

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



Lab Record

**Object-Oriented Modeling**

*Submitted in partial fulfillment for the 5<sup>th</sup> Semester Laboratory*

Bachelor of Engineering  
in  
Computer Science and Engineering

*Submitted by:*

**SNEHA N SHASTRI**

(1BM22CS283)

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 N SHASTRI (1BM22CS283) during the 5<sup>th</sup> Semester Oct 2024 - Jan2025.

Signature of the Faculty Incharge:

NAME OF THE FACULTY: Prof Prameetha Pai

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

## Table of Contents

1. Hotel Management System
2. Credit Card Processing
3. Library Management System
4. Stock Maintenance System
5. Passport Automation System

## 1. Hotel Management System

### SRS - Software Requirements Specification

#### SRS Document of Hotel management System

##### 1. Introduction

1.1 Purpose of this document: This document intends to provide an overall insight into the requirements, budget, schedule and constraints involved in the development of a software for hotel management.

1.2 Scope of this document: This document includes essential details of the project that can be used by all stakeholders in the duration of the project.

1.3 Overview: The hotel management system is a web application that allows user to book hotels & rooms in the location of their choice. Specific hotel (client). It also allows the user to avail the other facilities such as food options, room service etc.

##### 2 General description

Objective of the user: The user intends to book rooms of the hotel conveniently with just a few clicks and also avail the other amenities of the hotel.

##### Features:

- Display various room options available
- Options to avail room service and cleaning services
- Option to access food options

Versatile payment system to facilitate  
easy transactions.

Benefits:

Allows the user to book rooms easily  
without any hassle.  
makes the user's stay comfortable as  
they obtain access to the hotel's  
facilities via the app.

### 3. Functional requirements:

- Room booking : Reserve rooms for the guest accordingly in the database after successful payment.
- Options to avail amenities once checked in ( Eg. Room Service → An option where clicked passes the message to the particular hotel's system which is further communicated to the room service department.)
- Cancellation facility via website
  - Payment via debit, credit or UPI.

### 4 Interface requirements

- Communication with the user : A website with user friendly interface .
- Communication with the hotel : Updates in their database ( their database and the website is connected ) after the booking

### 5. Performance requirements:

- ~~Security~~ Speed - The data should be transferred at 25 Mbps / second
- Time - The entire booting process must take less than 1 minute

### 6. Design constraints:

- Use of a NoSQL database
- ~~Time~~ Availability of different methods for payment (credit, UPI, debit etc.)

### 7. Non-functional requirements:

- Secure payment gateway that validates the UPI ID, 16 digit credit / debit card number
- Robust : Recovery mechanism present in case of a failure or doesn't fail fast.

### 8. Preliminary Schedule and Budget

→ Requirement Specification - 11/01/2024

→ Design - 31/01/2024 to 6/01/2024

→ Development - 8/01/2024 to 20/01/2024

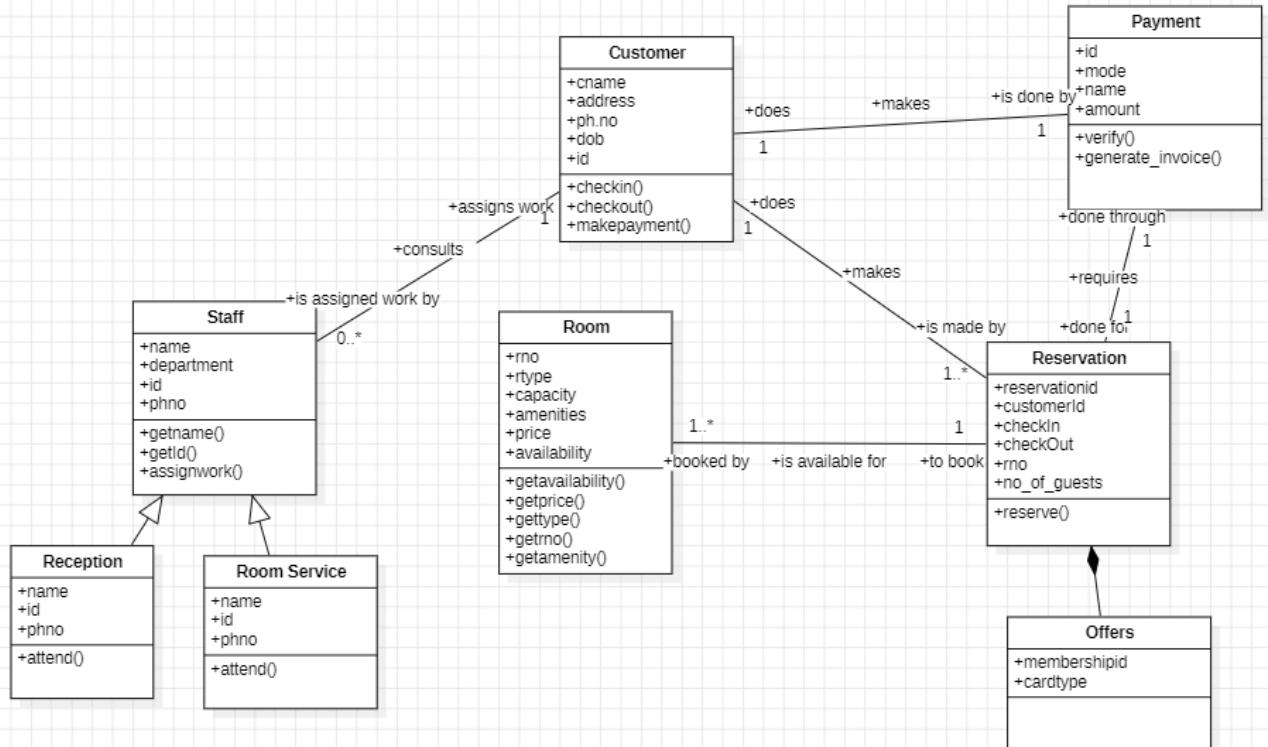
→ Testing - 21/01/2024 to 22/01/2024

Completion - 23/01/2024

✓ Basis and milestones for delivery

D/30/23 Basis and milestones for delivery

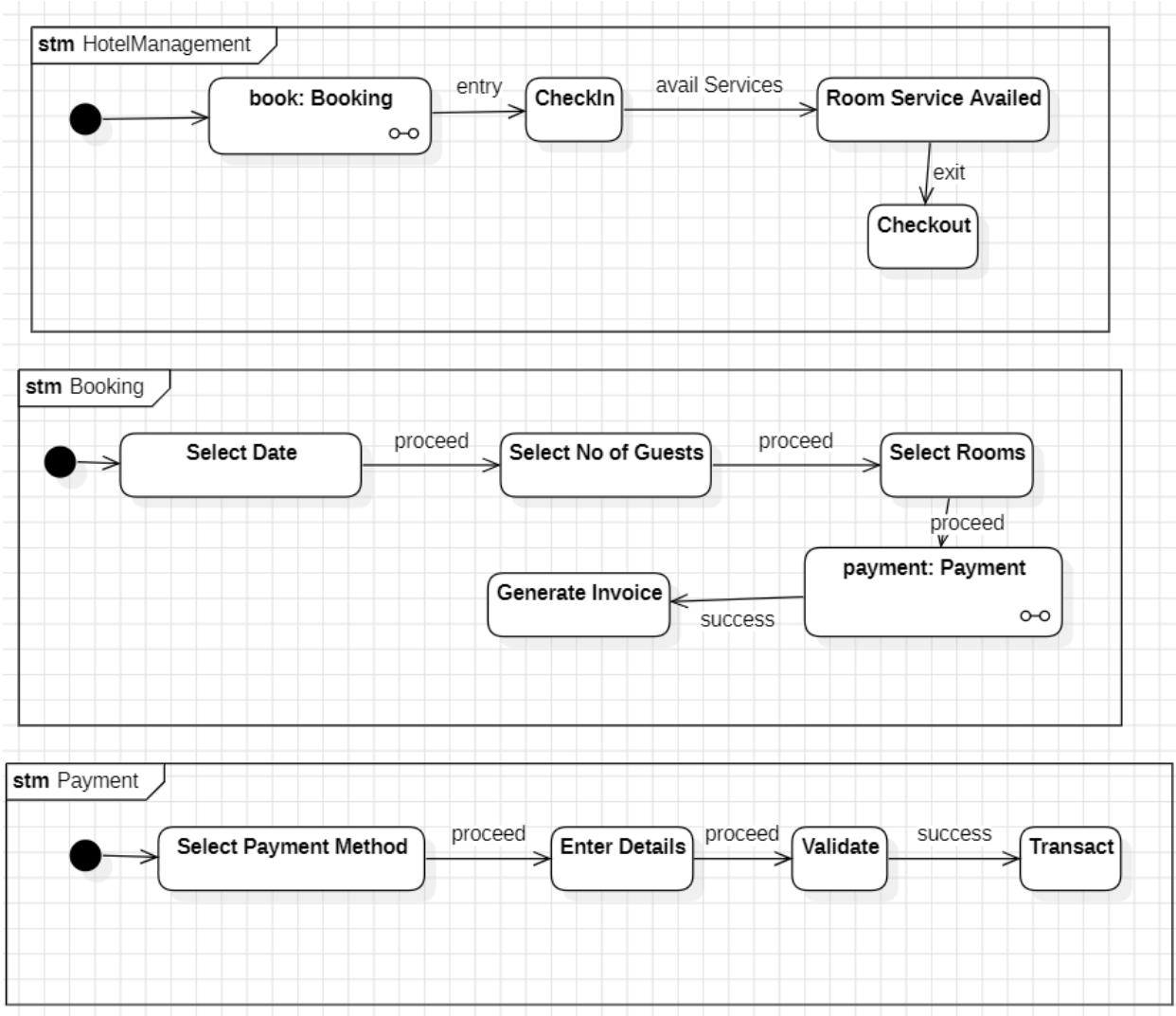
## Class Diagram



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.

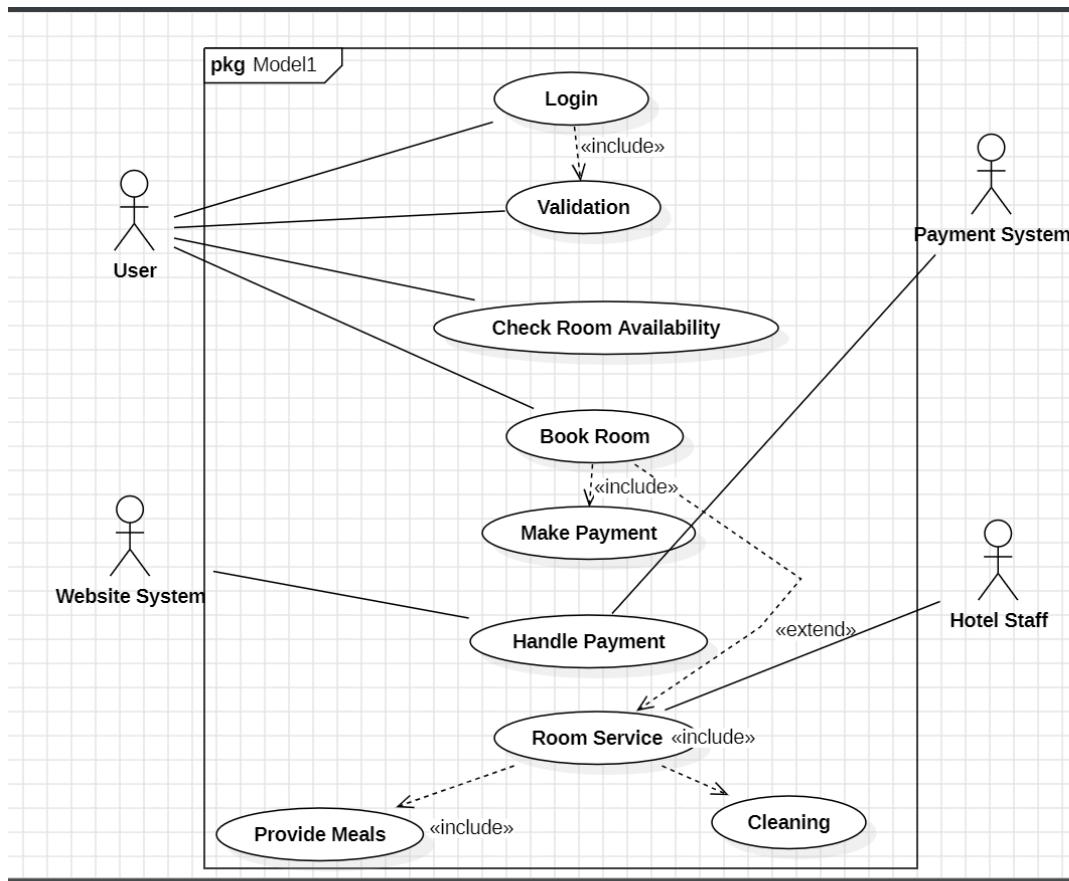
## State Diagram



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



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:

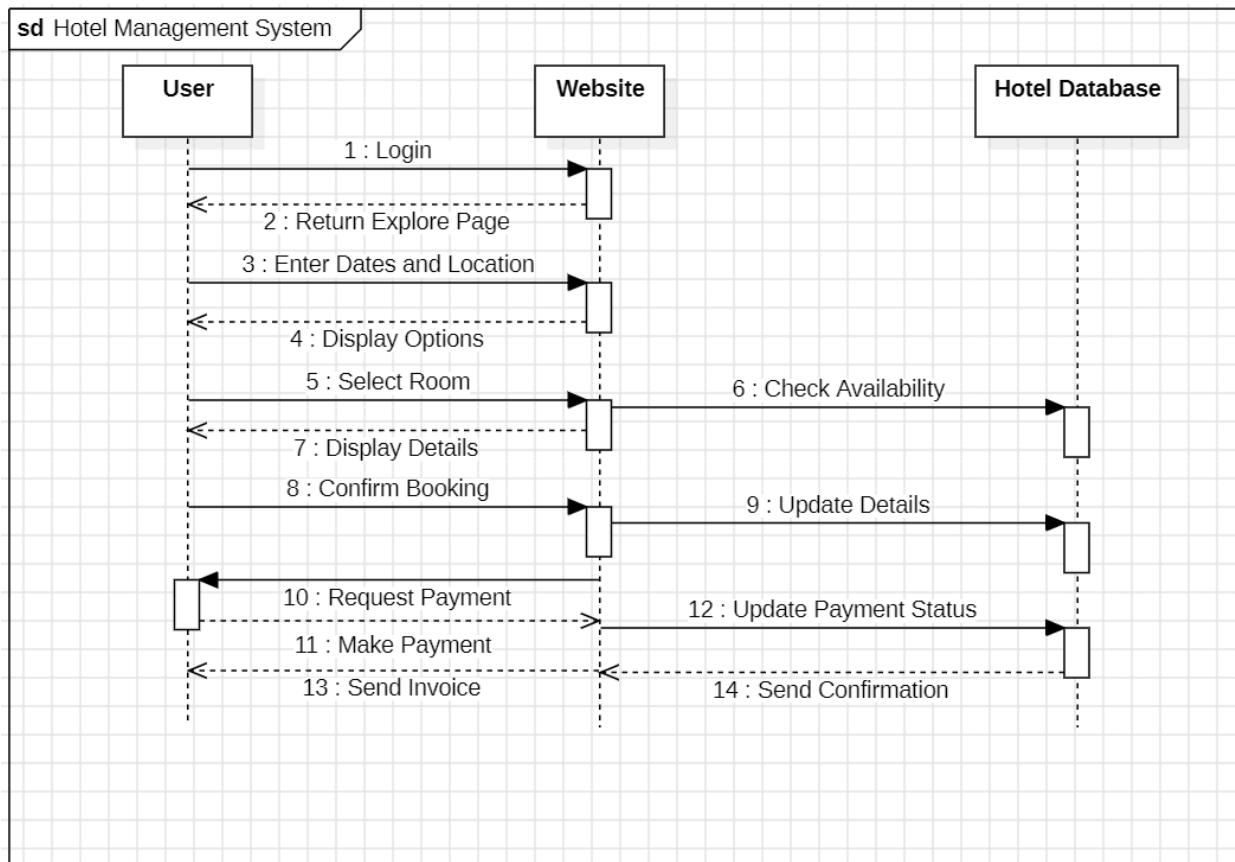
Validation - Validates the user from its database.

Handle payment - Also handles payment process.

Hotel Staff:

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

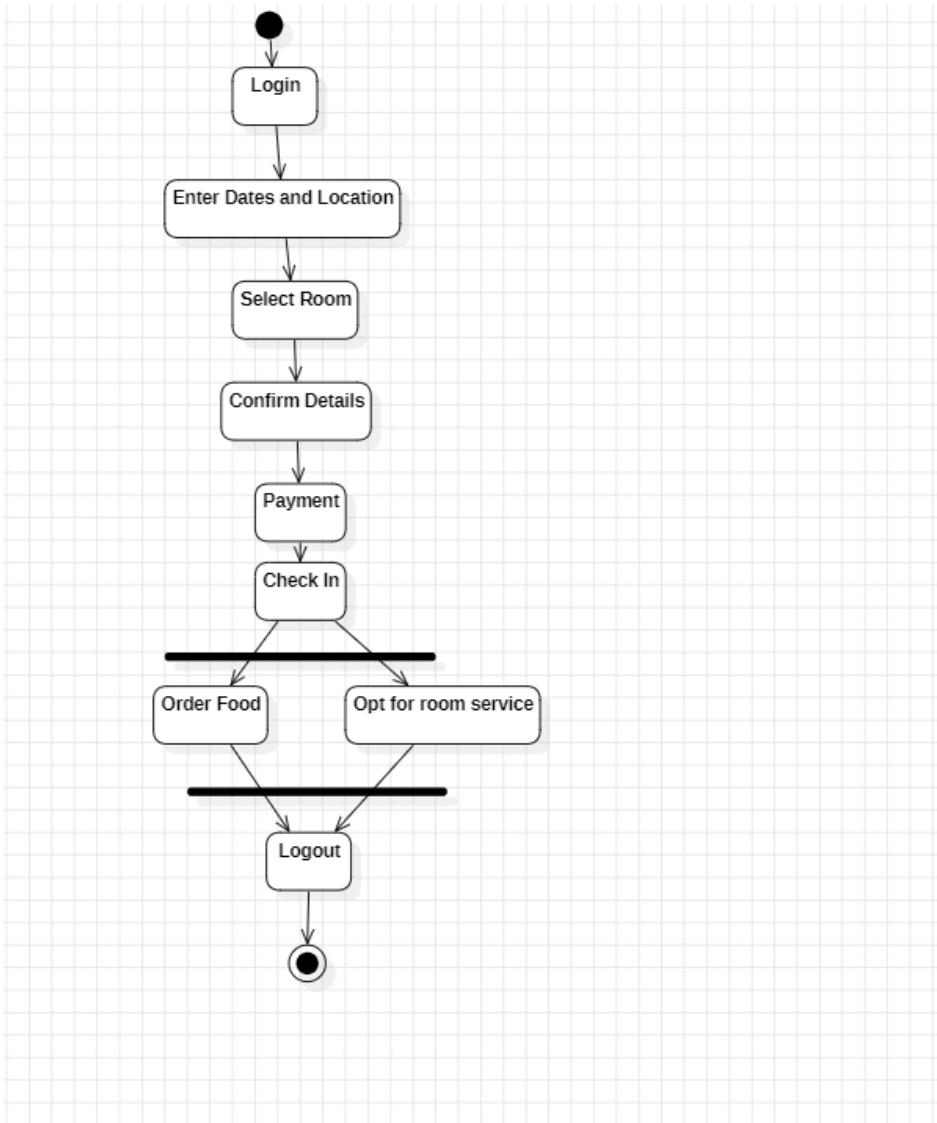
Sequence Diagram



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.

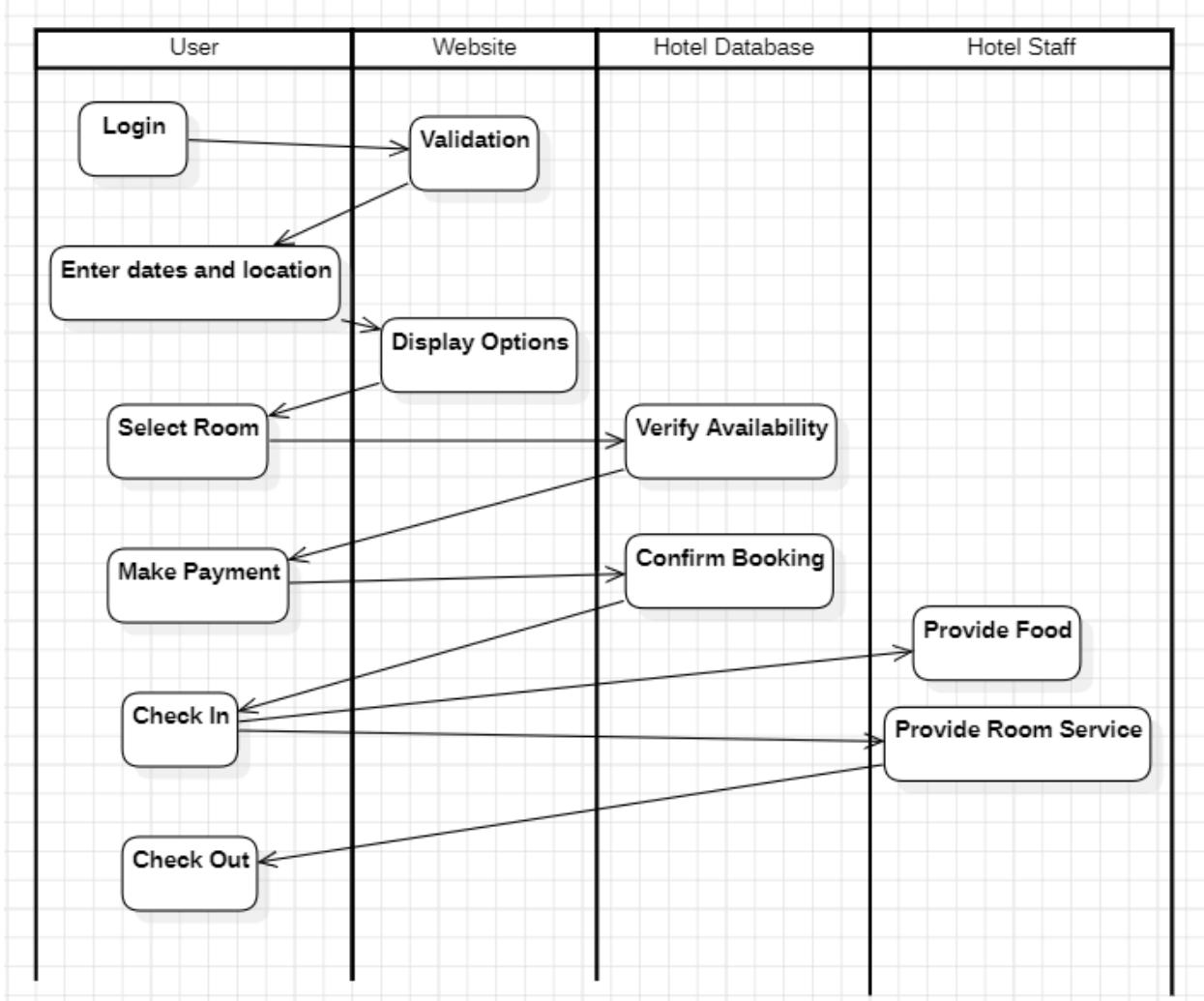
Activity Diagram



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.

## Activity Diagram using Swimlane



The diagram includes 4 swimlanes namely - User, Website, Hotel Database and Hotel Staff.

## 2. Credit Card Processing System

### SRS - Software Requirements Specification

30/9/2024 SRS Document for Credit Card System

#### 1 Introduction

1.1 Purpose of this document : This document intends to explain the requirements, design and development, algorithm, costs of a credit card system.

1.2 Scope of this document : This document includes essential details (requirements, costs, time etc.) of the project that can be referenced by all the stakeholders in the duration of the project.

1.3 Overview : A credit card system is a machine that allows to perform credit card transaction via cards issued by various banks. Its hardware includes a metallic body, a display system, buttons and a card swiper or tap to pay unit. Its software includes an algorithm that traces the user's bank account and performs the respective transaction (add to credit amount).

#### 2 General description

Objectives of the user : The retailers/wholesale who have procured the machine must be able to perform transactions for the customers or more specifically the customer who are the beneficiaries must be able to view the amount under their credit card dues.

### Features:

User friendly interface to facilitate transaction.

Diverse bank base (to permit transactions via a vast variety of cards).

Fast transactions

Dual mode (Swipe and tap to pay)

### Benefits:

Allows customers to buy items on credit easily.

Allows retailers / wholesalers to obtain a wide customer base.

### 3. Functional requirements:

- Swipe to pay system - On card swipe a transaction must be initiated.
- Tap to pay - On card tap a transaction must be initiated.
- PIN entry display - If the card is valid allow the customer to enter a pin.
- Button for transaction - On pressing the button if the pin is valid perform the transaction.
- Generate receipt on successful transaction.
- Perform internal transaction (add to credit amount) in the customer's bank account.

### 4. Interface Requirements:

- Communication with the user : Via a display screen.

### 5. Performance requirements:

- Accuracy - The machine must be able to perform 100% accurate transactions.



## 8 Preliminary Schedule and Budget

The project is intended to be started on  
October 1, 2024 (1/10/2024)

→ Requirement Specifications - 1/10/2024

→ Design - 3/10/2024 to 6/10/2024

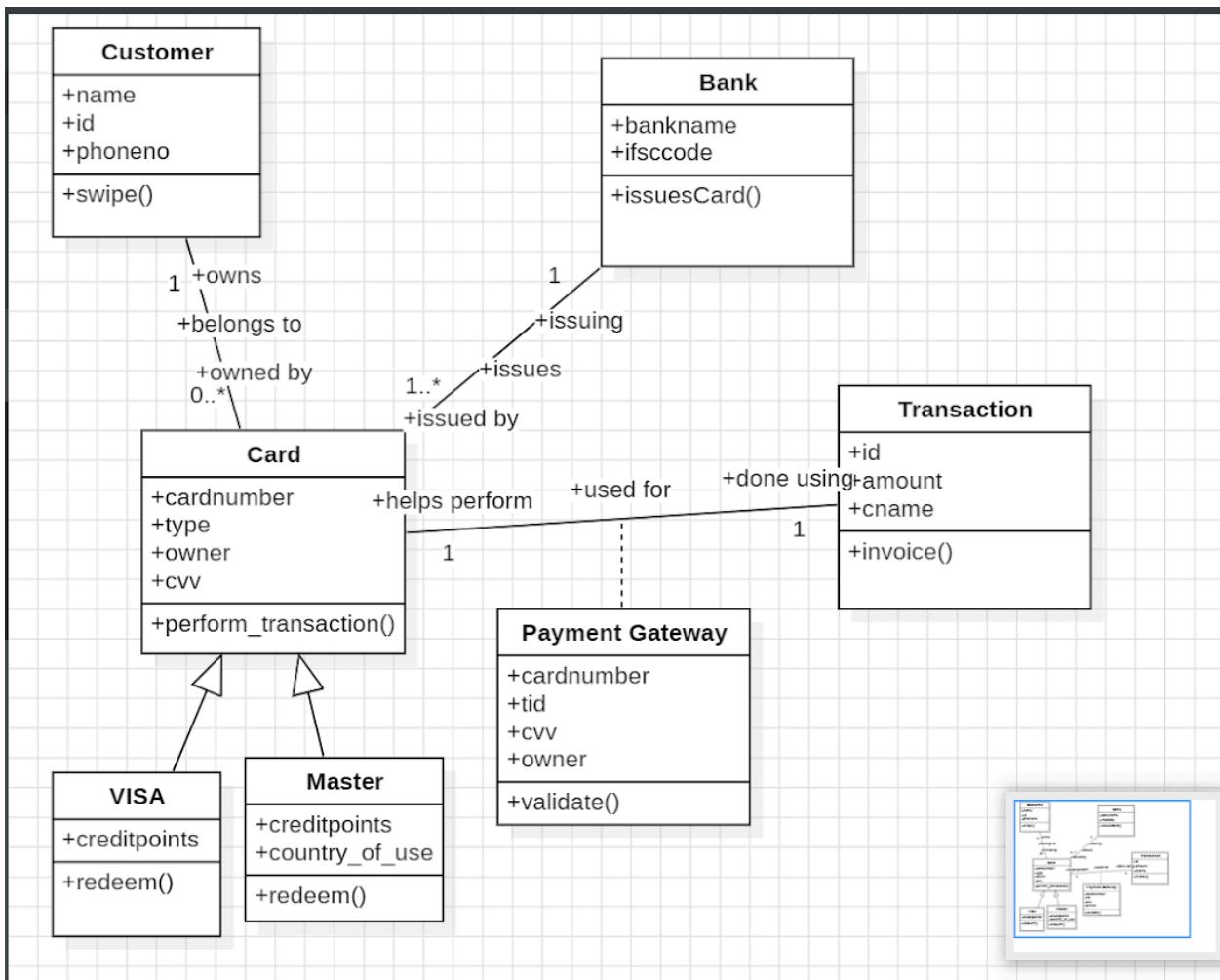
→ Development - 8/10/2024 to 20/10/2024

→ Testing - 21/10/2024 to 22/10/2024

Hence the project is intended to be completed  
by 23/10/2024 with a budget of ₹ 400.

✓

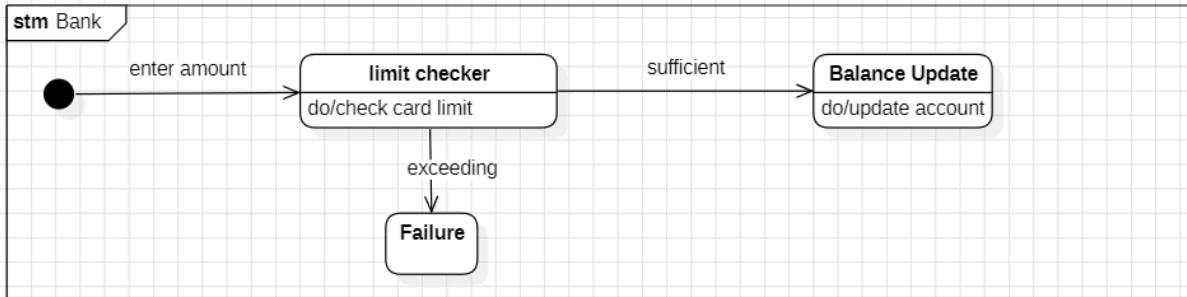
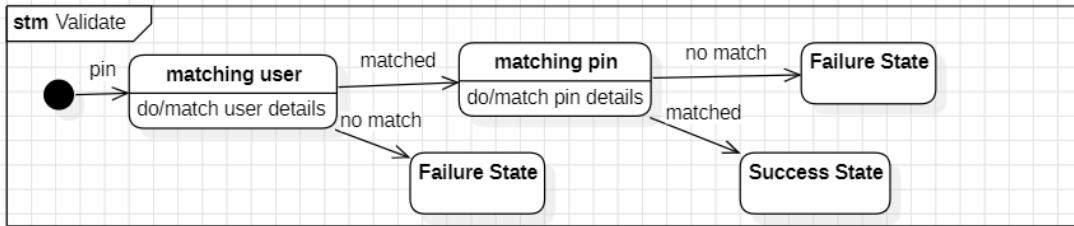
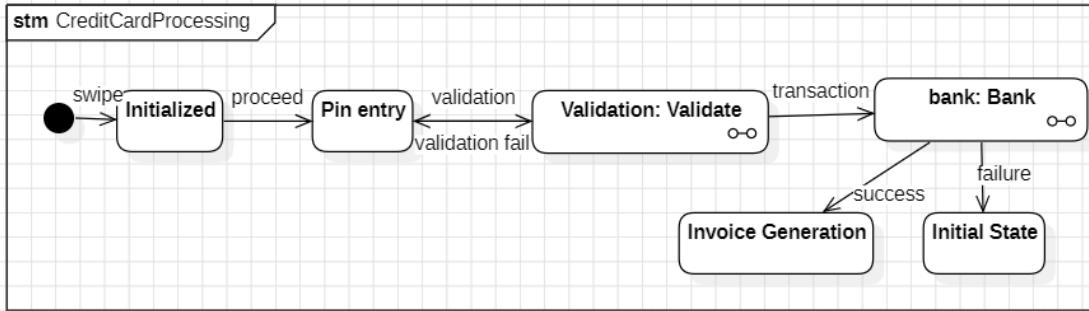
## Class Diagram



## 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.

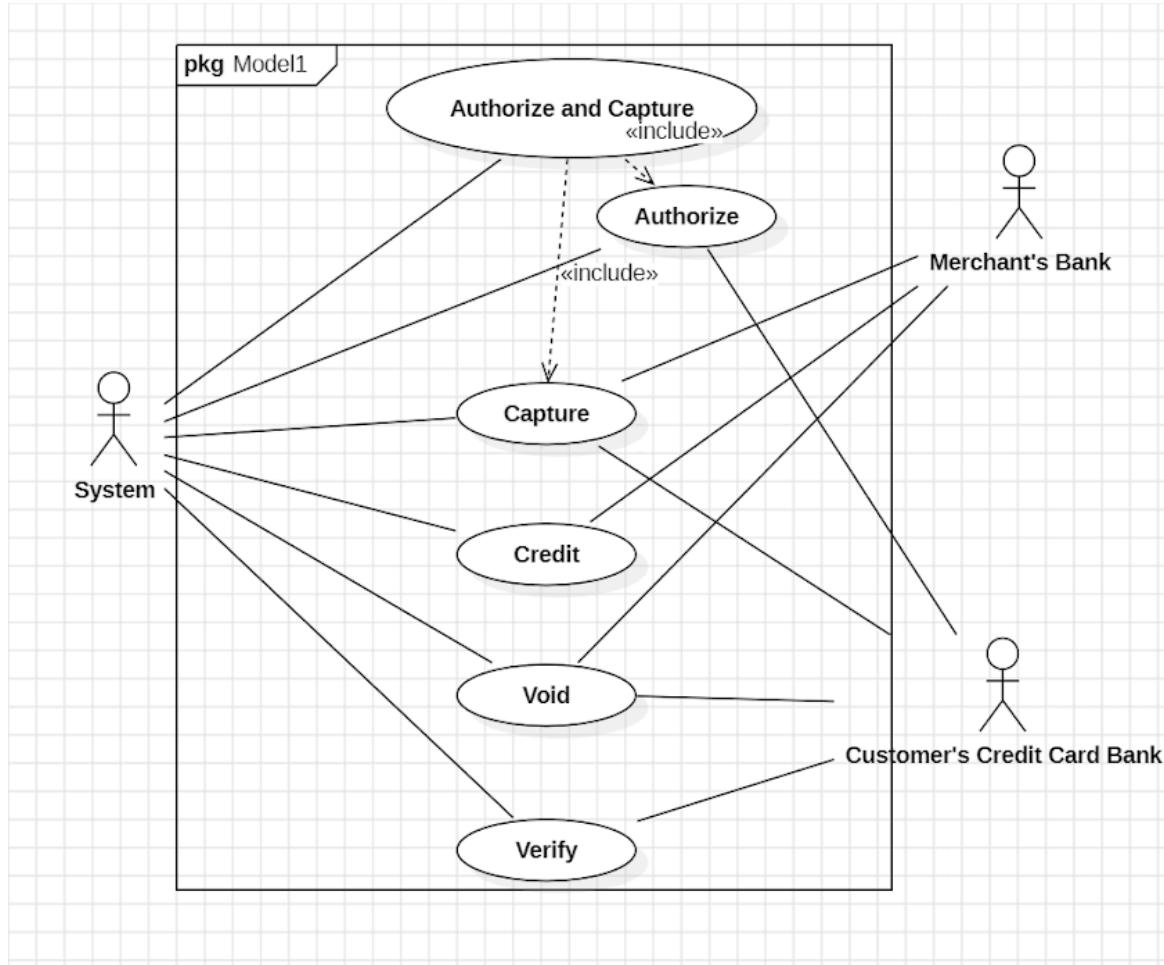
## State Diagram



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.

## Use Case Diagram



Description:

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

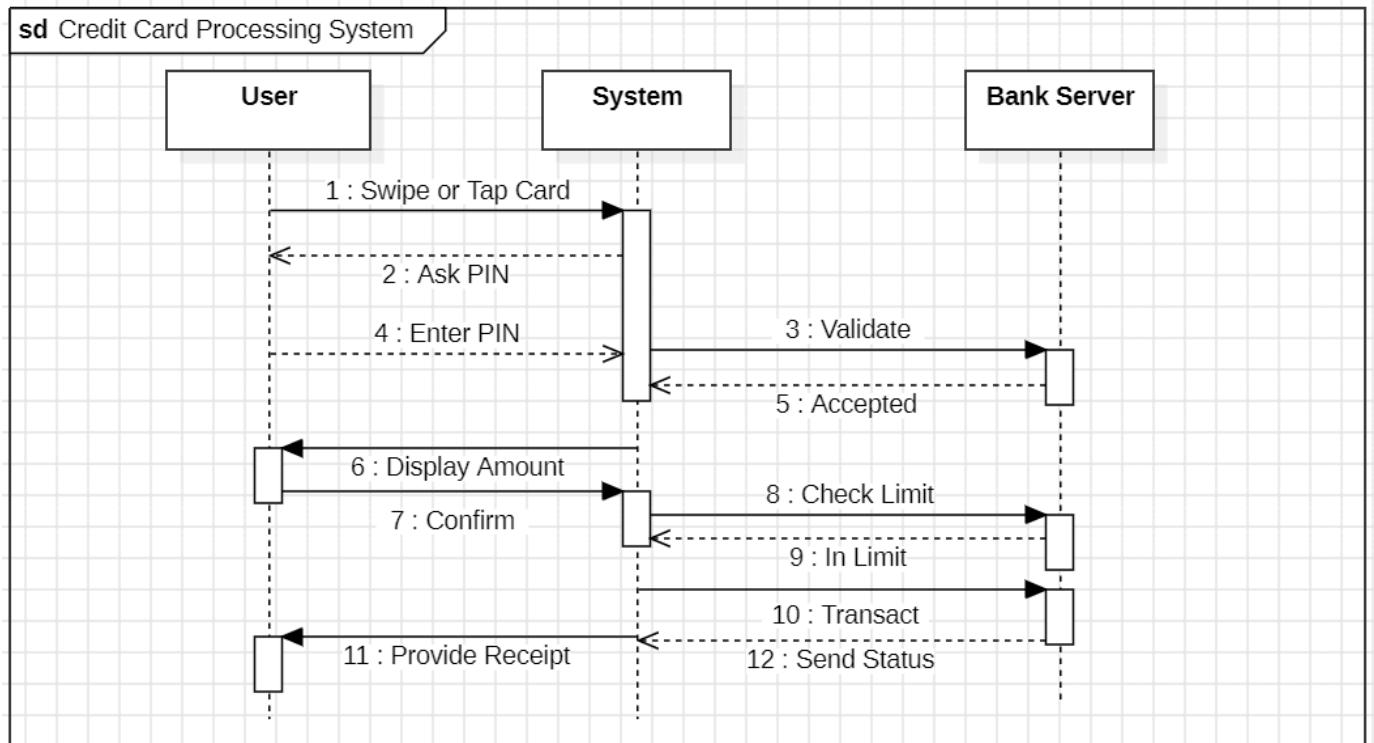
System: Initiates and manages multiple use cases, including:

- 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.

Customer's Credit Card Bank: Validates the authorization and ensures sufficient funds for the transaction.

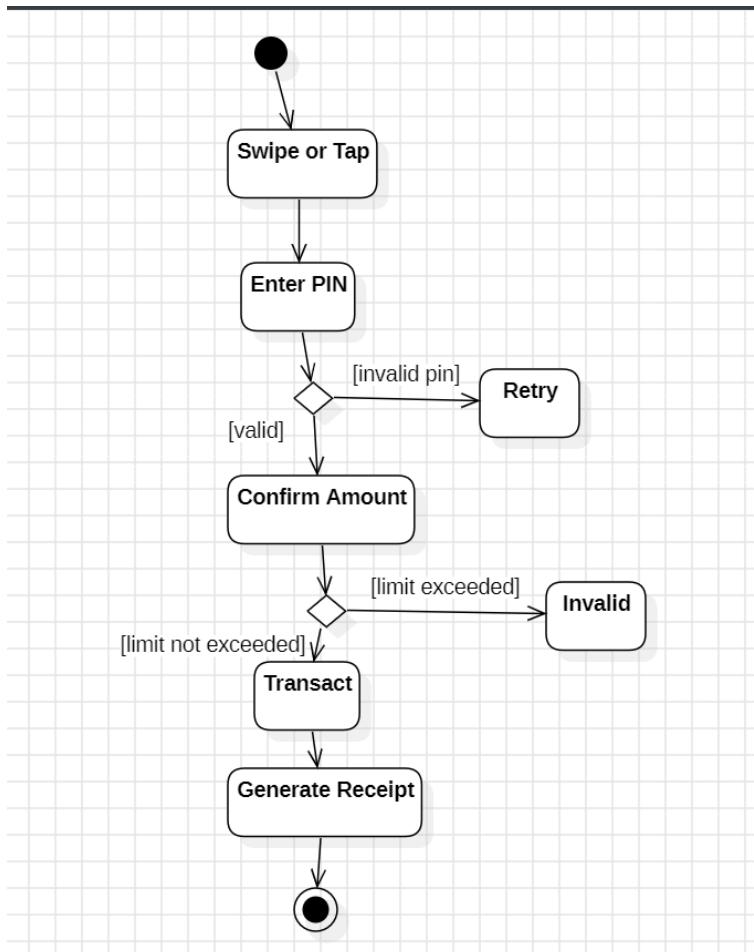
### Sequence Diagram



### 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.

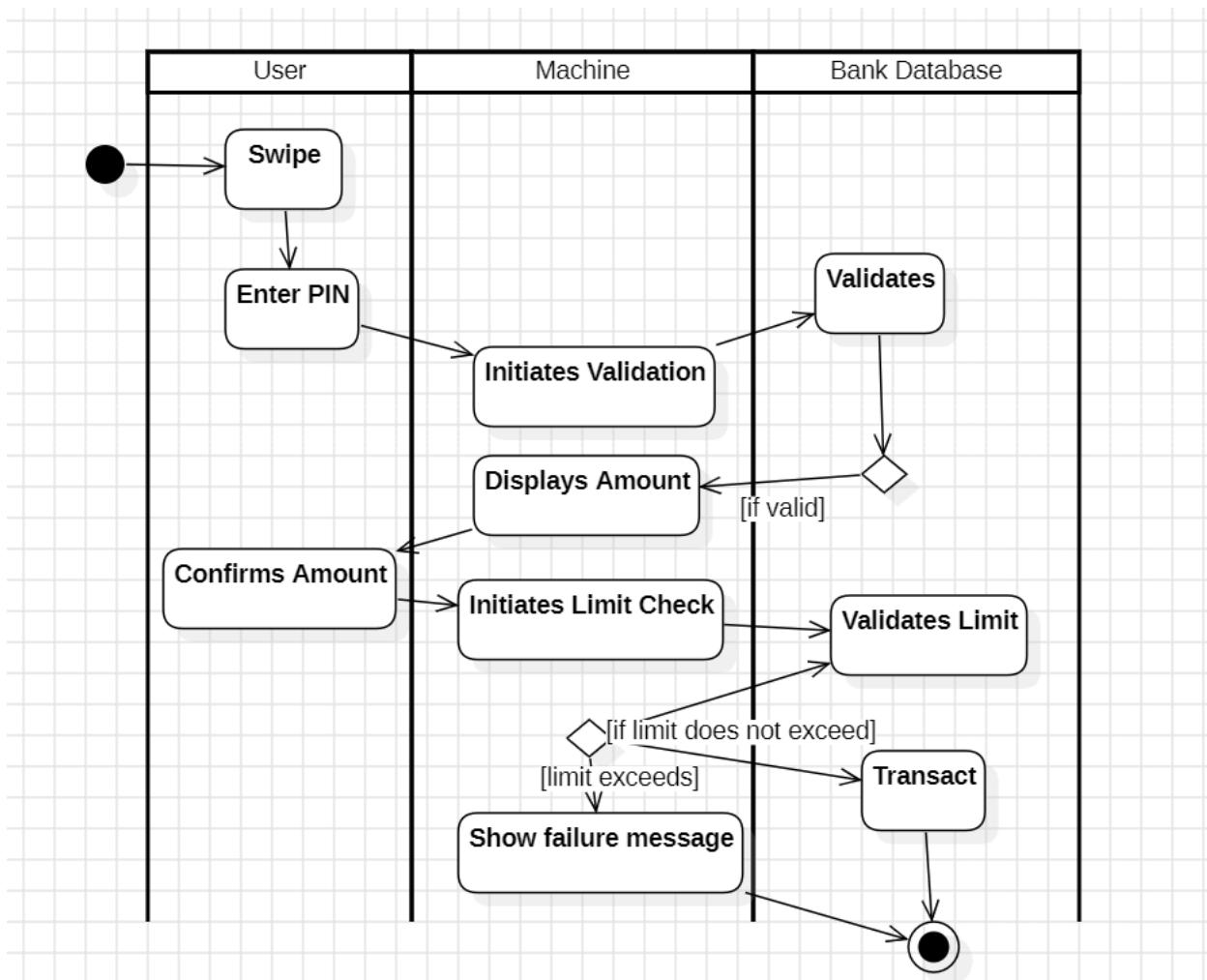
## Activity Diagram



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.

## Activity Diagram using Swimlane



There are 3 swimlanes namely User, Machine and the Bank Database.

### 3. Library Management System

SRS - Software Requirements Specification

07/10/2024

## SRS Document of Library Management System

### 1. Introduction

1.1 Purpose of this document: This document is to provide an overview of the requirements, budget, schedule and constraints involved in the development of a software for library management.

1.2 Scope of this document: This document includes essential details that can be referenced by all the stakeholders during the development.

1.3 Overview: The product - A library management system ~~process~~ Software includes a user interface and an admin interface via which borrowing operations are managed.

### 2. General description:

Objective of the user - The users here are two - the students / borrowers and the admin / library manager. So the borrower would want to record the details of who has borrowed and <sup>when</sup> ~~when~~. The admin would want to keep a note of the possible fines and approve any requests for books.

Features:

A user friendly interface to record borrowing details.

Inbuilt functionality to calculate fines in case of exceeding due date.

Functionality to launch a request to procure any books or ask for pdf or printable copies.

Benefits:

- This software helps in easy tracking of borrowing and returning by keeping a record of each borrower's details.
- Reduces usage of paper as the register now goes digital.

### 3 Functional requirements

- ~~• Login / Signup :~~ A user login / signup and an admin login / signup which ~~will help~~ in registering a user by valid username and password.
- User Dashboard : A dashboard that helps the user to track their borrowing where columns such as Name of Book, Date of Issue, Date of Return and Fine are mentioned.
- Admin Dashboard : A dashboard that helps the library manager to keep track of all the borrowers, their due dates and fines in case of delays.

### 4 Interface requirements

- ✓ A UI made using React, HTML, CSS with 2 navigation routes one for user and one for admin.

- Communication established wirelessly with a printer to facilitate printing of articles.

## 5 Performance requirements

Speed - The data must be transferred at the rate of 25 mbps. This helps in loading the page faster.

Time - The process of storing the data in the database must take less than a minute.

## 6 Design constraints

- Use a NoSQL database.
- Use React.js for the website frontend to help better routing.
- Use Figma for the design.
- Comply with copyright restrictions while allowing downloads of e-books or ~~printing~~ of e-books.

## 7 Non functional attributes

• Reliability - The software must have ~~very~~ a well established back-up so as to help recover from any failures.

• Authorization - Using the firebase authentication system a valid username and a password no less than 8 characters (including alphabets, numbers & special characters) is validated.

• Compatibility : It must be compatible on both Desktop and mobile (ios and android).

8 Preliminary Schedule and Budget  
The project is intended to start on 09/10/2024

Requirement Specification - 11/10/2024

Design - 14/10/2024

Development - 24/10/2024

Testing - 26/10/2024

Completion - 27/10/2024

Budget: \$200

~~Project~~ It is important to ensure

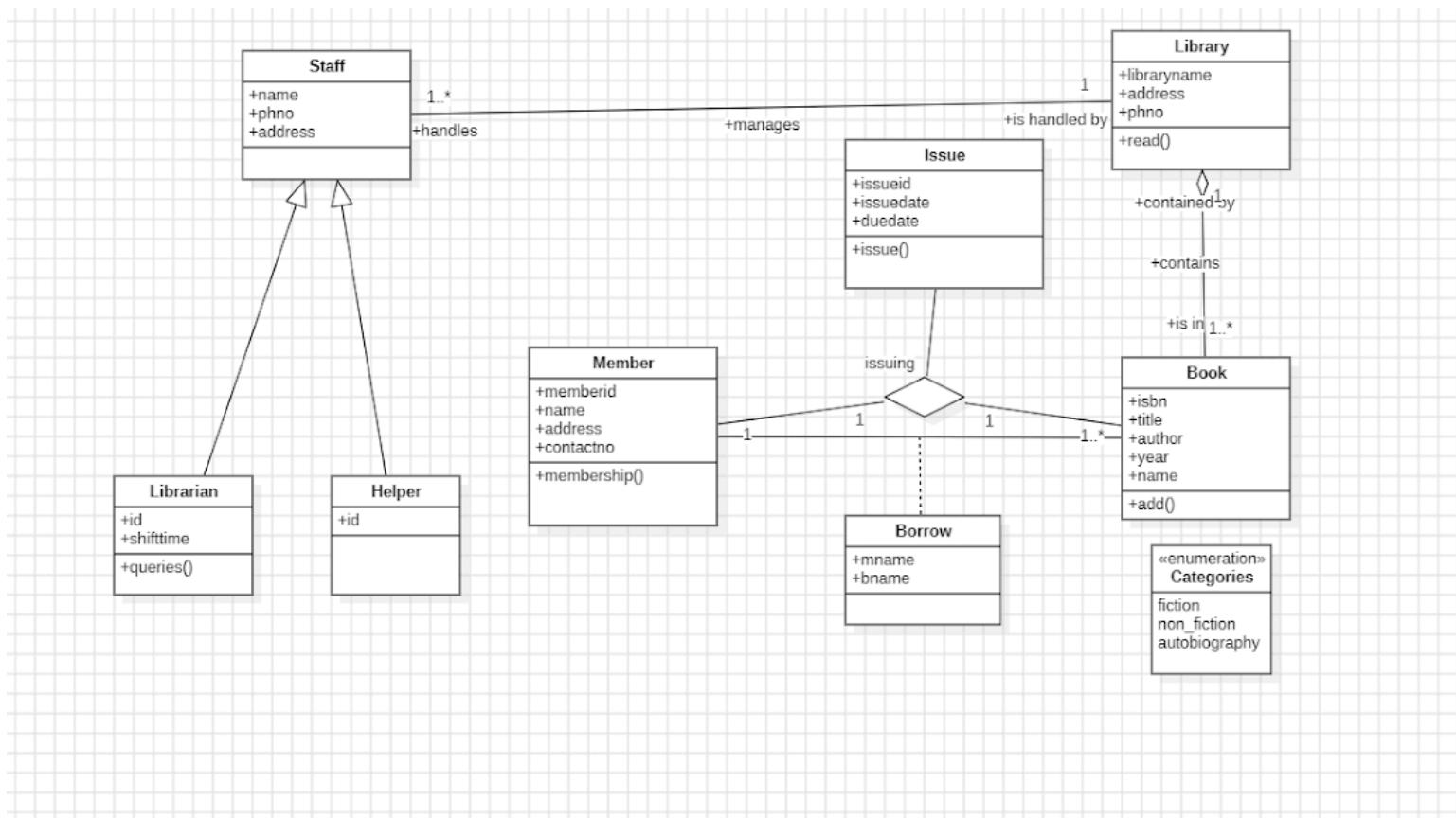
that the project will be successful. It is important to have a good understanding of the requirements and to have detailed plans for the work.

It is important to hire experienced and professional workers for the job. It is better to have a good team of workers than to have many workers who are not experienced.

It is also important to have a good budget. It is important to have a budget that is realistic and can be met. It is also important to have a budget that is flexible and can be adjusted if necessary.

It is important to have a good plan and to follow it. It is also important to have a good team and to work together. It is important to have a good budget and to stick to it. It is also important to have a good timeline and to meet the deadline.

## Class Diagram

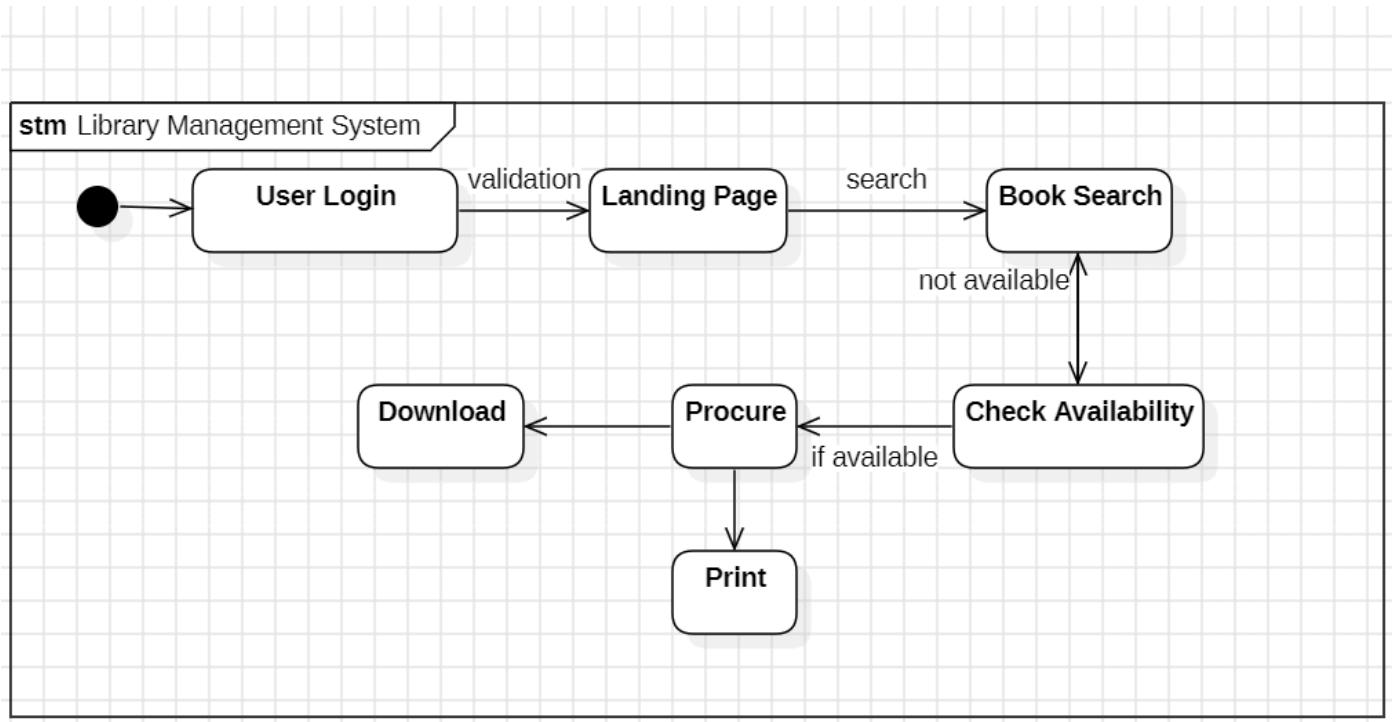


### Description:

- **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.

- Categories: Represent an enumeration for book classifications (fiction, non-fiction, autobiography).
- Through these relationships, the diagram incorporates generalization (Staff to Librarian/Helper), composition (Library to Book), and aggregation (Library to Staff).

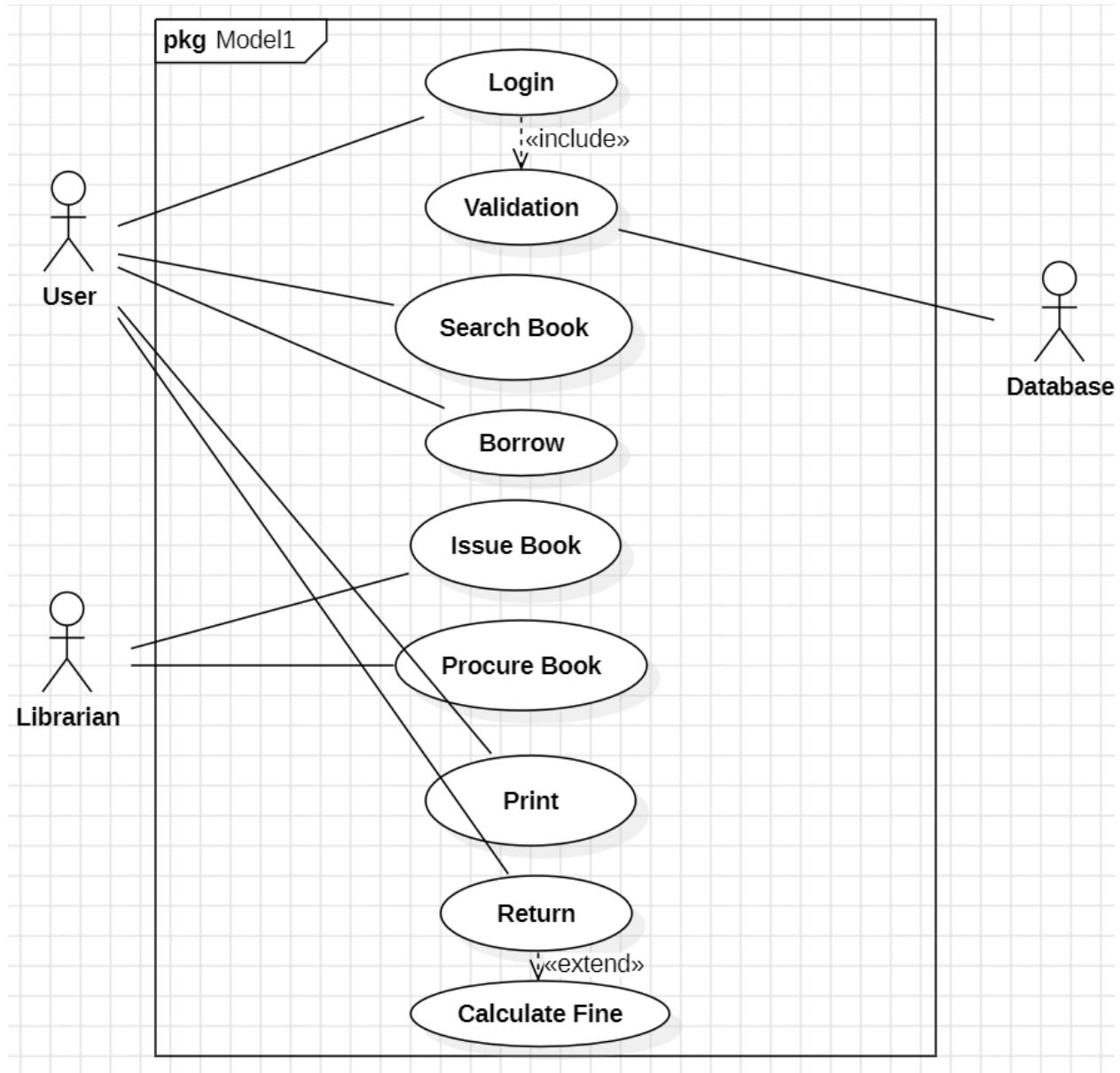
## State Diagram



## 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



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.

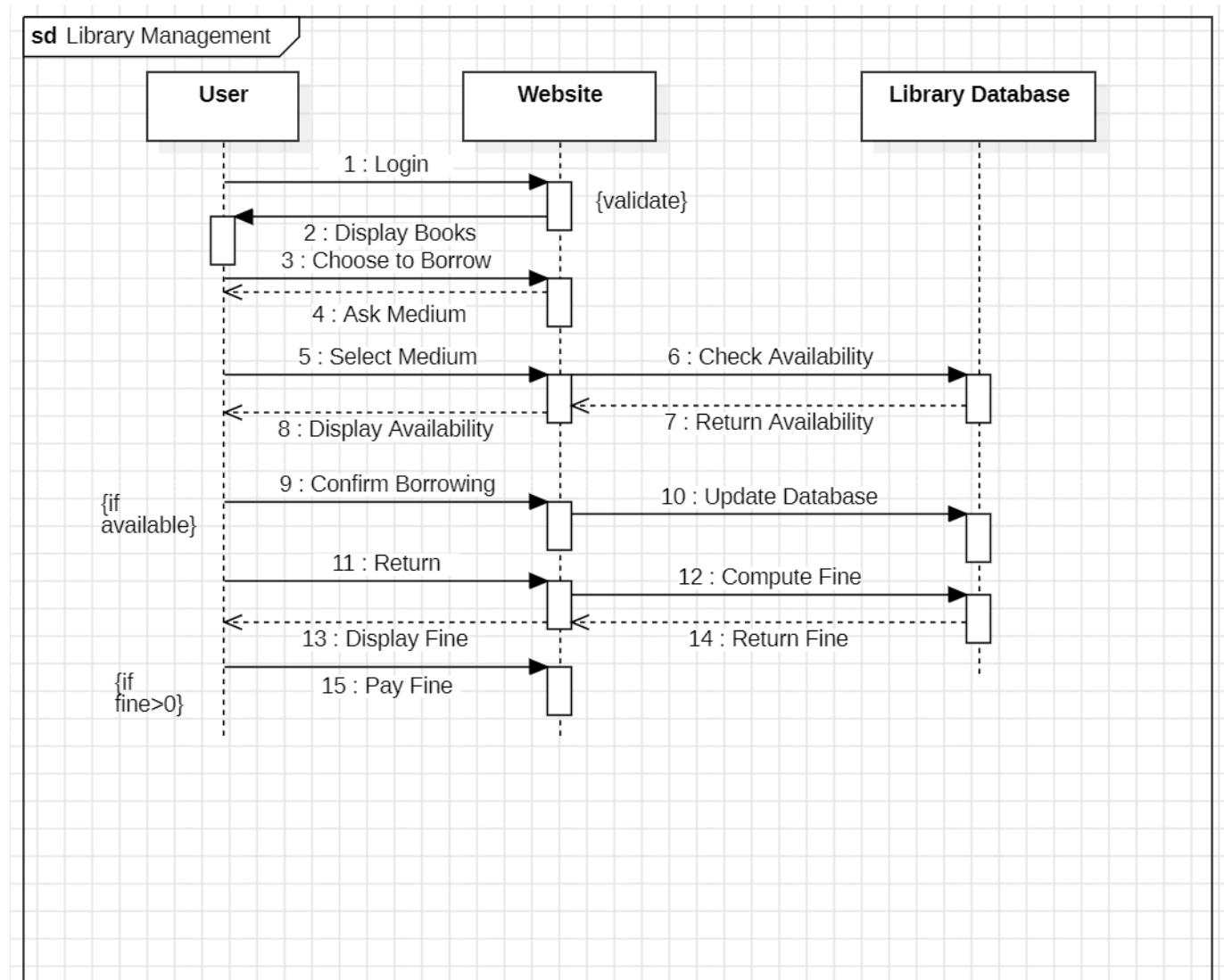
- Borrow: Users can borrow books, which triggers the Issue Book use case managed by the Librarian.
- Return: Handles the process of returning borrowed books and extends to Calculate Fine in case of overdue returns.

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



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.

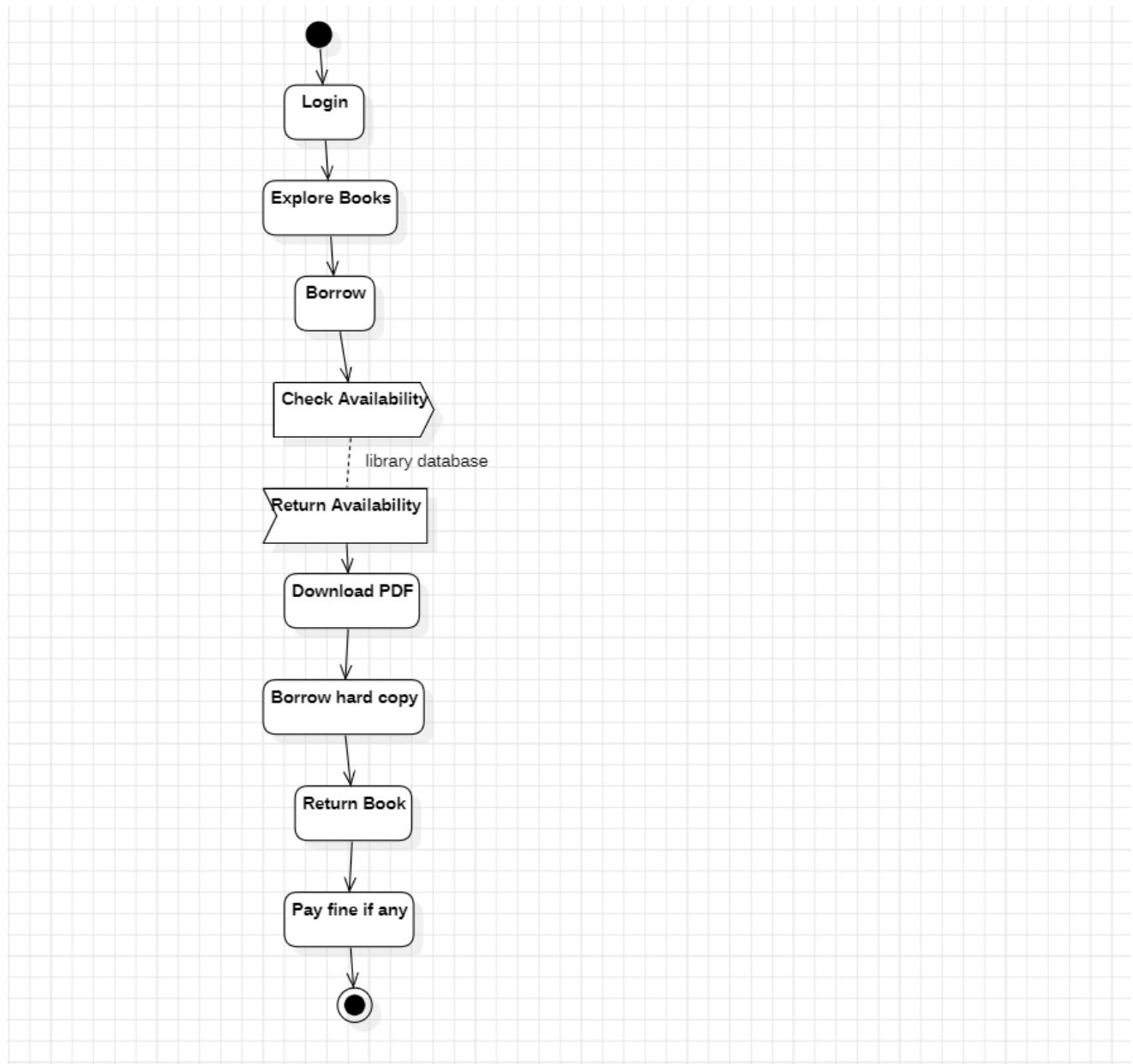
Compute Fine: The website requests the library database to compute any late return fine.

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.

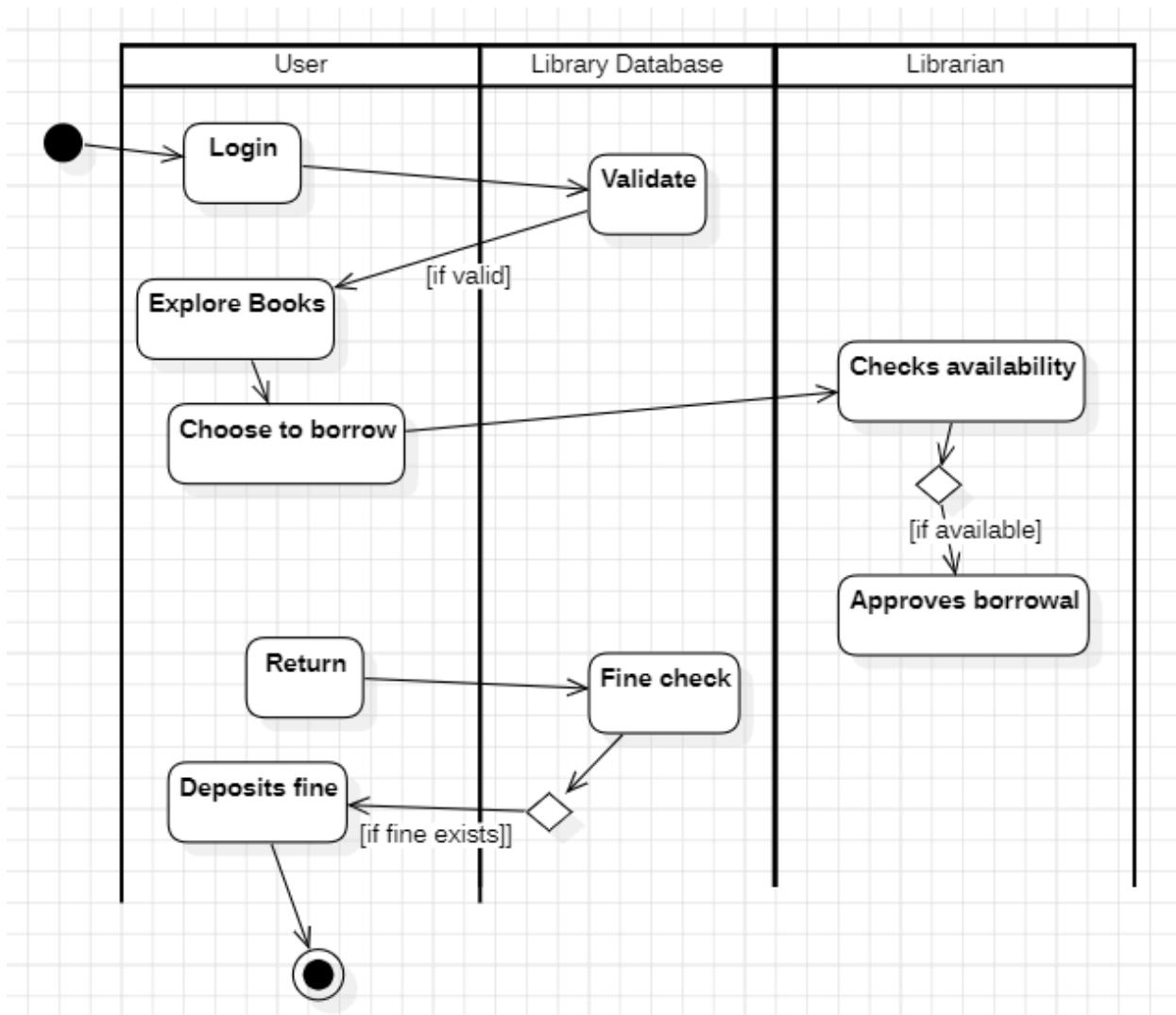
## Activity Diagram



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.

Activity Diagram using Swimlane:



There are 3 swimlanes namely User, Library Database and Librarian.

#### 4. Stock Maintenance System

SRS - Software Requirements Specification

#### SRS Document of Stock Maintenance System

##### 1 Introduction

1.1 Purpose of this document: This document intends to provide an overview of the requirements, schedule, budget and constraints involved in the development of a stock maintenance system.

1.2 Scope of this document: This document contains essential details which can be referenced by all stakeholders during the process of development of the software.

1.3 Overview - The product is used to manage the inventory of the company. It is used to keep a track of the number of items of each category available and the number of items that must be procured.

##### 2 General description

Objective of the user: The inventory manager must be able to keep a record of the goods in stock.

Features: The application includes a user friendly page that helps in updating stock details such as Name of item, Quantity, Upcoming dispatches.

The upcoming dispatches are updated based on the orders.

##### Benefits:

- Allows the stock manager to efficiently keep a record of the stock details in a digital form.

### 3 Functional requirements:

- Stock Records page - A page in which the details such as no. of items, quantity and upcoming dispatch can be recorded.
- Redirecting to Database - On pressing the record button the details must be stored in a database.

### 4 Interface requirements

A user friendly interface that ~~can~~ allows the stock manager to record data.

Communication with the database: Data is stored in firebase via firebase authentication and communication protocol.

### 5 Performance Requirements

Speed: The data must be loaded at the speed of 2.5 Mbps.

Time: It must take less than 1 minute for data to be stored and retrieved.

### 6 Design Constraints

- Use a NoSQL database - Firebase
- Use Figma for designing

### 7 Non functional attributes

- Reliability - A strong back-up to recover data in case of any failures
- Robust - Very less failures and even if failure occurs must be able to revert back immediately.

- Security (Authorization) : The inventory manager must be given username and password credentials which is validated using firebase authorization.

## 8 Preliminary Schedule and Budget

The project is intended to start on

09/10/2024.

Task	Date of Completion	Cost
Requirement Specification	11/10/2024	\$ 5
Design	14/10/2024	\$ 10
Development	24/10/2024	\$ 25
Testing	26/10/2024	\$ 8

Completion - 27/10/2024

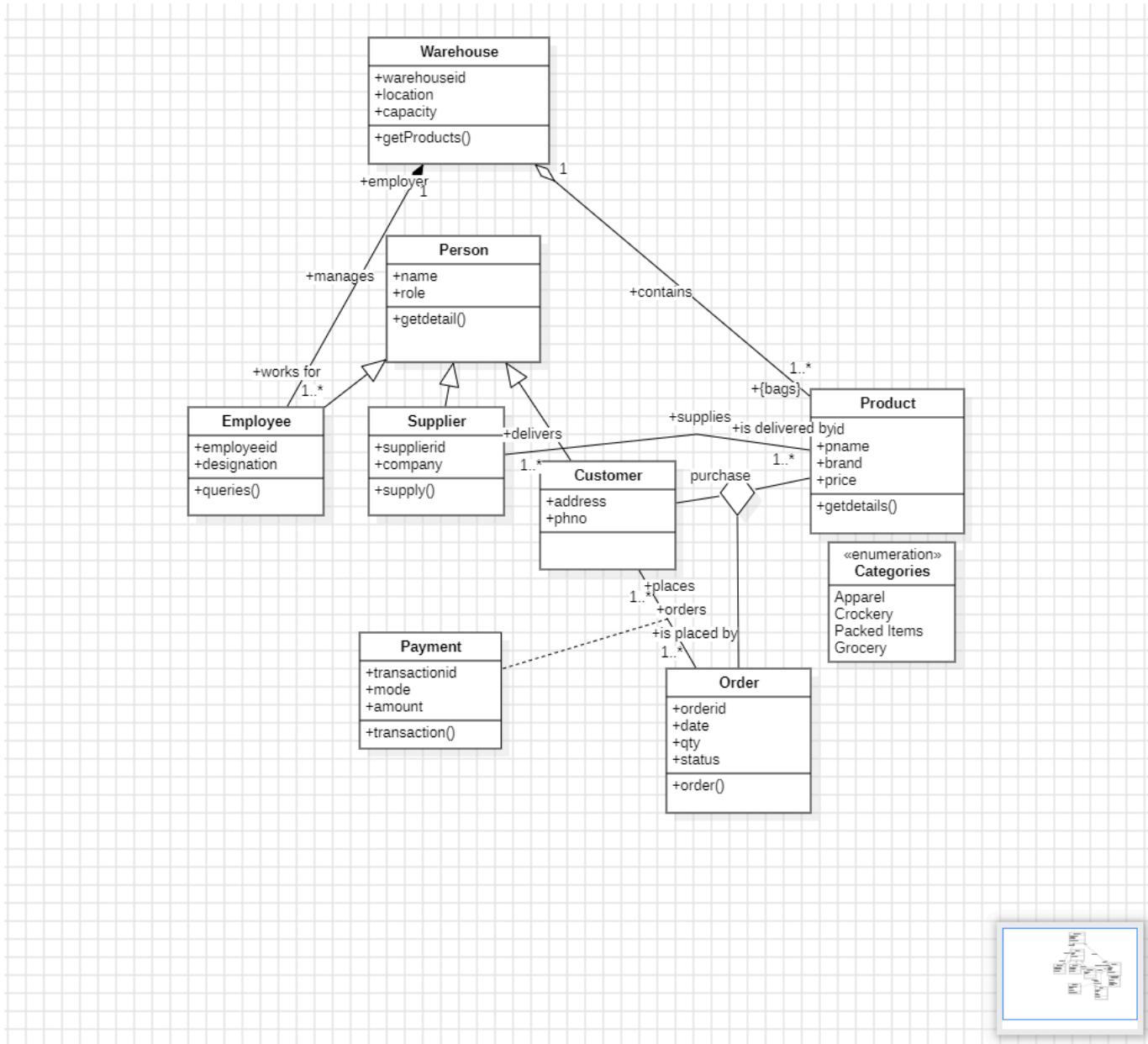
Budget - \$ 48

~~Task 1 completed~~

~~Task 2~~

Final budget breakdown  
basis of proposed tasks A - tasks  
completed tasks B - tasks  
not yet started and yet to be done  
Total budget based on current tasks

## Class Diagram

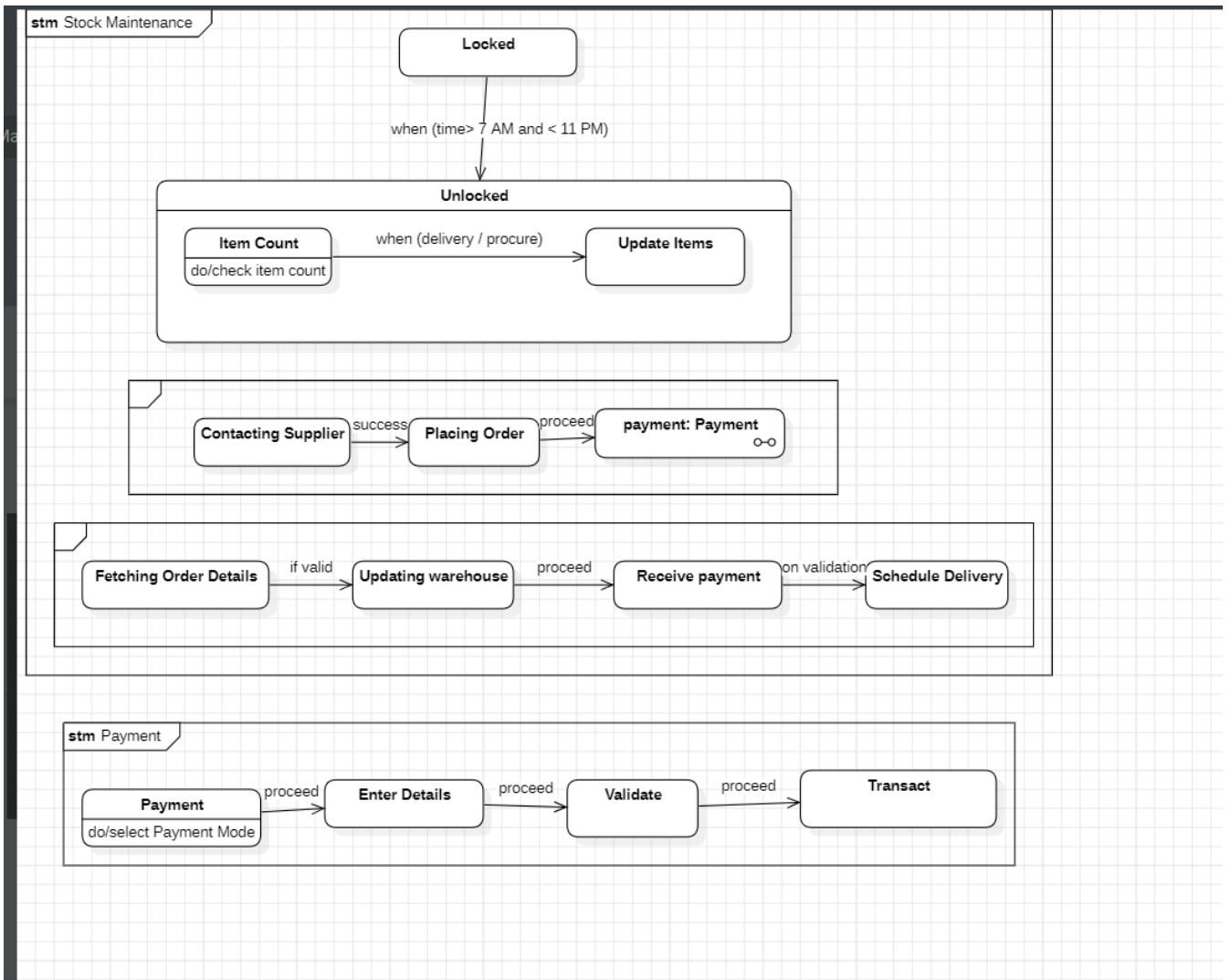


### Description of Class Diagram:

- **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

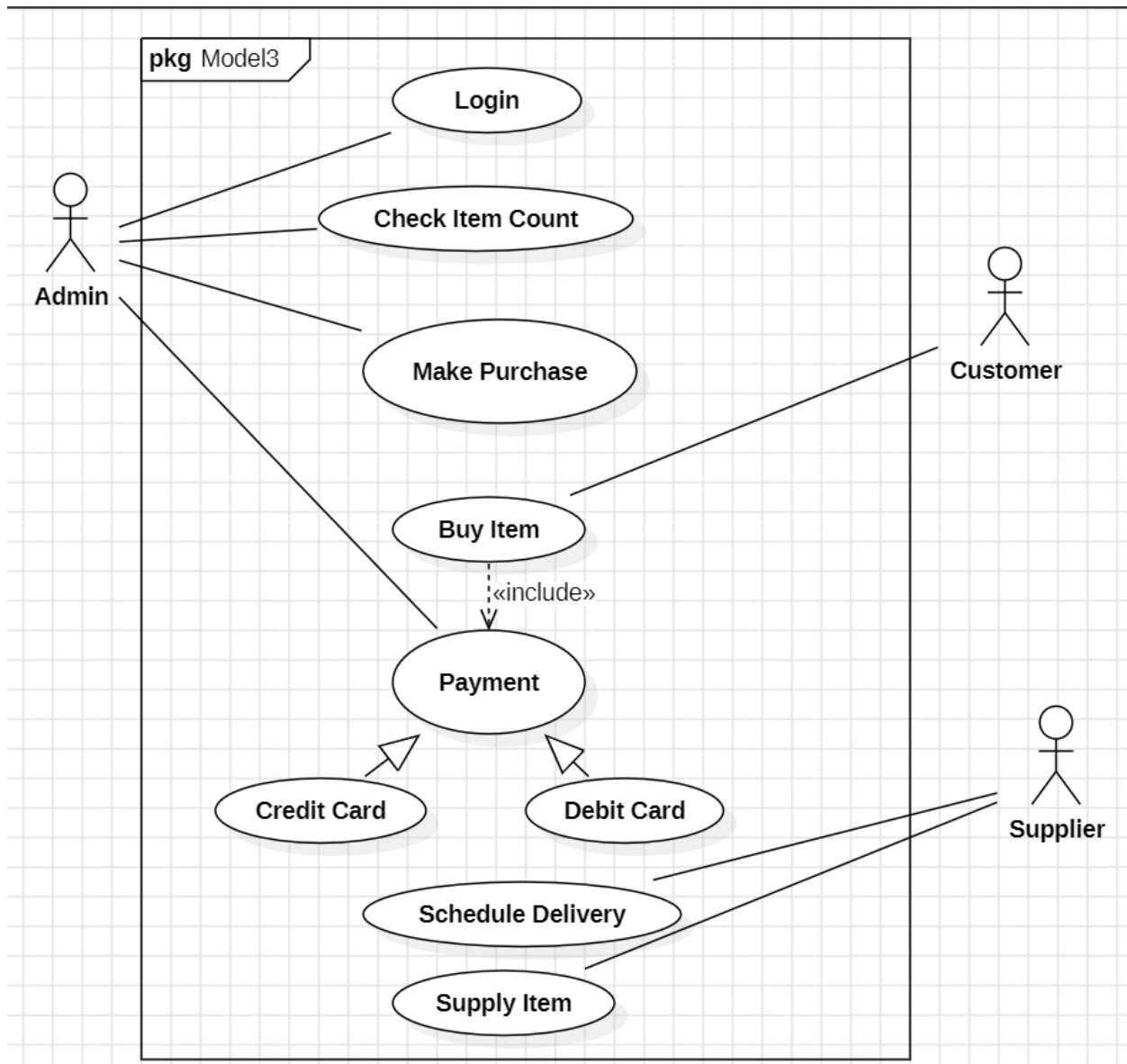


## Description of State Diagram:

- 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 Diagram



## Description of Use Case Diagram:

Actors Involved: Admin, Customer, and Supplier.

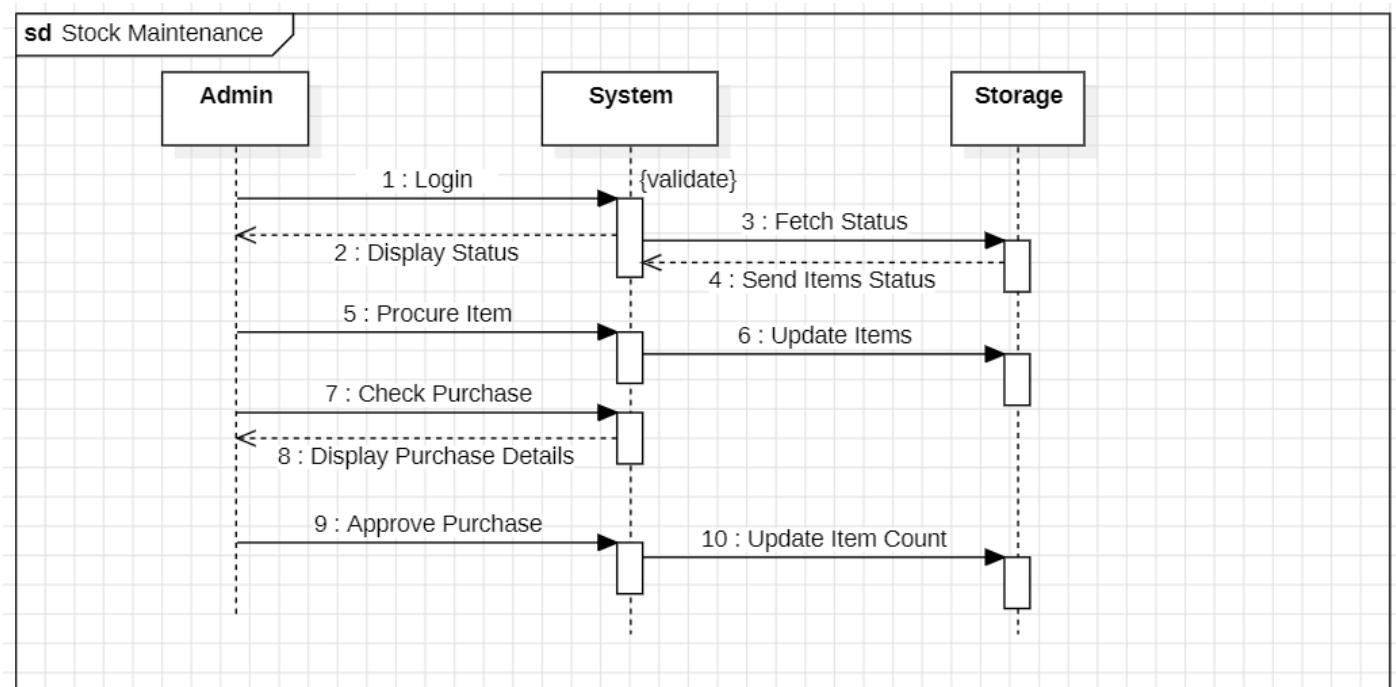
Admin: Logs into the system, checks item count, and makes purchases by buying items.

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



## 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.

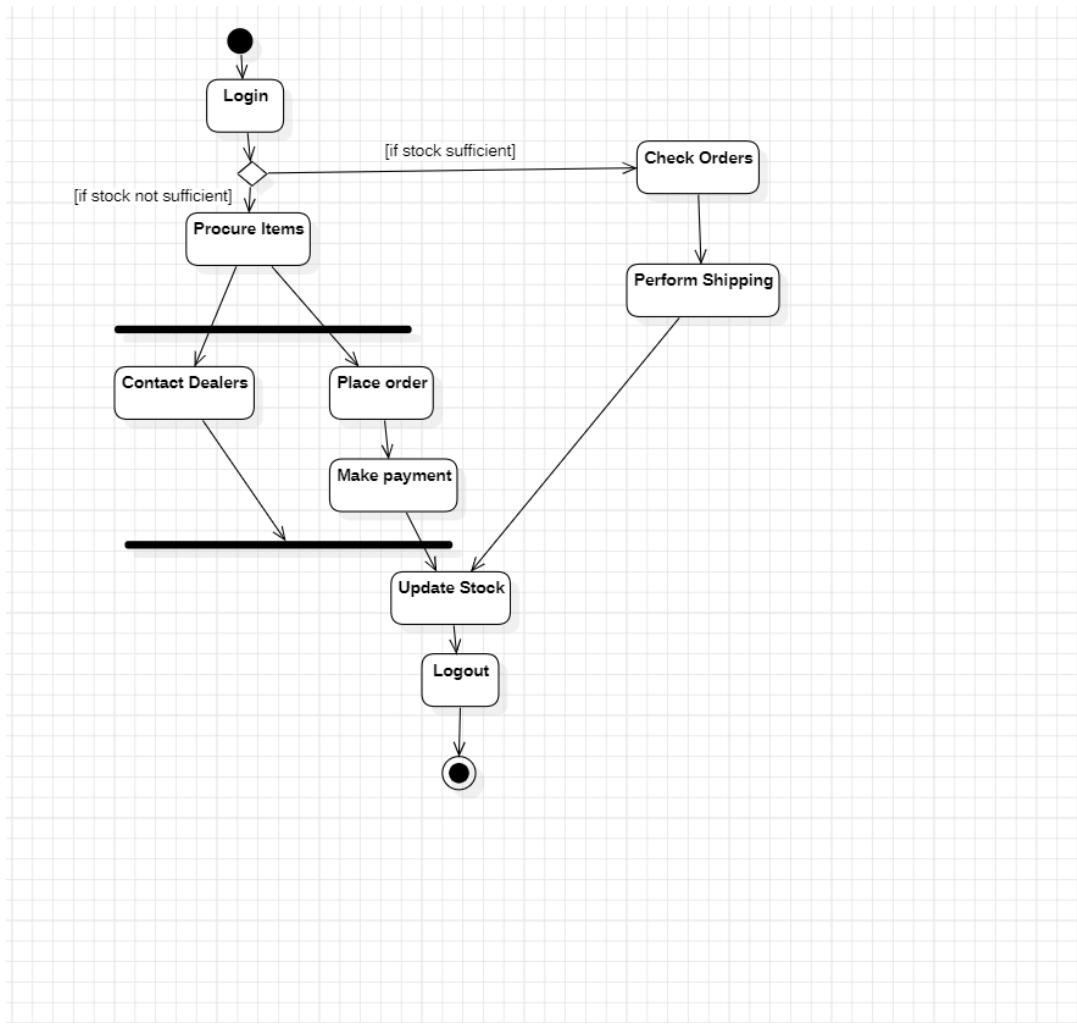
Check Purchase: The admin checks the purchase details to verify the procurement process.

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.

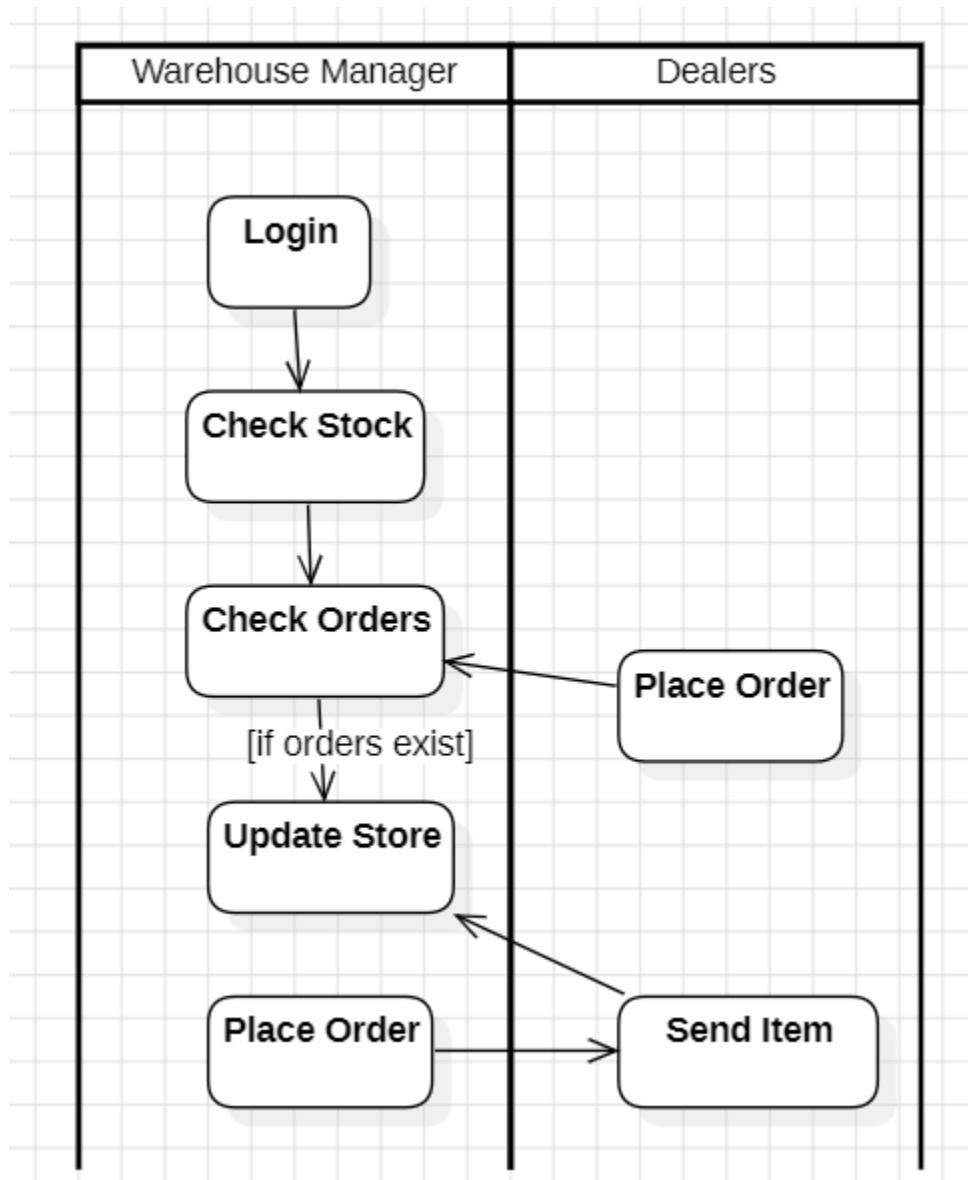
## Activity Diagram



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.

Activity Diagram using Swimlanes:



There are 2 swimlanes Warehouse Manager and Dealer.

## 5. Passport Automation System

### SRS - Software Requirements Specification

#### SRS for Passport Automation System

##### 1. Introduction

1.1 Purpose of this document : This document intends to provide an overview of the requirements, schedule, budget and constraints involved in the development of a passport automation system.

1.2 Scope of this document : This document contains essential details that can be referenced by all stakeholders during the process of development of the software.

1.3 Overview : The product is used to manage the ~~the~~ passport issuing and ~~the~~ verification process.

##### 2 General description

Objective of the user : The user should be able to apply for a passport via a website. He / She should be able to fill a form, submit it the necessary details and also be able to take up an online video verification process.

##### Features :

- A website with a user friendly interface.  
( UI following color ideology )
- A window to register / fill a form to apply for passport.
- A dashboard to track progress of the application
- A video conferencing system ~~to~~ for video meeting verification via video meeting.

## Benefits:

- Allows the users to easily apply for the passport.
- Avoids hassle at the passport verification centre

## 3 Functional requirements

- A landing page showcasing the various options available such as Apply Login, sign up.
- Login / Sign Up window - Login if account already exists using phone number and a verified OTP. If account does not exist create an account using phone number and valid ID proof (Aadhar Number). Then redirect to login.
- Dashboard - A user dashboard on logging in which displays the application status.
- Form - A form containing fields to fill essential details while applying for passport.
- Video meet system - A video meet conferencing option that connects with the concerned authority when the first form is approved.

## 4. Interface Requirements

- User friendly interface that allows users to easily apply for passports.
- Database - ~~communication~~ The details of each user are stored in a database which can be fetched by the authorities hence facilitating user admin communication.

#### Performance requirements

Speed : The data must be loaded at 21 MBps.

Time : Data must be retrieved less than a minute.

#### Design constraints

- Use a NoSQL database

- Use Figma for design

- Comply with government policies required for passport issuing

#### Non functional requirements

- Security : A secure authentication system using phone number and OTP to allow only authorized access. Since the system accepts very confidential data security is of utmost concern.

- Reliability : Very less chances of failure.

- Robust : A backup system containing user data so as to recover back even in case of loss.

#### Preliminary Schedule and Budget

Start date - 12/10/2024

Task	Date of completion	Cost
------	--------------------	------

Requirement Specification	14/10/2024	\$ 5
---------------------------	------------	------

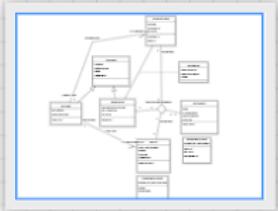
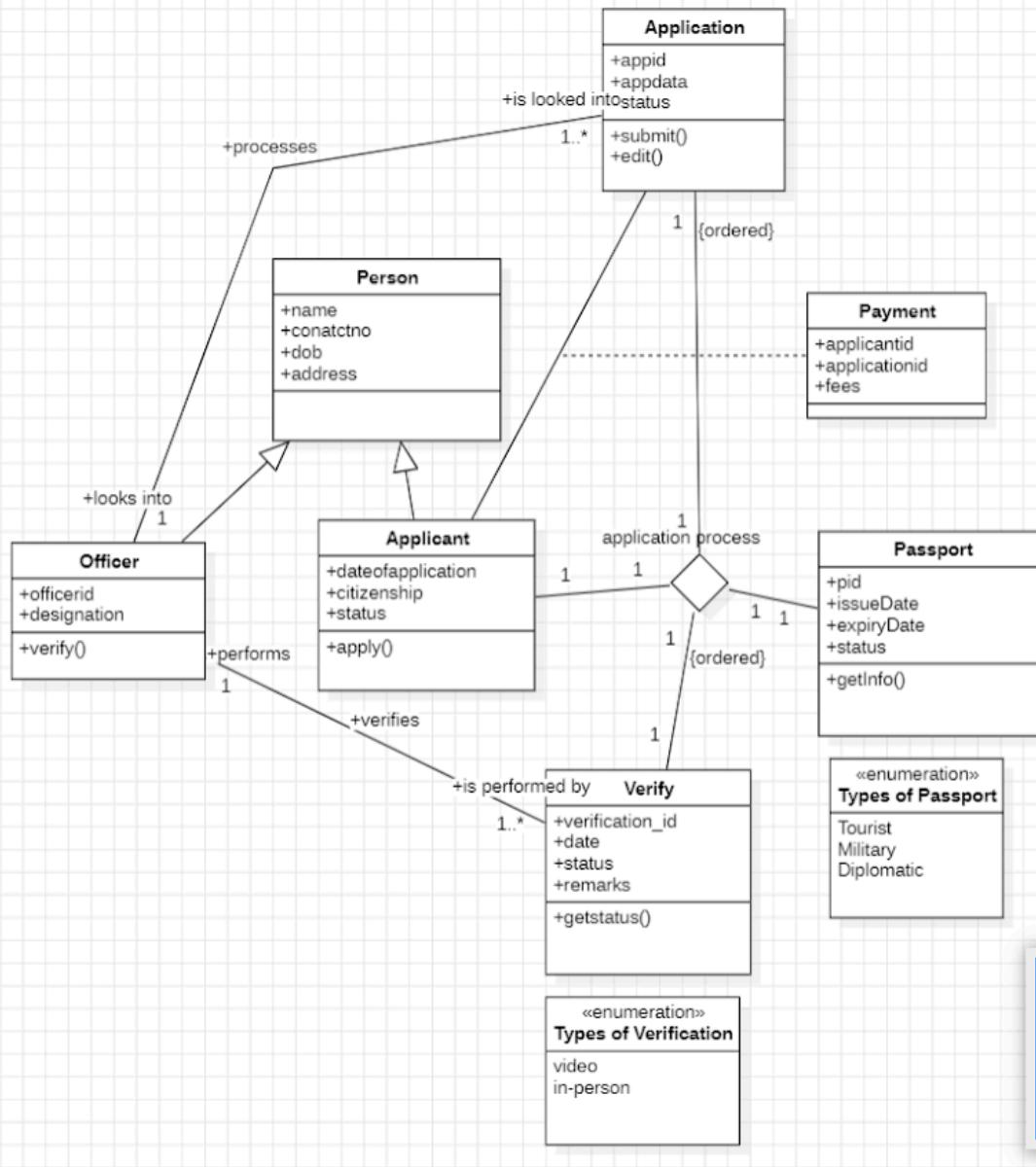
Design	22/10/2024	\$ 10
--------	------------	-------

Development	11/11/2024	\$ 20
-------------	------------	-------

Testing	31/11/2024	\$ 8
---------	------------	------

Total cost		\$ 49
------------	--	-------

## Class Diagram

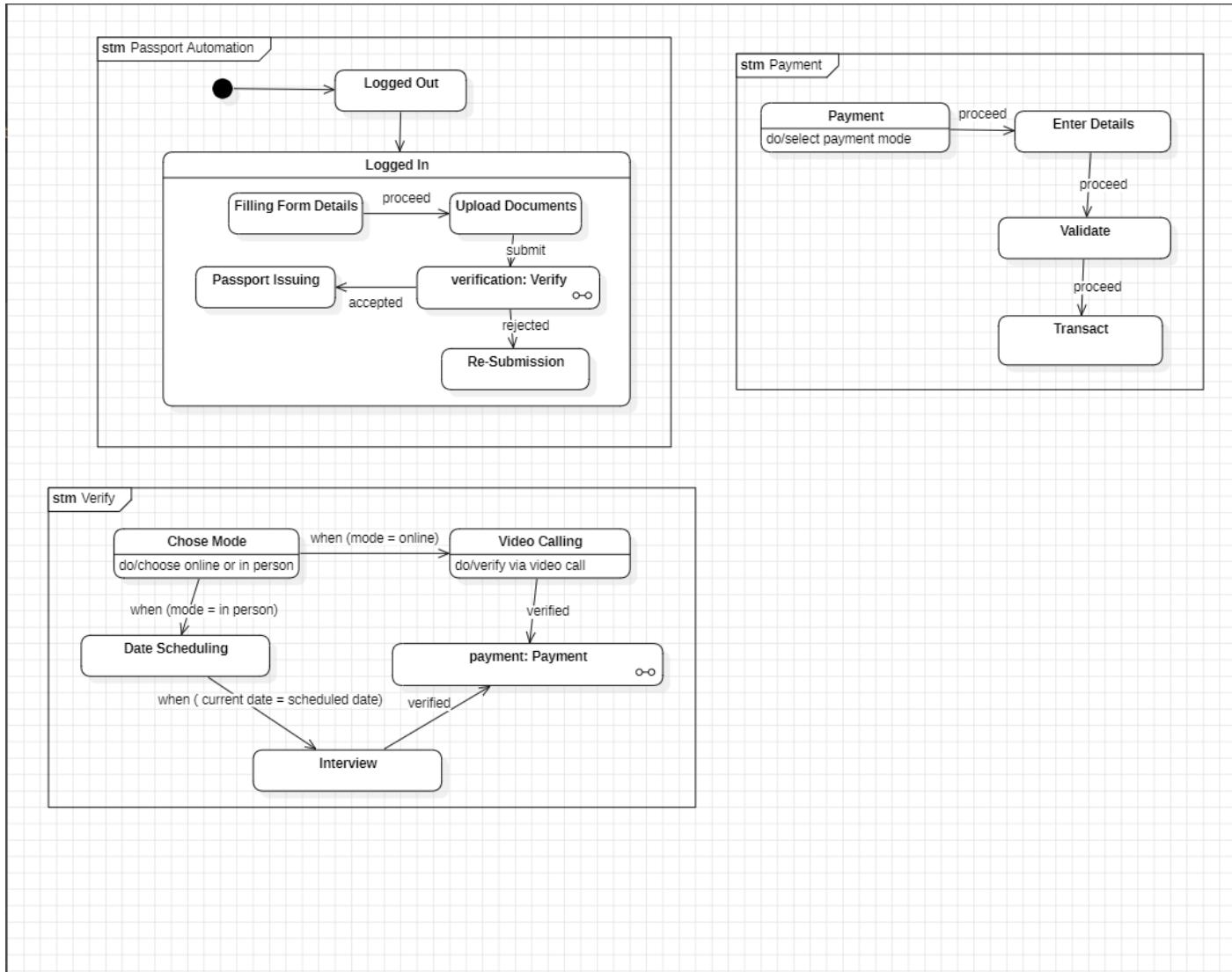


## Description of Class Diagram

- 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

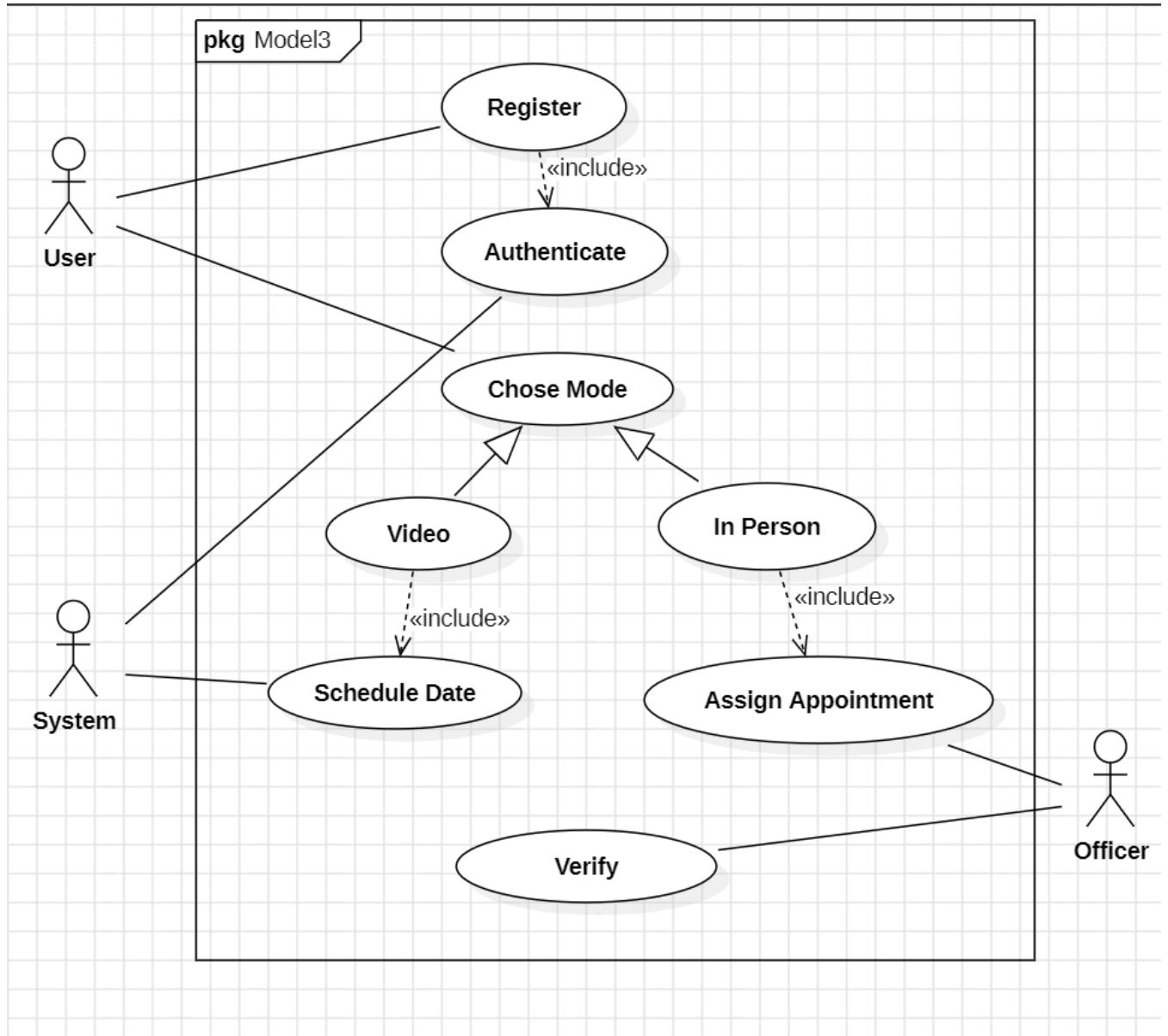


## Description of State Diagram:

- 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



### Description of Use Case Diagram:

Actors Involved: User, System, and Officer.

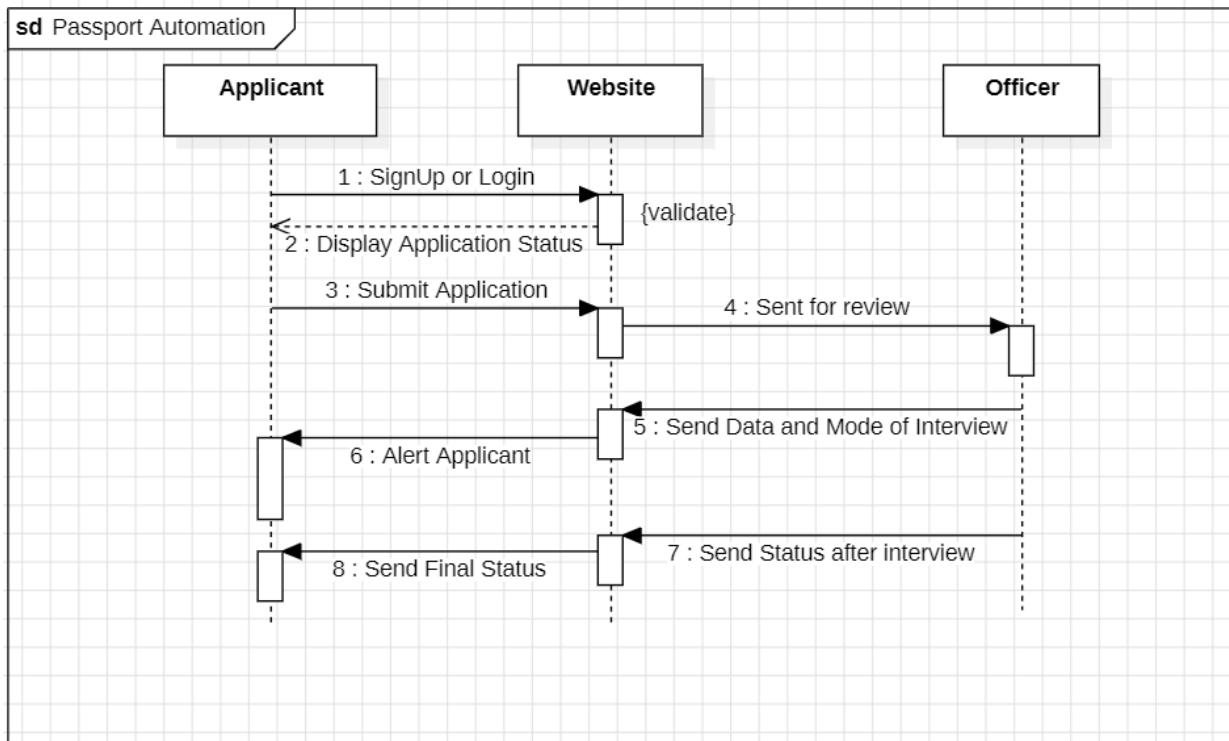
User: Registers and authenticates to access the system. They choose a mode—Video or In-Person.

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



### 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.

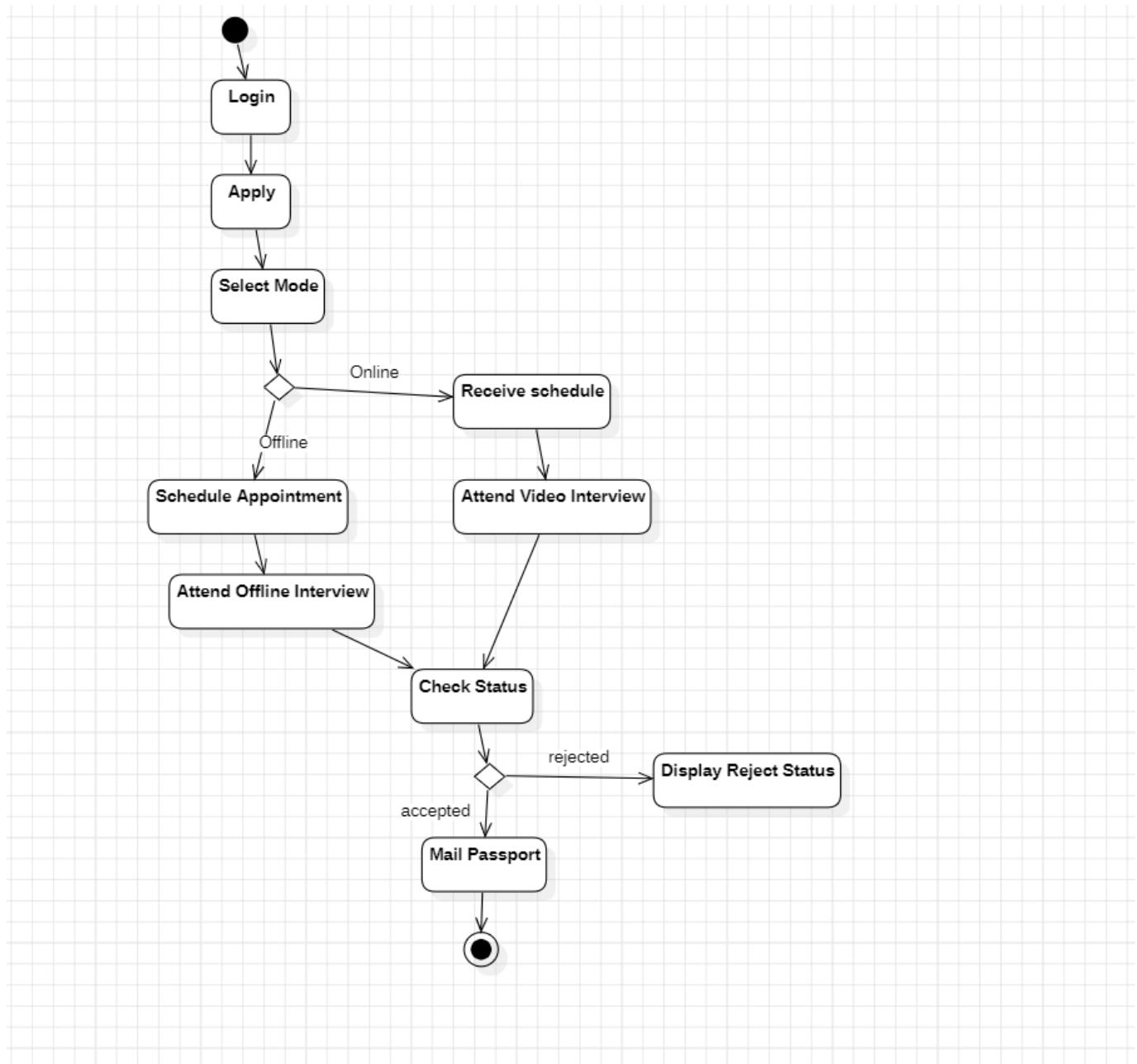
**Send Data and Mode of Interview:** The officer sends back the reviewed data along with the mode and schedule of the interview.

**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.

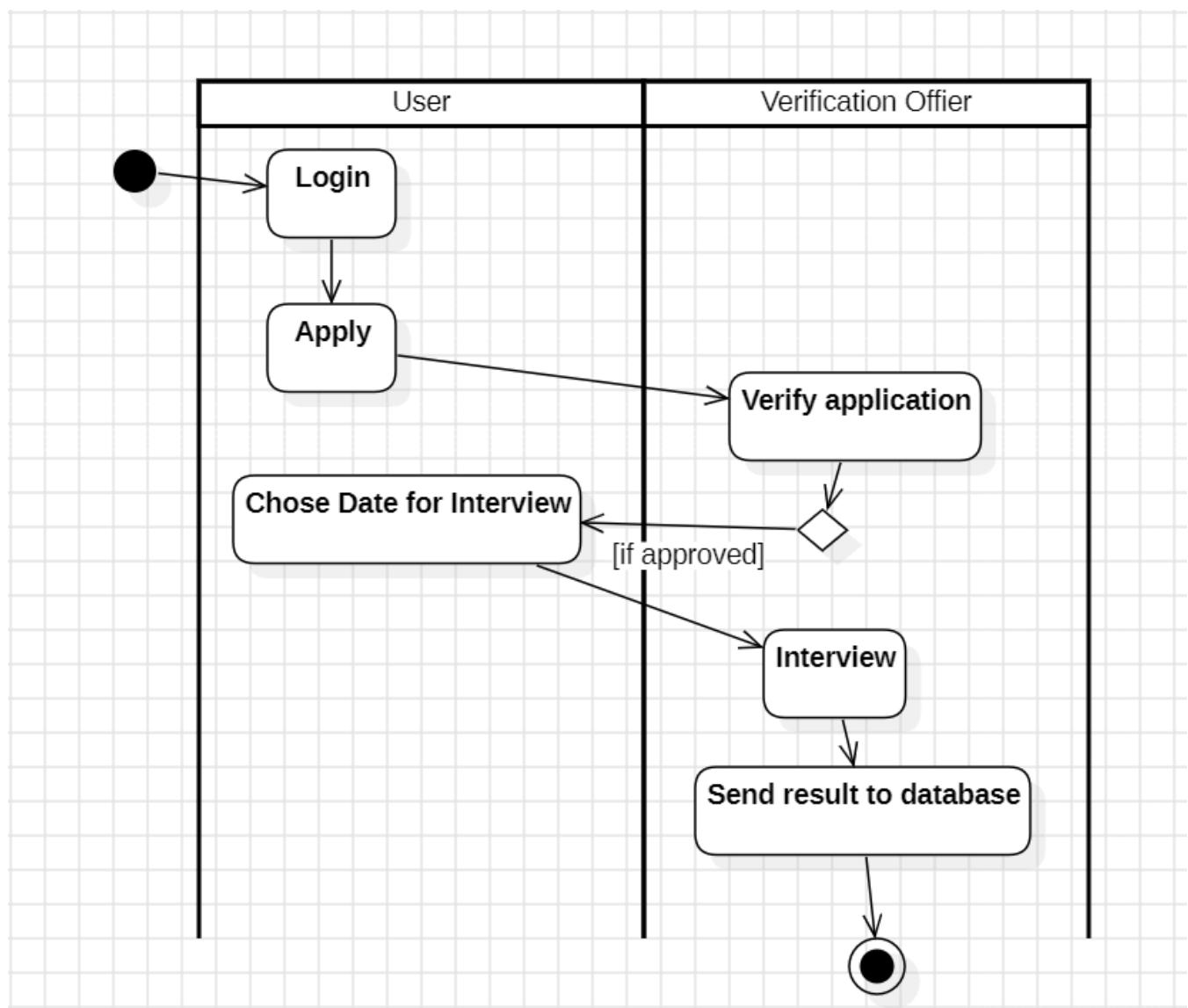
### Activity Diagram



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.

Activity Diagram with Swimlane



The swimlane has 2 lanes namely User and Verification Officer.

