview.

Alert Applicant: The website notifies the applicant about the interview details or the next steps.

Send Status after Interview: After the interview, the officer updates the application status, which is sent back to the website.

Send Final Status: The website communicates the final status of the application to the applicant.

Activity Diagram

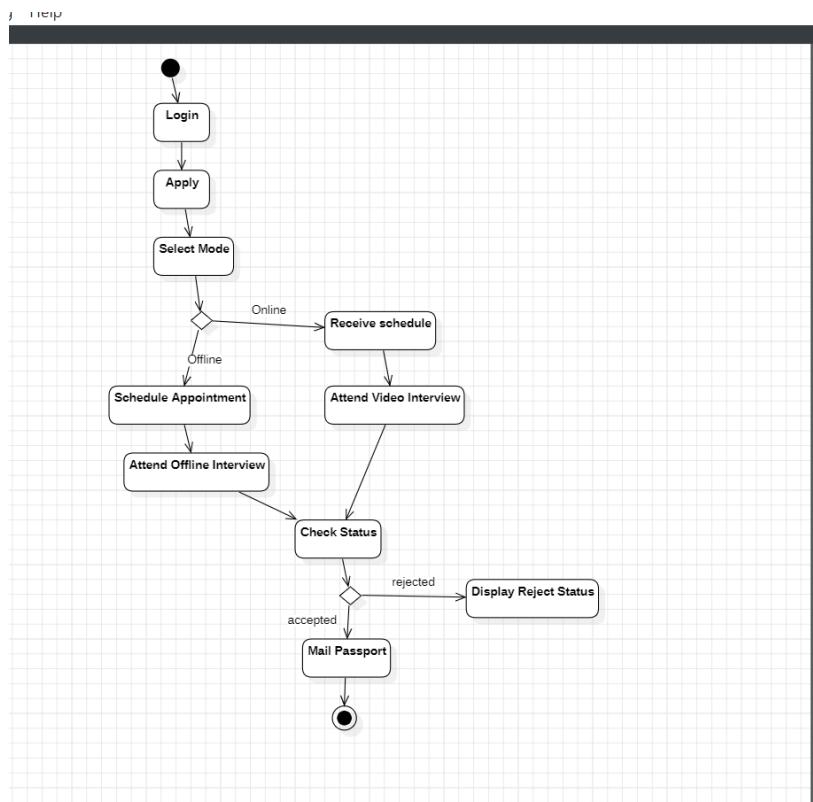


Figure 5.8

Description:

- The user logs in to the application. They apply for the passport.
- They select the mode i.e online or offline.
- If online they receive a schedule and attend the interview online.
- If offline they schedule an appointment and attend the interview offline.
- Post this they check the status of their application on the platform. If accepted the passport is mailed to them.