B.M.S. COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

Lab Record

Object-Oriented Modeling

Submitted in partial fulfillment for the 5th Semester Laboratory

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

SHASHANK SP
(1BM22CS256)

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
Oct 2024 - Jan 2025

B.M.S. COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Object Oriented Modelling (23CS5PCOOM) laboratory has been carried out by SHASHANK SP (1BM22CS256) during the 5th Semester Oct 2024 - Jan 2025.

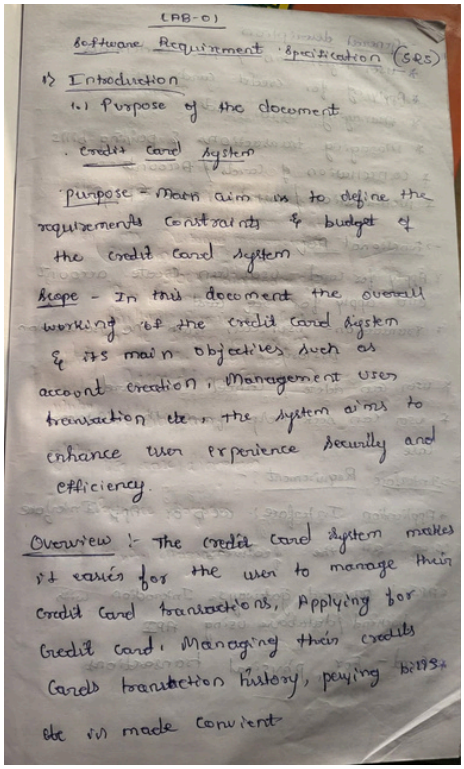
Signature of the Faculty Incharge:
NAME OF THE FACULTY: Prof. Sunayana S
Department of Computer Science and Engineering
B.M.S. College of Engineering, Bangalore

Table of Contents

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

1. HotelManagementSystem

SRS - Software Requirements Specification


(AB-01)

Software Requirement Specification (SRS)

1) Introduction

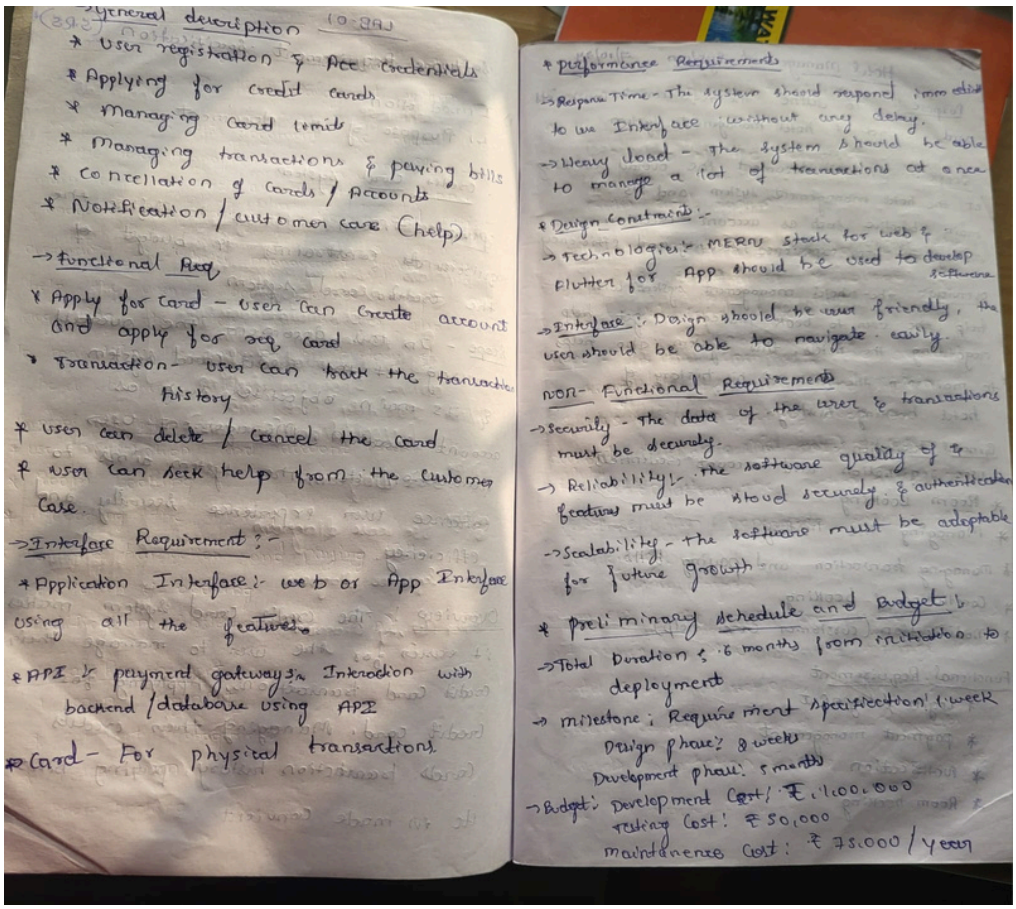
1.1 Purpose of the document

. Credit Card System

Purpose - Main aim is to define the requirements constraints & budget of the credit card system

Scope - In this document the overall working of the credit card system & its main objectives such as account creation, Management user transaction etc, the system aims to enhance user experience security and efficiency.

Overview : The credit card system makes it easier for the user to manage their credit card transactions, Applying for credit card, Managing their credits cards transaction history, paying bills etc is made convient


General description (0-2PJ

* User registration & Acc credentials
* Applying for credit cards
* Managing card limit
* Managing transactions & paying bills
* Cancellation of cards / Accounts
* Notification / customer care (help)

→ functional Req

* Apply for card - User can create account and apply for req card
* Transaction - User can track the transaction history
* User can delete / cancel the card
* User can seek help from the customer care.

→ Interface Requirement :-

* Application Interface : web or App Inkyfore using all the features.
* API : payment gateways, Interaction with backend / database using API
* Card - For physical transactions

* performance Requirements

→ Response Time - The system should respond immediately to user Interface without any delay.

→ Heavy load - The system should be able to manage a lot of transactions at once

* Design Constraints :-

→ Technologies : MERN stack for web & Flutter for App should be used to develop software

→ Interface : Design should be user friendly, the user should be able to navigate easily.

non- Functional Requirements

→ Security - The data of the user & transactions must be securely

→ Reliability : The software quality of the features must be stored securely & authenticated

→ Scalability - The software must be adaptable for future growth

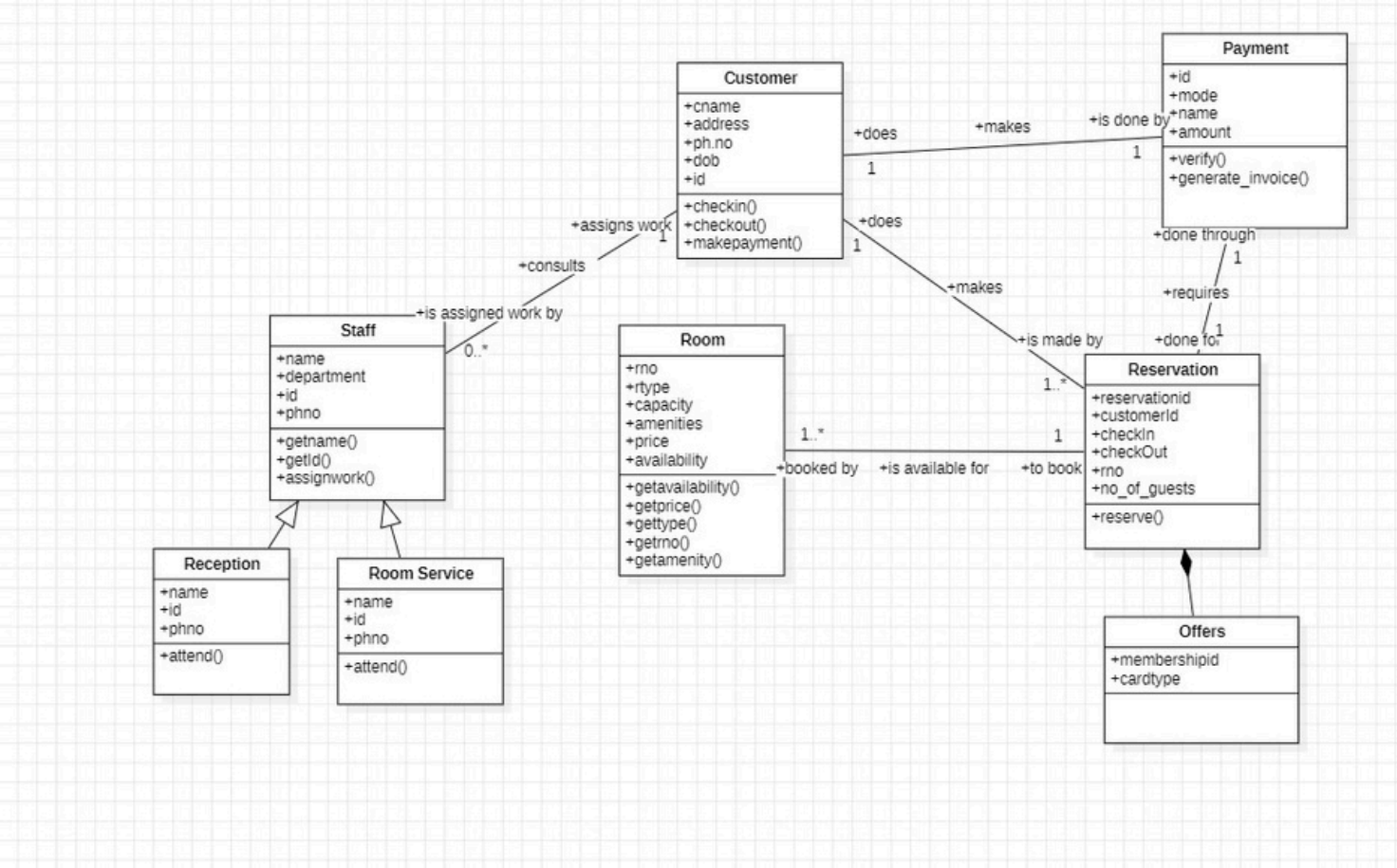
* preliminary schedule and Budget :-

→ Total Duration : 6 months from initiation to deployment

→ milestone : Requirement Specification : 1 week
Design phase : 2 weeks
Development phase : 3 months

→ Budget : Development Cost : ₹ 1,100,000
Testing Cost : ₹ 50,000
maintenance Cost : ₹ 75,000 / year

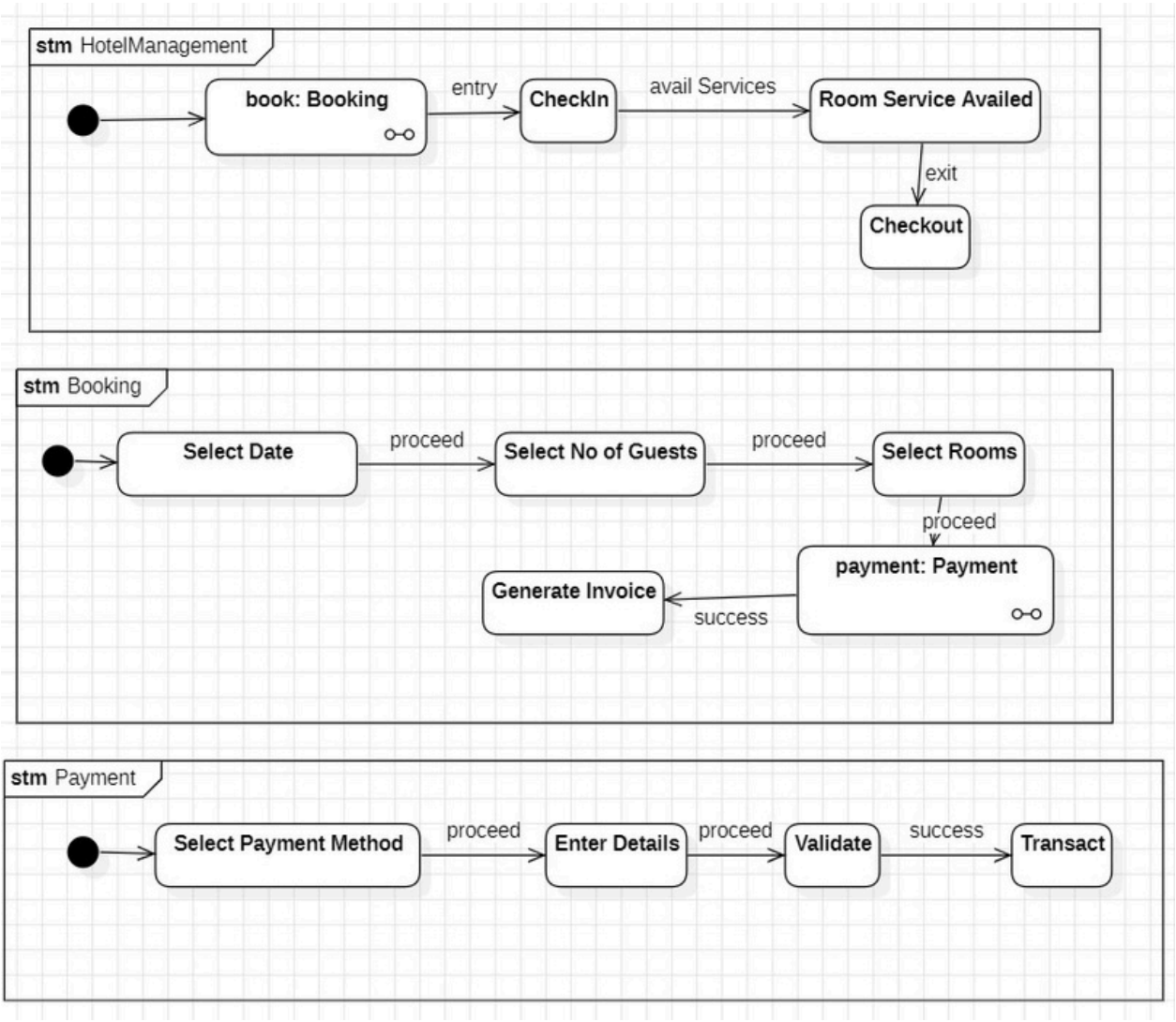
Class Diagram



Description:

- Customer-Classthathasattributesandfunctionsindicativeofthecustomerwho books the hotel.
- Room-Indicatestheroomthatcanbebooked.
- Payment-Aclassthathasattributesandfunctionsforatransaction.
- Staff-Indicativeofhotelstaff.
- Reservation - Contains attributes and functions aiding the process of reservation.
- Generalization-TwoclassesReceptionandRoomServiceinheritpropertiesoftheStaff class.
- Composition-TheOffersclassisacompositionofReservationi.eifreservationceases to exist this class does not exist.

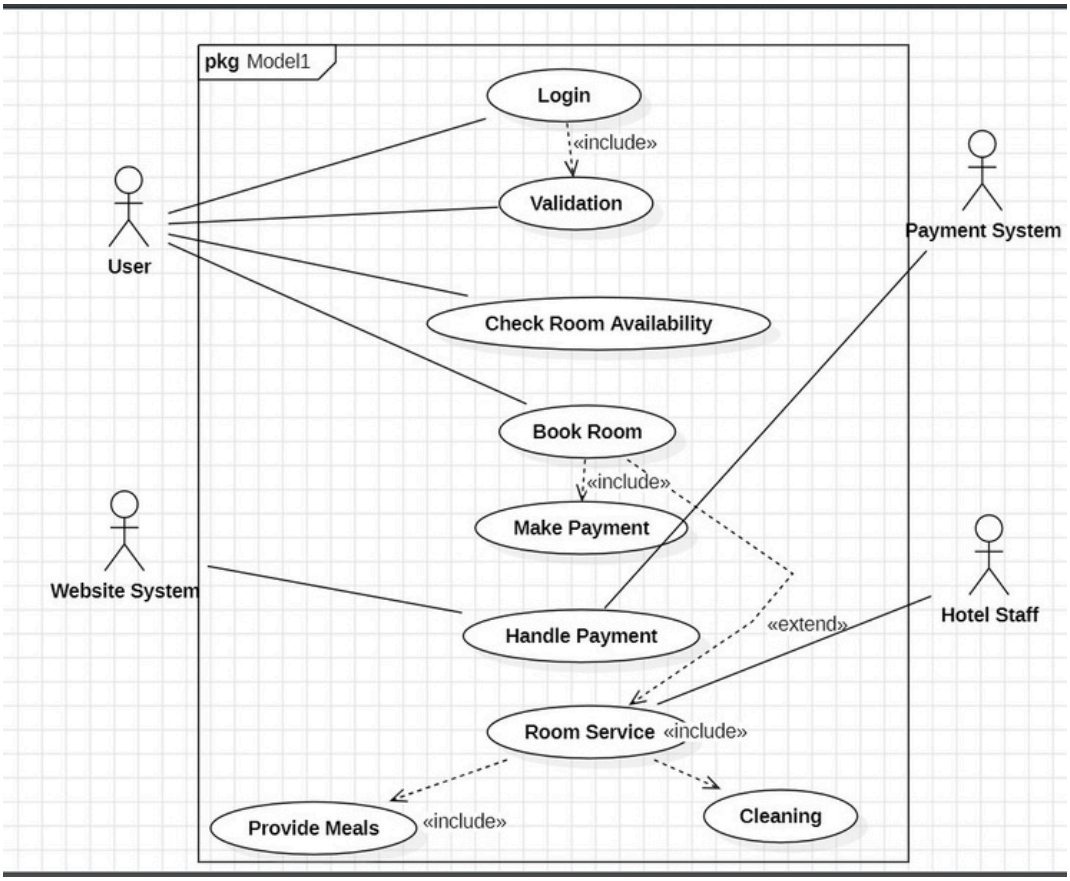
State Diagram



Description:

- ThereisamainHotelManagementstatediagramandtwosubmachinestates called Booking and Payment.
- HotelManagement:ContainsstatessuchasCheckin-oncetheguestenters, Room Service Availed - if the guest opts for it and CheckOut on exit
- Booking:Thissubmachinestatehasstatestofacilitatebookingprocessi.eSelectDate, Select no of guests, rooms, a submachine called payment and a state to generate invoice.
- Payment:Thissubmachinehasstatesforpaymentpurposesuchas EnterDetails,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:

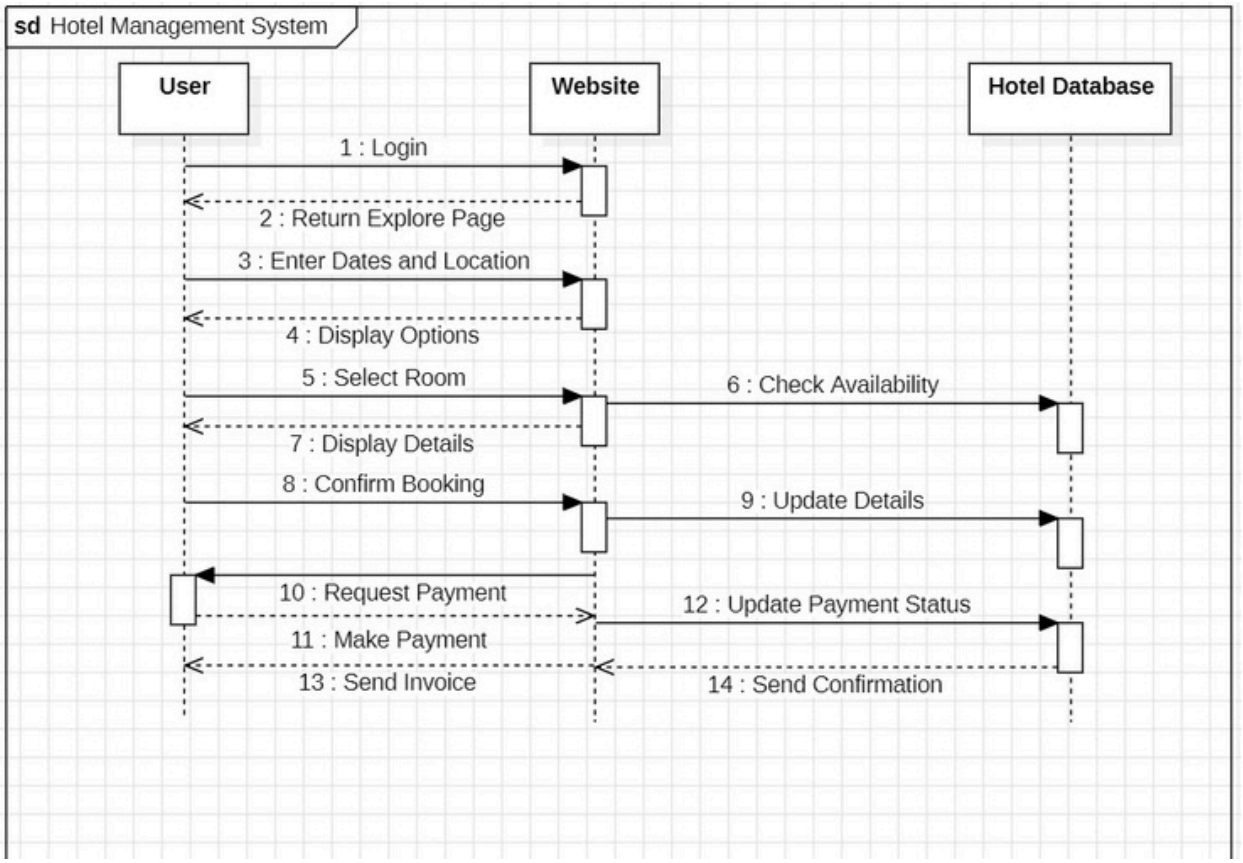
Validation - Validates the user from its database.

Handle payment - Also handles payment process.

Hotel Staff:

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

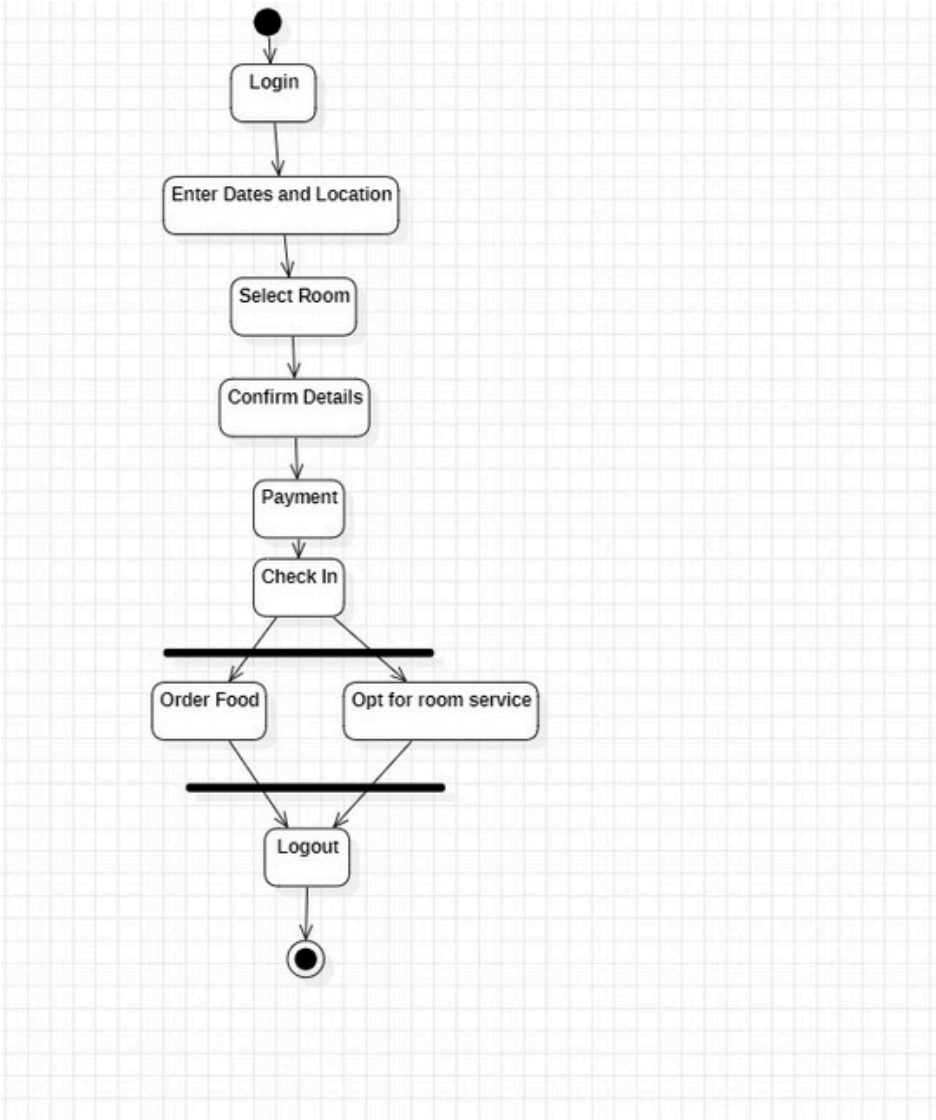
Sequence Diagram



Description:

- Theuserlogsintothewebsite.Iftheuserisvalidthentheexplorepageisreturned.
- Theuserentersthedateandlocation.Thewebsitedisplaystheavailablehoteloptions.
- Oncetheuserselectstheroomoftheirchoicethewebsiteenquiresthehoteldatabase about its availability.
- Thewebsitedisplaysthedetails.Theuserconfirmstheirbooking,theuser'sdetailsare updated in the hotel database.
- Thereisapaymentrequestinitiatedbythewebsite,theusermakesthepaymentand this is updated in the hotel database.
- Onconfirmationfromhoteldatabaseaninvoiceissenttotheuser.

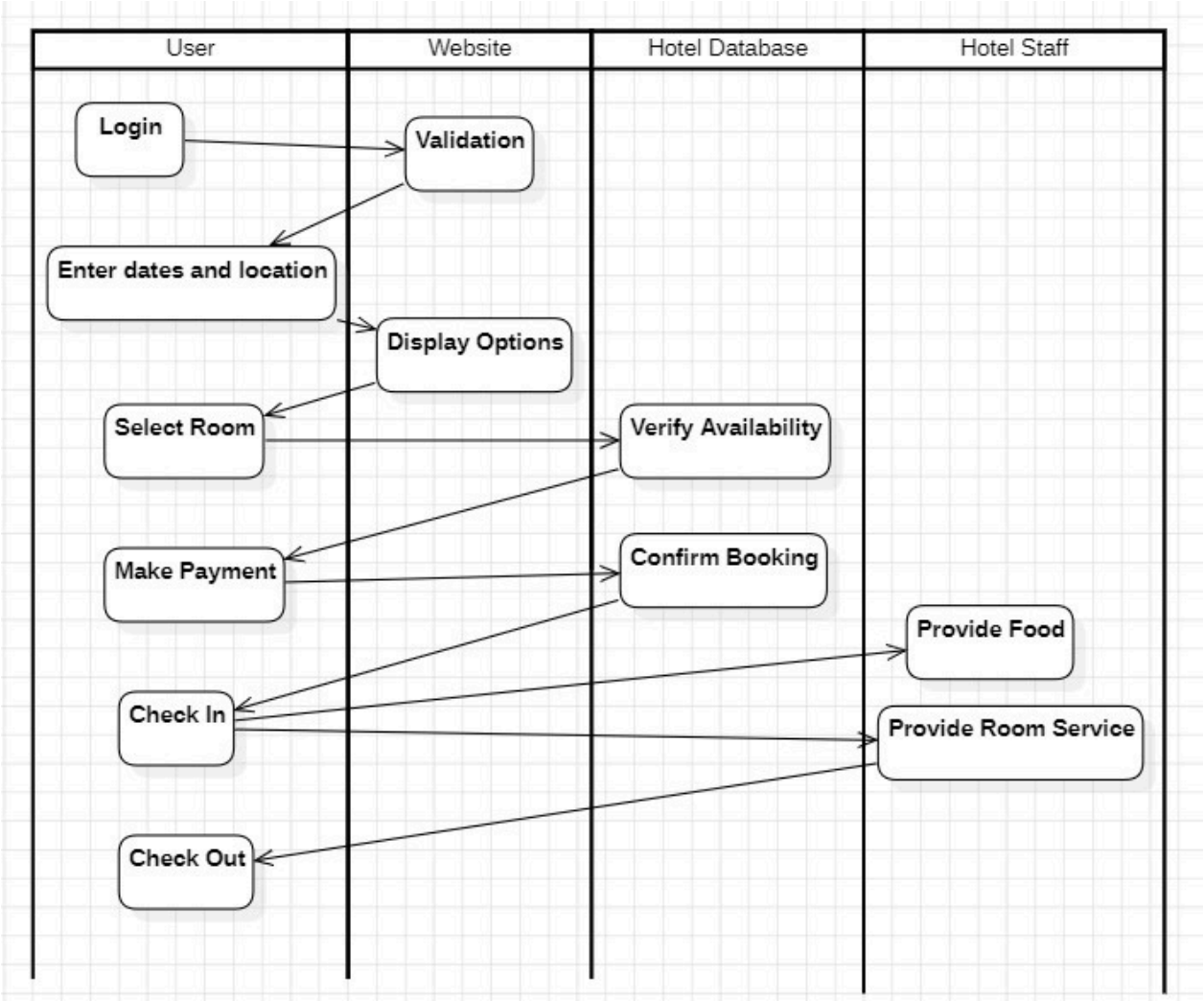
Activity Diagram



Description:

- Theuserlogsin,entersthedateandlocationinthewebsite.
- Oncethewebsitedisplaystheavailableoptionstheyselectaroom,confirmdetailsand make payment.
- Postcheckintheymayorderfoodandoptforroomservicesimultaneously.
- Theuserchecksout.

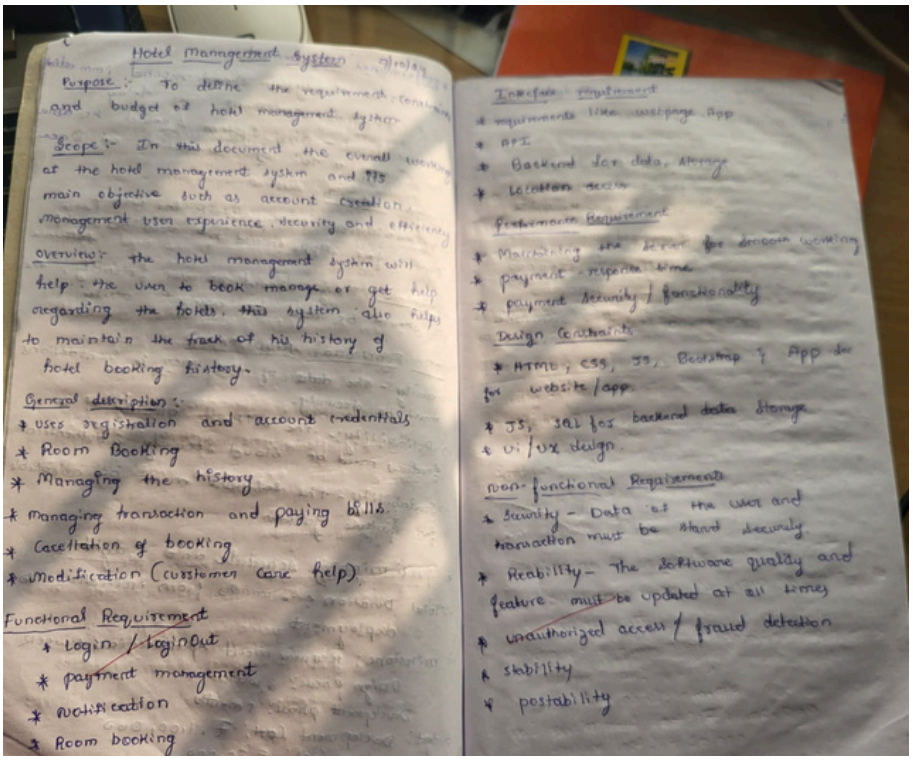
Activity Diagram using Swimlane

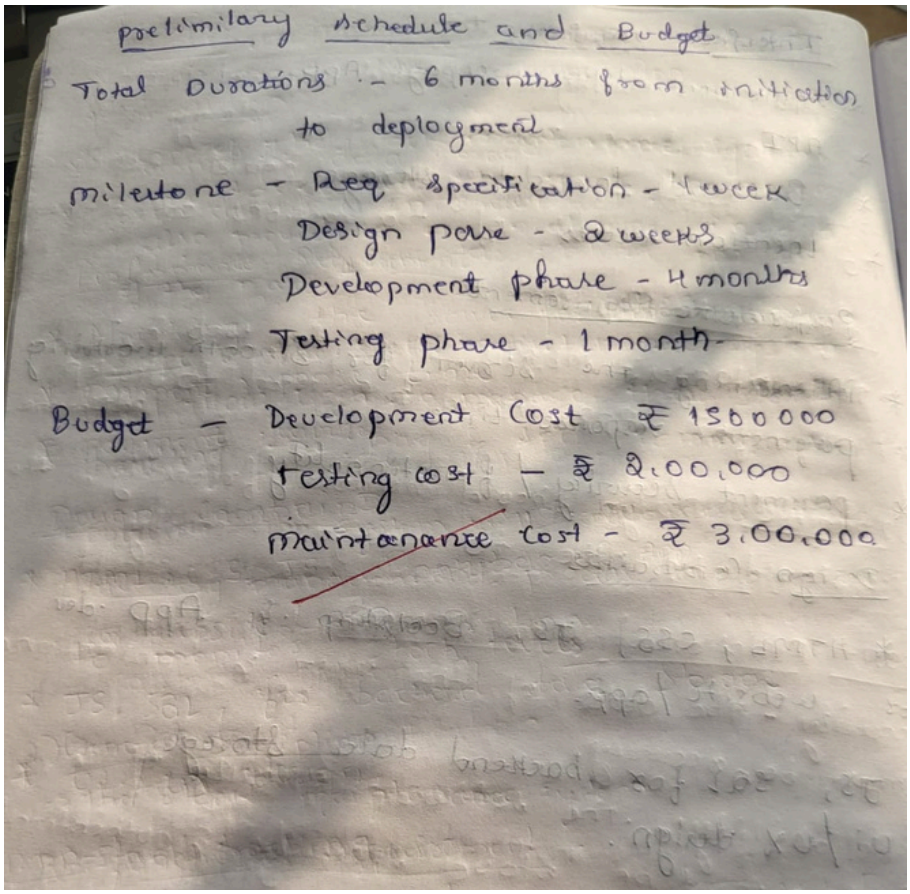


The diagram includes 4 swimlanes namely - User, Website, Hotel Database and Hotel Staff.

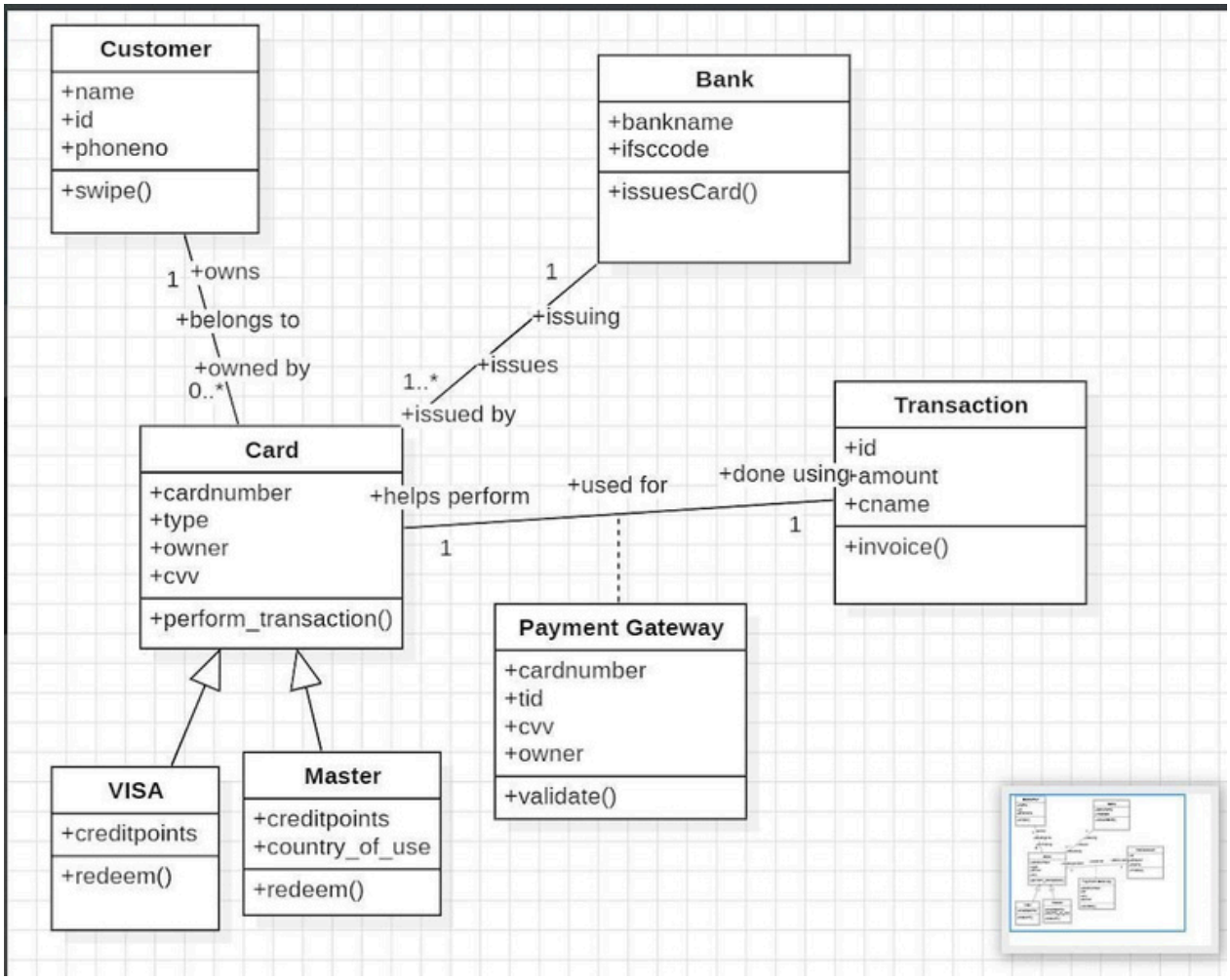
2. Credit Card Processing System

SRS - Software Requirements Specification





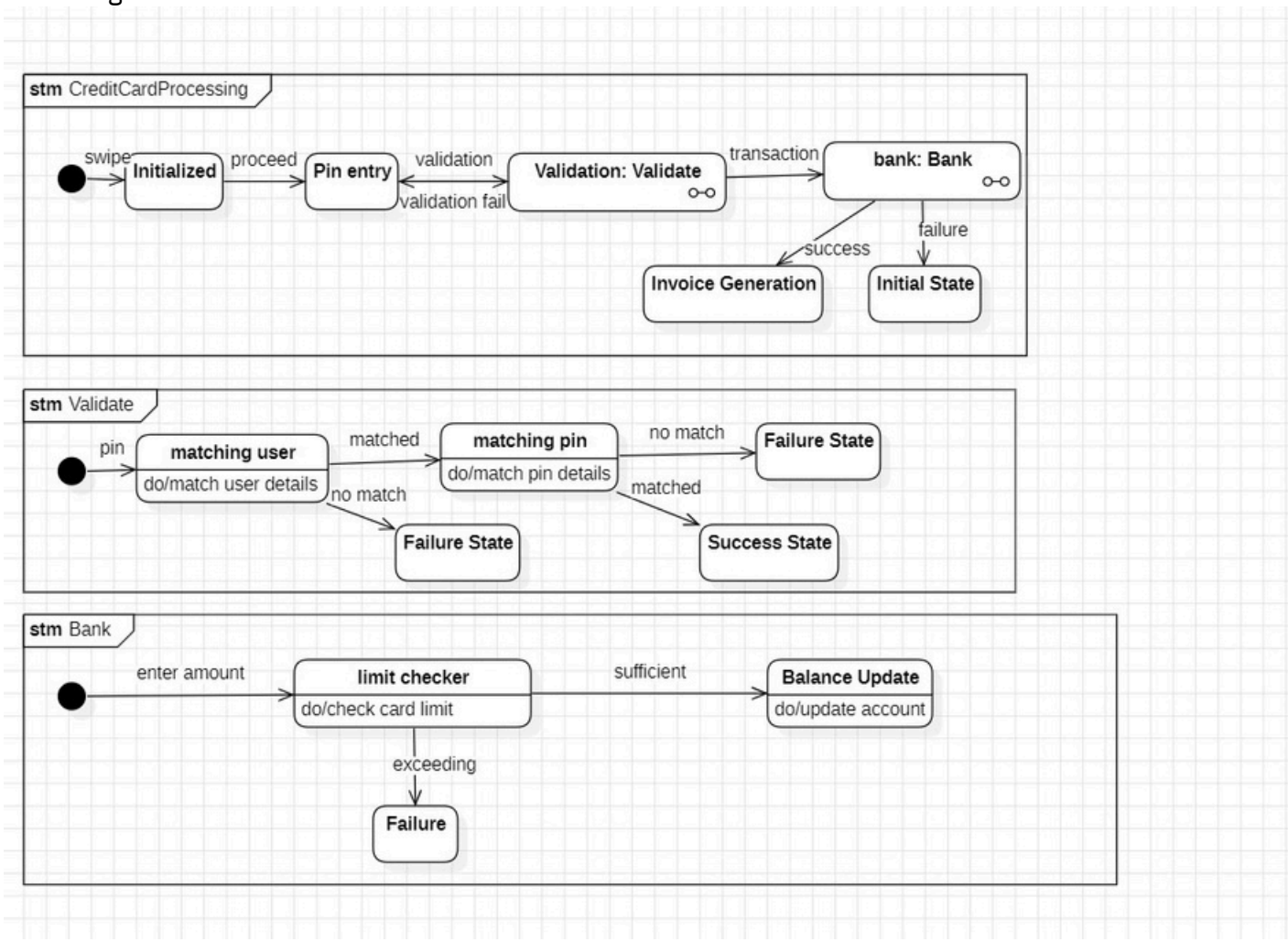
Class Diagram



Description:

- Customer-Classindicativeofcustomerusingthecard.
- Bank-Indicativeofthebankofthecreditcard.
- Card-Indicativeofthecardusedbythecustomer.
- Transaction-Indicativeofthetransactionthatoccursonusingcreditcard.
- Generalization-VISA,MasterinheritpropertiesfromCard.
- AssociationClass-PaymentGatewayistheassociationclassassociatingCard and Transaction.

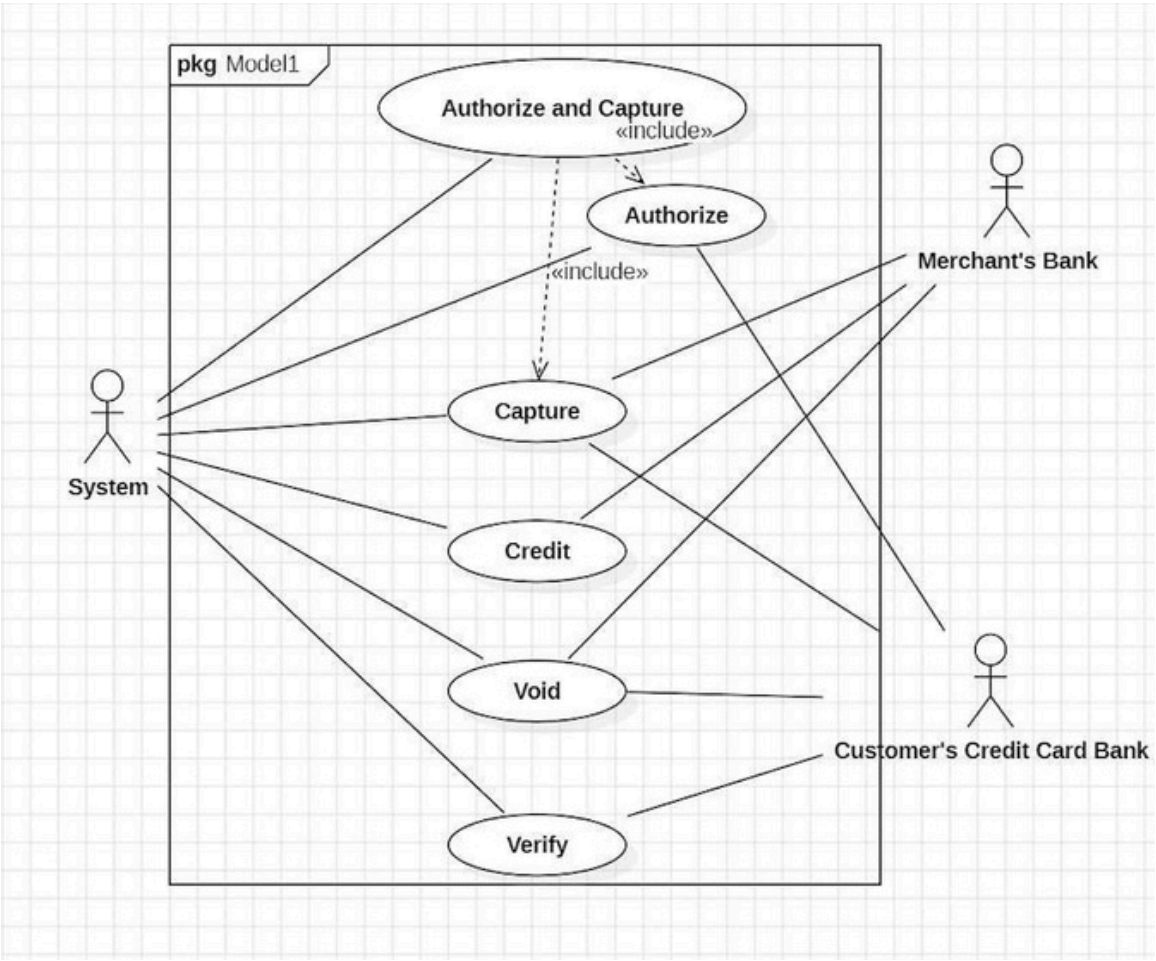
State Diagram



Description:

- The state diagram represents a CreditCardProcessing system with two submachines, ValidateandBank.
- ThemainsystembeginswiththeInitializedstate(cardswipe)andtransitionsto Pin Entry, followed by validation via the Validate submachine.
- Ifsuccessful,itproceedstotheBanksubmachinefortransactionprocessing.TheValidate submachine handles user and PIN matching, leading to success or failure.
- TheBanksubmachinecheckstransactionlimitsandupdatesbalances,withtransitions 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.

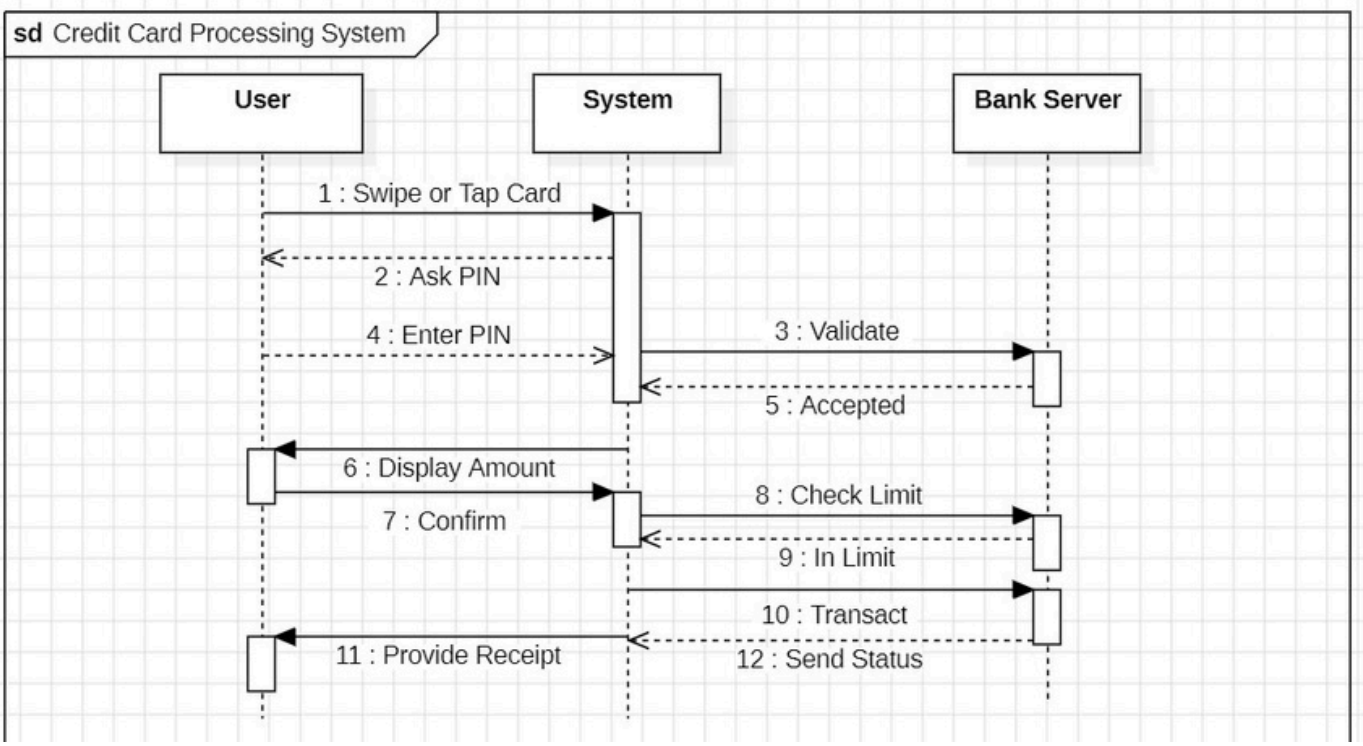
System: Initiates and manages multiple use cases, including:

- AuthorizeandCapture:AcompositeprocessthatincludesAuthorizeandCapture operations.
- Authorize:ValidatesthepaymentwiththeCustomer'sCreditCardBank.
- Capture:Completesthepaymentprocessbyconfirmingthetransactionwiththe Merchant's Bank.
- Credit:Handlesrefundingorcreditingthecustomer'saccount.
- Void:Cancelsanauthorizationortransaction.
- Verify:Verifiesthecardholder'sinformation.

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.

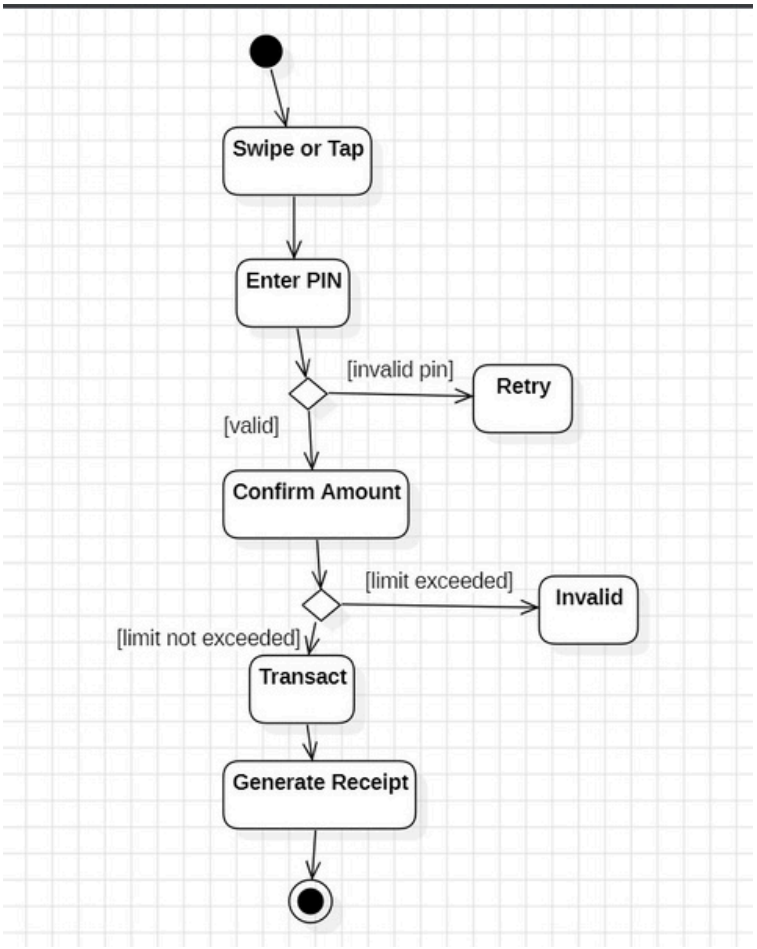
Sequence Diagram



Description:

- Theuserswipesortapstheircard.
- Thesystemaskstheusertoenterthe PIN.
- TheuserentersthePIN,andthesystemvalidatesitwiththebankserver.
- Thebankserverrespondswithacceptanceifthe PINisvalid.
- Thesystemdisplaysthetransactionamounttotheuser.
- Theuserconfirmstheamount.
- Thesystemchecksthecreditlimitwiththebankserver.
- Ifthelimitissufficient,thebankserverprocessesthetransaction.
- Thetransactionstatusissentbacktothesystem.
- Thesystemprovidesareceipttotheuser.

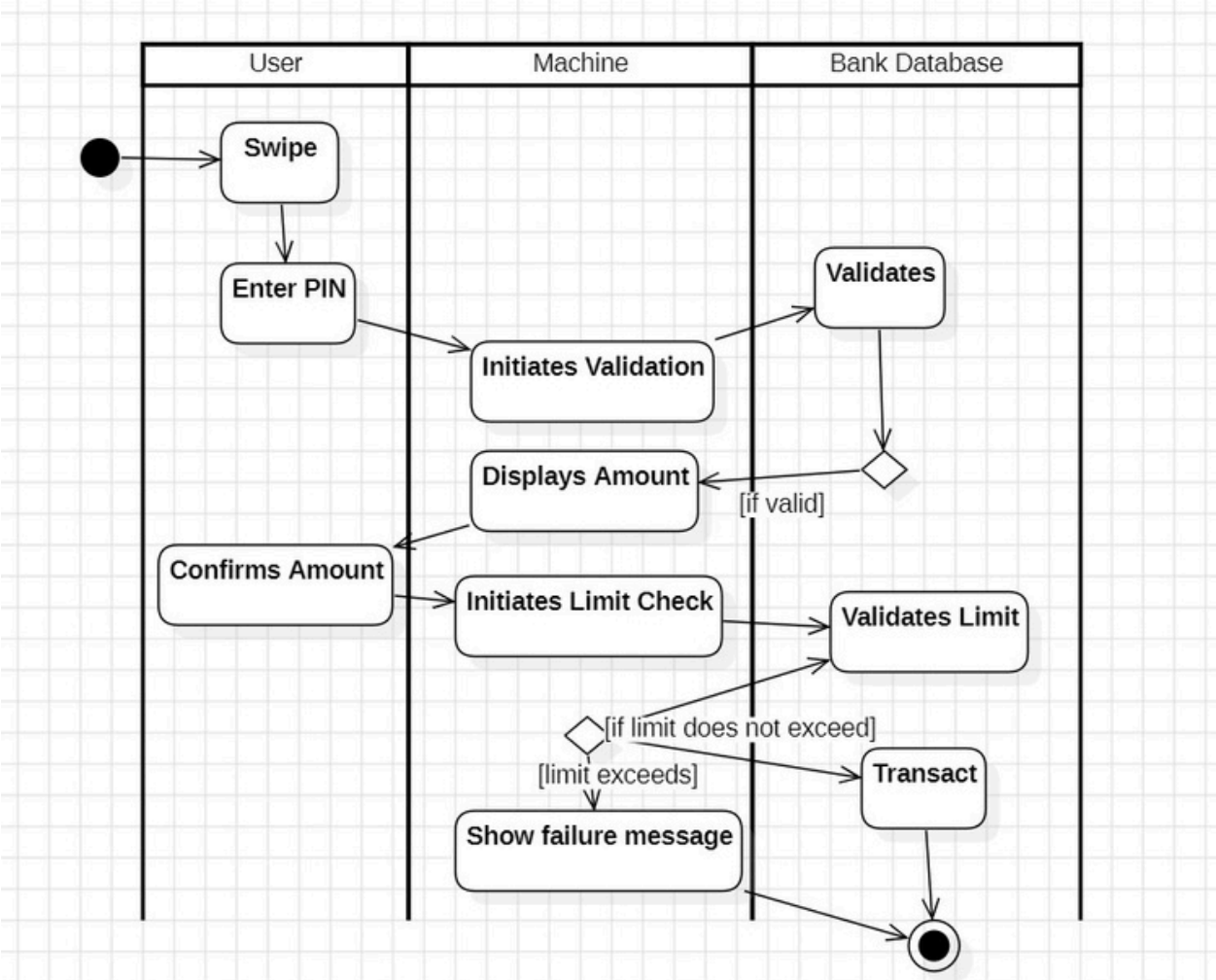
Activity Diagram



Description:

- Theuserswipesortapsthecard.
- TheyenterthePIN.IfthePINisrighttheuserispromptedtoconfirmiftheamount displayed is right.
- Furtherifthelimithasnotbeenreachedthenthetransactionhappensandareceiptis generated else the transaction is invalid.

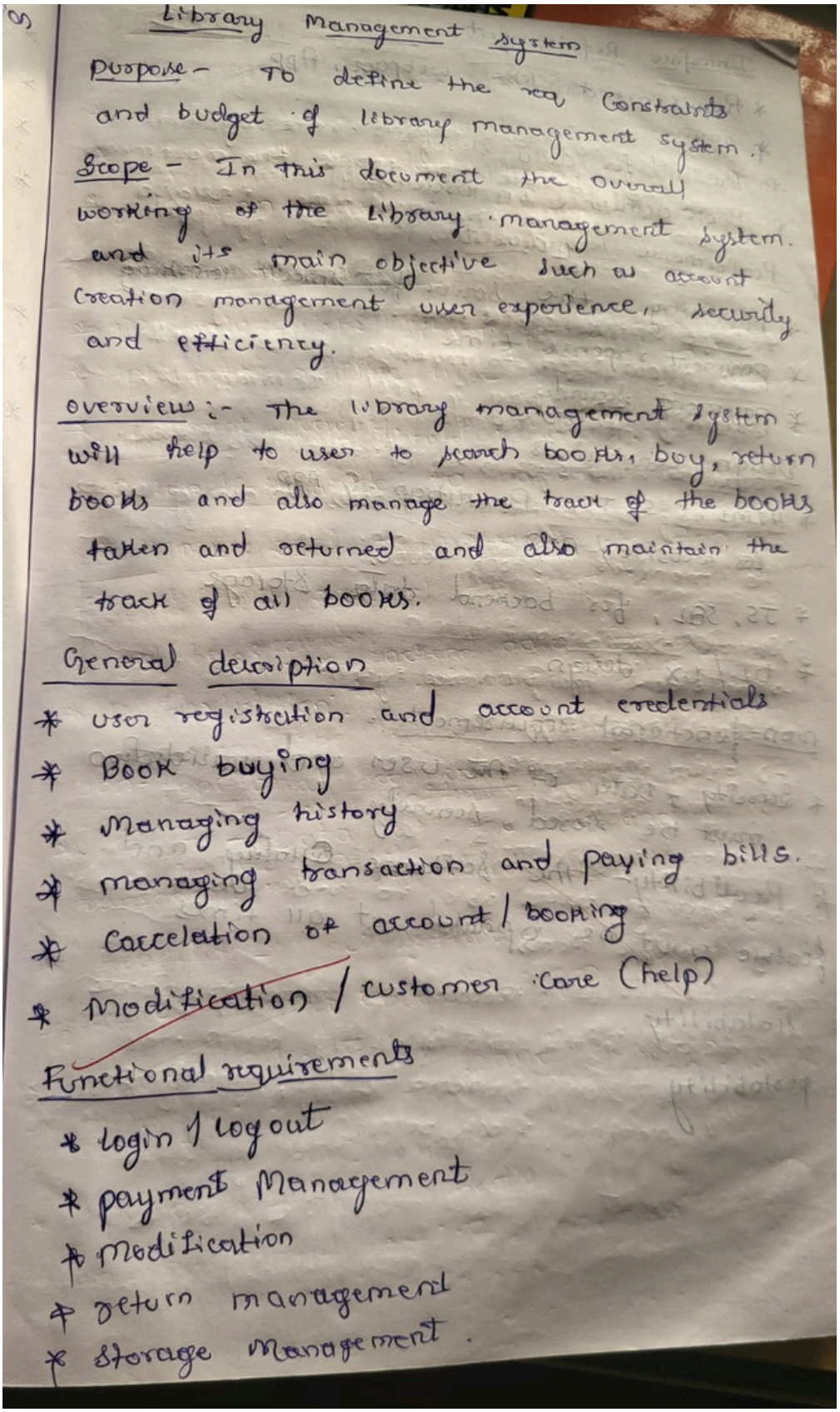
Activity Diagram using Swimlane



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

3. Library Management System

SRS - Software Requirements Specification



Library Management System

Purpose - To define the req Constraints and budget of library management system.

Scope - In this document the overall working of the Library management system. and its main objective such as account creation management user experience, security and efficiency.

Overview :- The library management system will help to user to search books, buy, return books and also manage the track of the books taken and returned and also maintain the track of all books.

General description

* User registration and account credentials
* Book buying
* Managing history
* managing transaction and paying bills.
* Cancelation of account / booking
* Modification / customer care (help)

Functional requirements

* login / logout
* payment Management
* Modification
* return management
* Storage Management.

Interface Requirement

* Requirement like webpage, App
* API
* Backend for data storage
* Book details (In order).

Performance Requirement

* Maintaing the sever for smooth working
* payment reponse time
* payment security.

Design Constraints

* HTML, CSS, JS, Bootstrap - for App for website / App
* JS, SQL, for backend data storage
* UI / UX design.

non-functional requirement :-

* Security - Data of the user and validation must be stored securely
* Reliability - the software Quality and feature must be updated at all times
* Scalability
* portability.

preliminary schedule and Budget

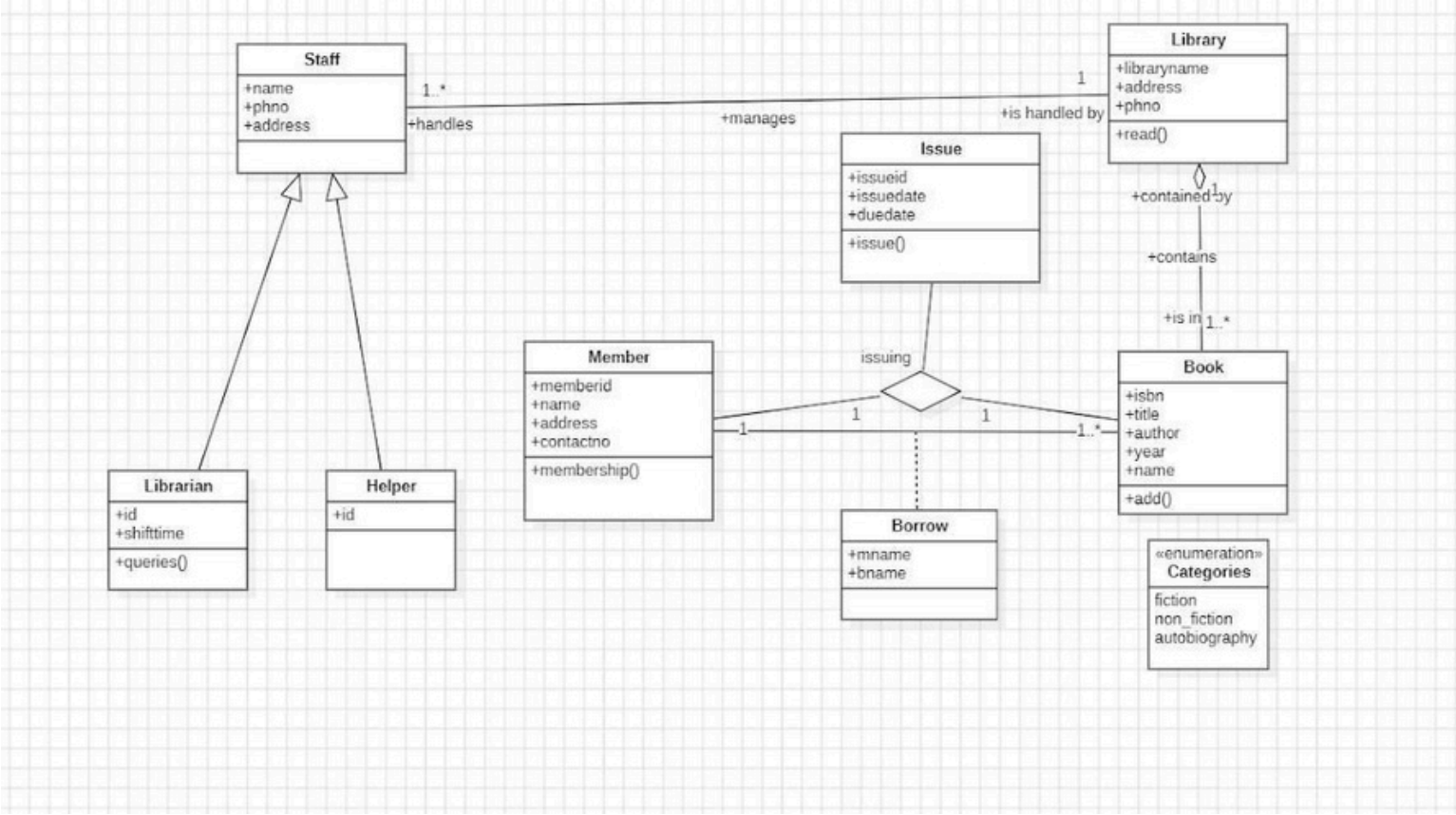
Total Duration - 3 months

milestone - Req specification - 1 week
Design phase - 1 week
Development phase - 2 months
Tuting - 2 weeks

Budget - Development Cost - ₹ 1,00,000
Testing cost - ₹ 25,000
maintanance cost - ₹ 50,000

7/10/2024

Class Diagram

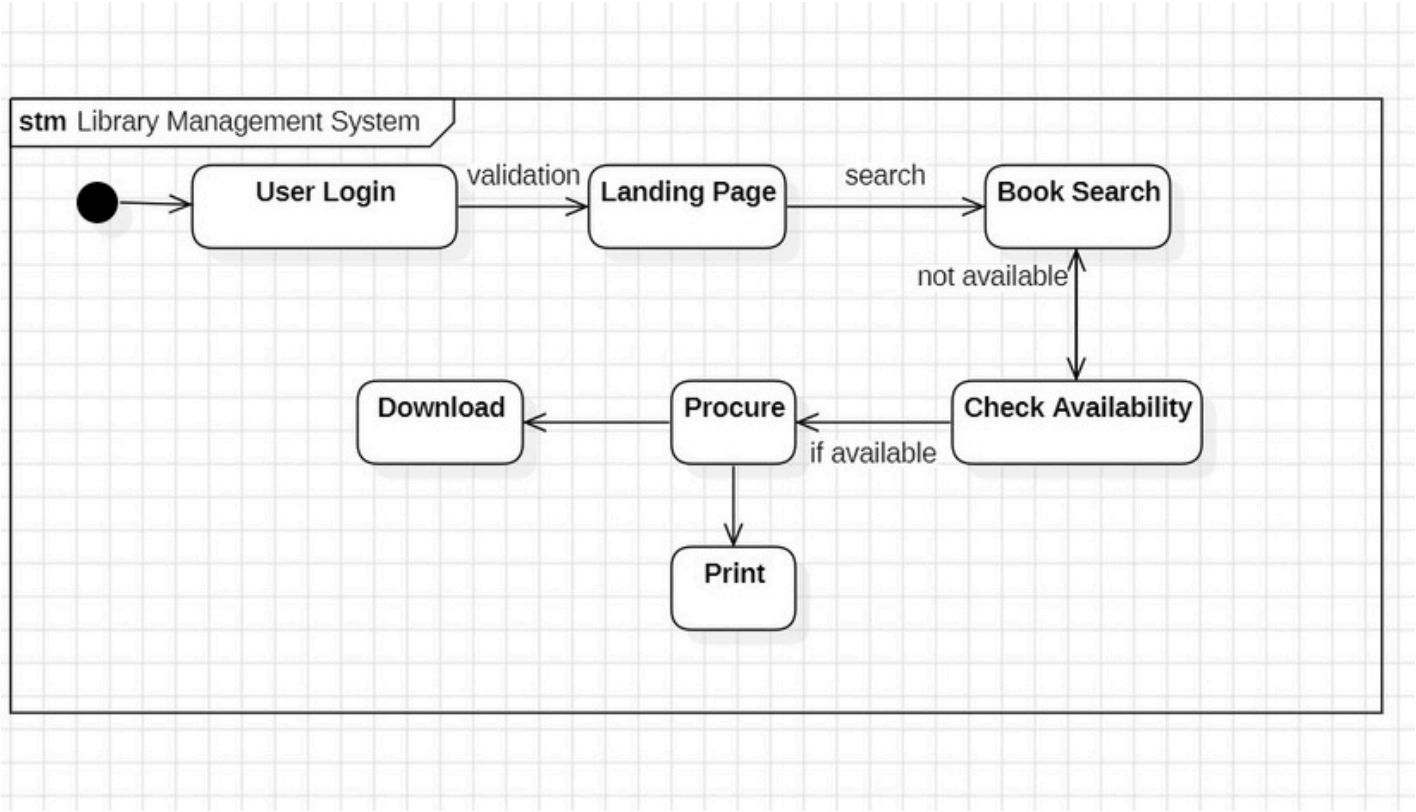


Description:

- StaffClass:Representslibrarystaffwithattributessuchasname,phonenumber,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.
- BookClass:RepresentsbookswithattributessuchasISBN,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.
- MemberClass:RepresentslibrarymemberswithattributeslikememberID,name, address, and contact number. Members are associated with the Borrow class, which connects them to books.
- IssueClass:Managestheissuingofbookswithattributeslikeissue ID,issuedate,and due date. It acts as an intermediary between Member and Book, forming an association.
- BorrowClass:Representstheactofborrowing,connectingMemberandBook. 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).

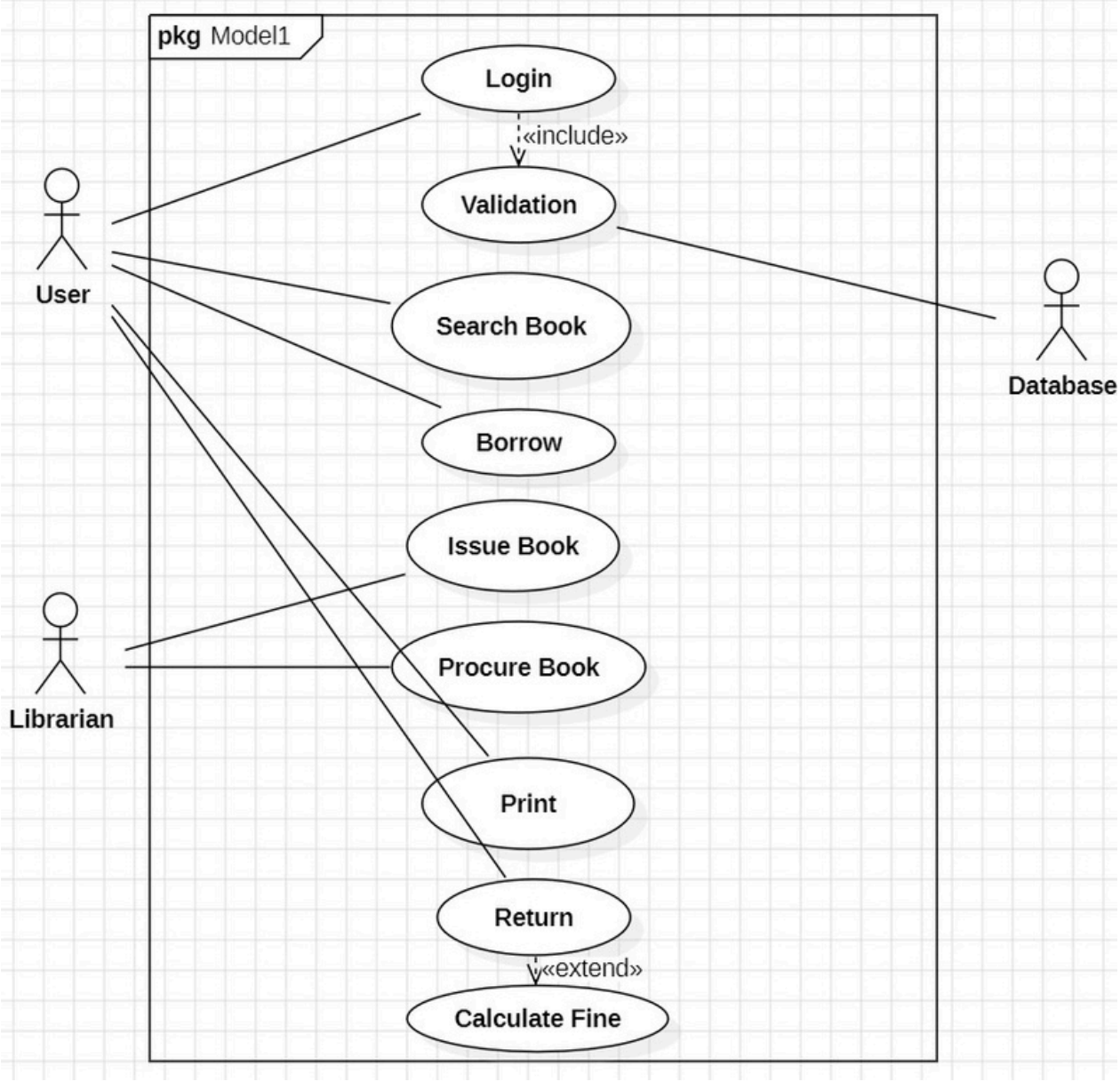
State Diagram



Description:

- ThediagramrepresentsaLibraryManagementSystemwithstatesforuserinteraction. The process starts at User Login, followed by validation and transition to the Landing Page.
- UserscansearchforbooksviatheBookSearchstate,whichleadstoCheckAvailability. If the book is available, it transitions to Procure for obtaining the book, followed by options to Print or Download.
- Ifthebookisnotavailable,itloopsbacktoBookSearchforfurthersearches.

Use Case Diagram



Description:

Actors Involved: User, Librarian, and Database.
User:

- Login:Theuserlogsintothesystem,whichincludesValidationoftheircredentials against the Database.
- SearchBook:Allowstheusertosearchforbooksinthelibrary'sdatabase.

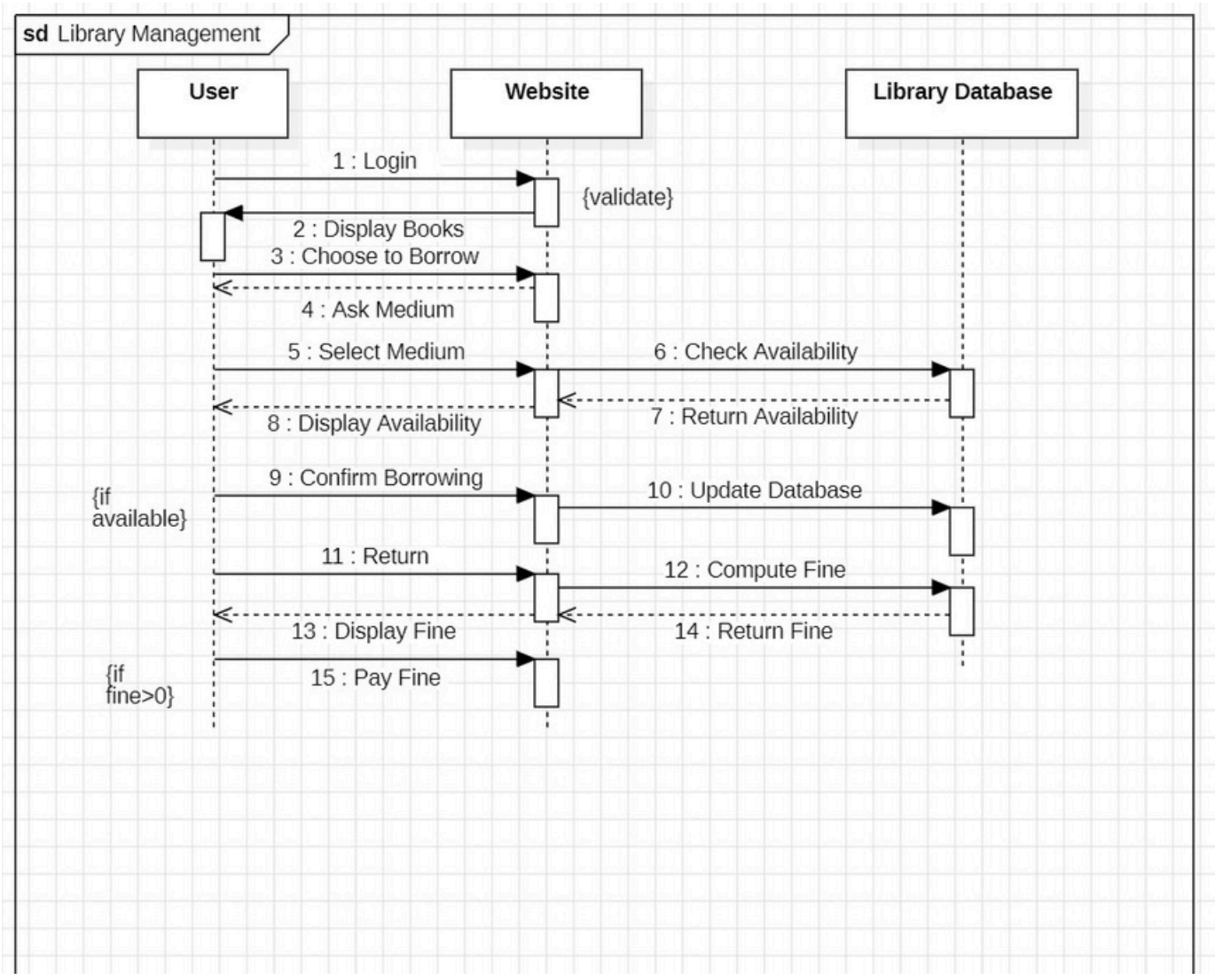
- Borrow:Userscanborrowbooks,whichtriggerstheIssueBookusecasemanagedbythe Librarian.
- Return:HandlestheprocessofreturningborrowedbooksandextendstoCalculateFine in case of overdue returns.

Librarian:

- IssueBook:Overseestheissuanceofbookstousers.
- ProcureBook:Handlestheprocurementofnewbooksforthelibrary.
- Print:Printsreportsordetailsasneededforadministrativepurposes.

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

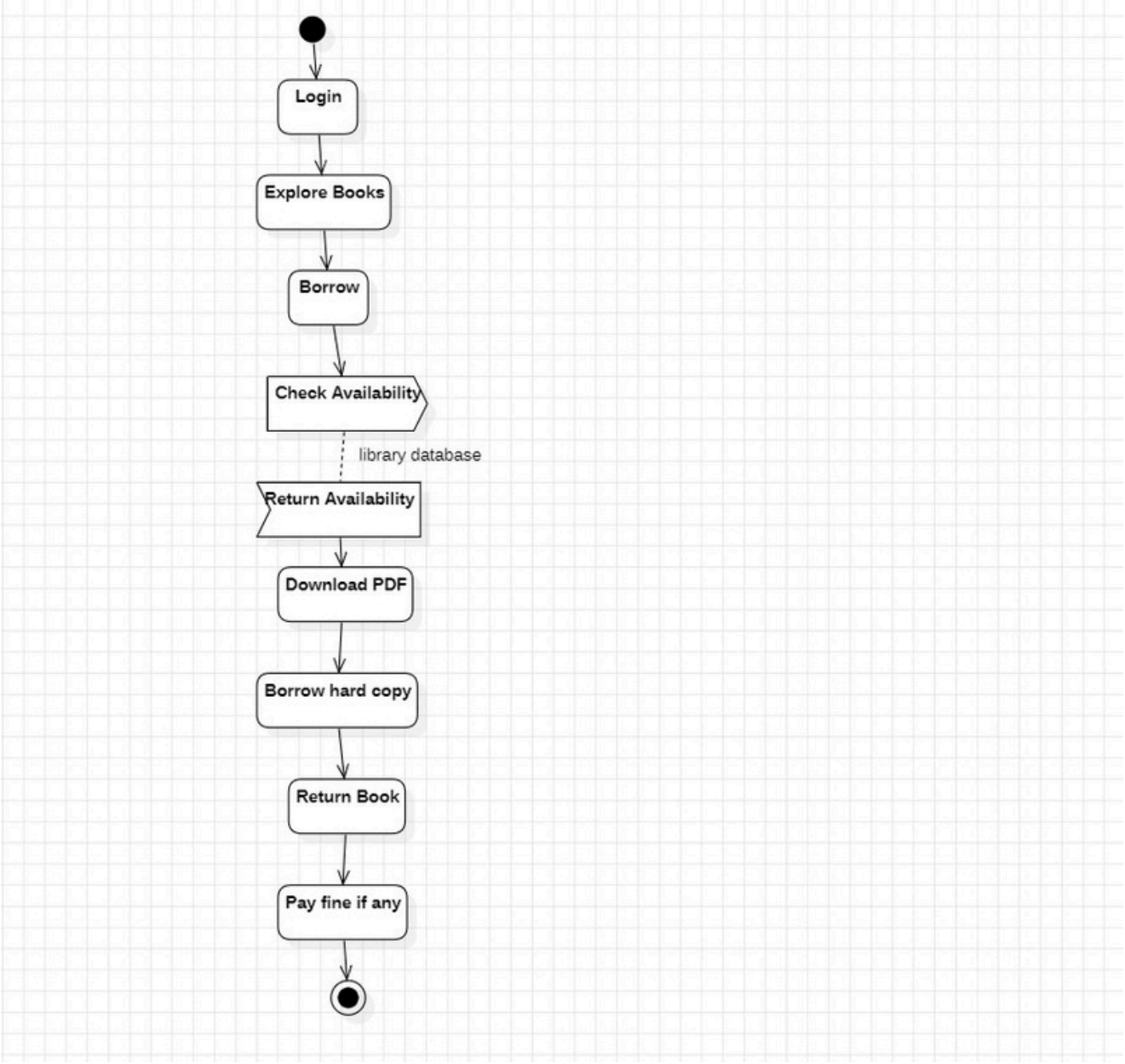
Sequence Diagram



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.

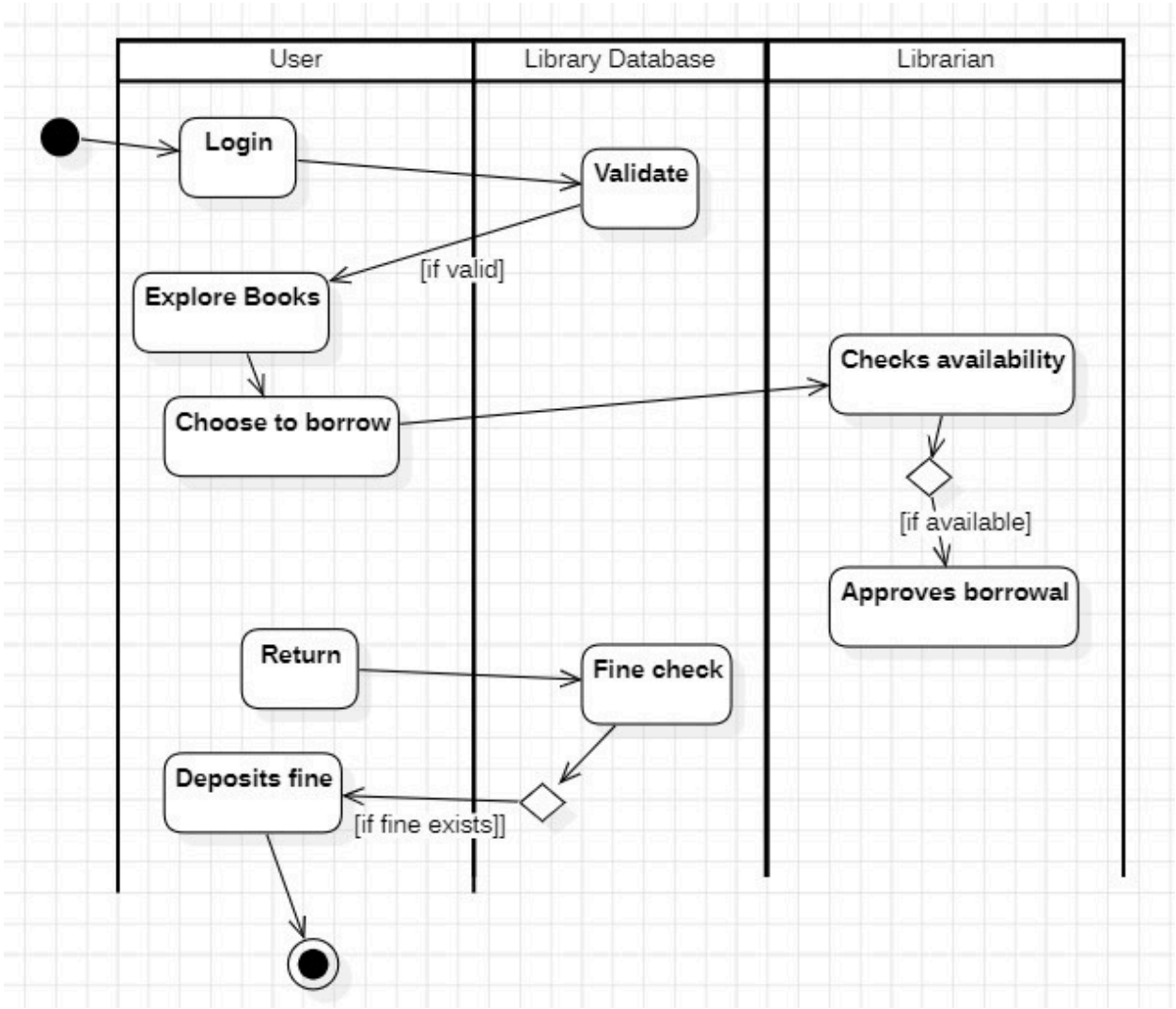
Activity Diagram



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.

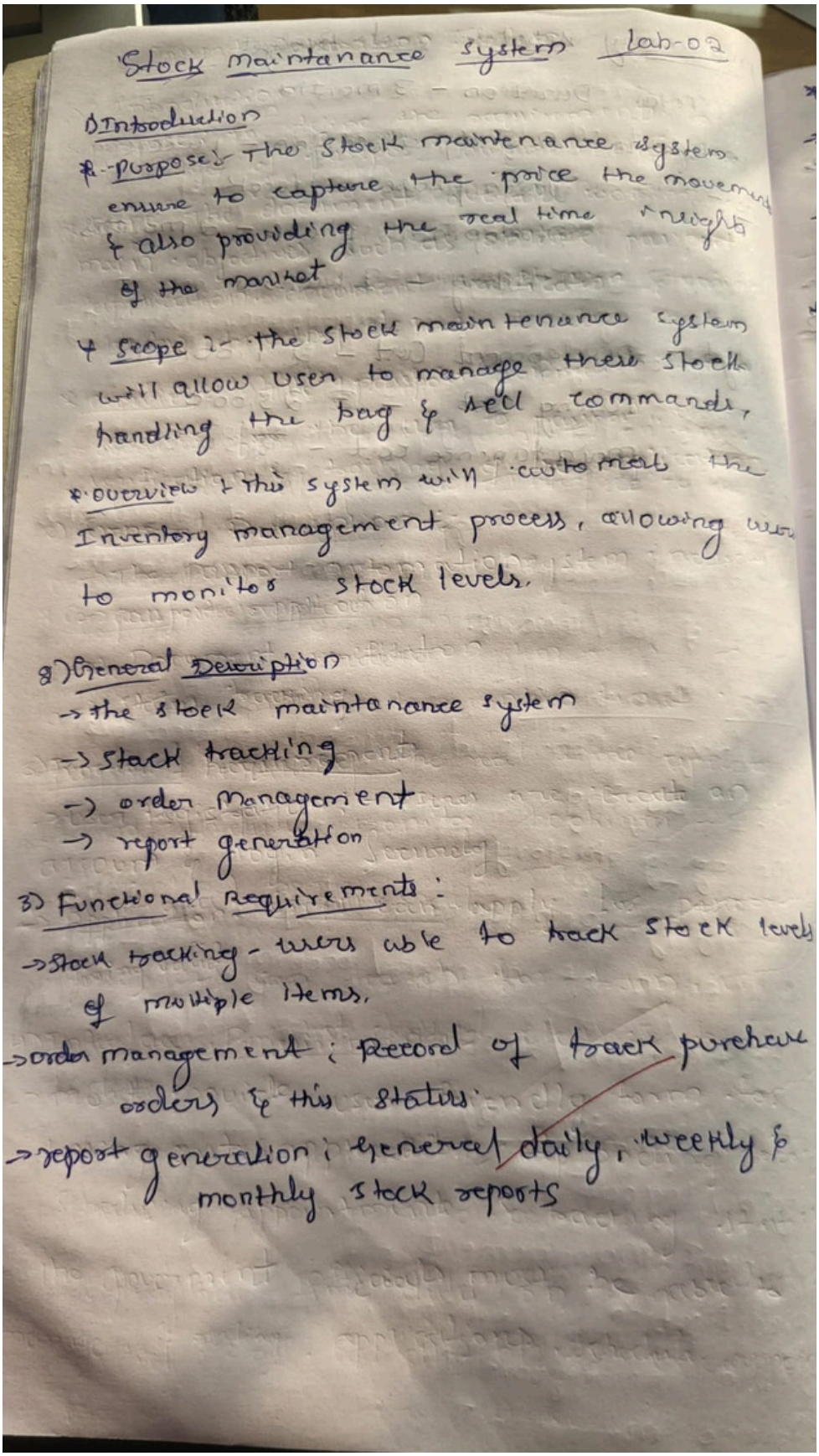
Activity Diagram using Swimlane:



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

4. Stock Maintenance System

SRS - Software Requirements Specification



Stock maintanance system lab-02

1) Introduction

* Purpose :- The stock maintenance system ensure to capture the price the movement & also providing the real time insights of the market

* Scope :- the stock maintenance system will allow user to manage their stock handling the buy & sell commands,

* Overview :- this system will automate the Inventory management process, allowing users to monitor stock levels.

2) General Description

-> the stock maintanance system
-> stock tracking
-> order management
-> report generation

3) Functional Requirements :

-> Stock tracking - users able to track stock levels of multiple items.

-> order management ; Record of track purchase orders & this status.

-> report generation ; General daily, weekly & monthly stock reports

* Interface Requirements.

→ UI : A GUI easy to navigate for users with minimal technical knowledge

→ Data Interface : the system will support imports expoit of excel & c & v files.

* performance Requirements:

→ Response time - system should respond to user Input within 2 seconds.

→ Data Accuracy : All inventory data must be updated in real-time with an accuracy

* Design constaints:

→ The system must be development using a web-build framework to ensure accessability.

→ system should run on standard servers without need for specialised hard.

* Non-Functional Requirements

→ security: system must impliment user authenticating encryption of sentive data

→ Scalability - to handle increased inventory.

→ Portability - system should be compatible to any platform.

* preliminary budget

→ Development time:- 3 months including testing

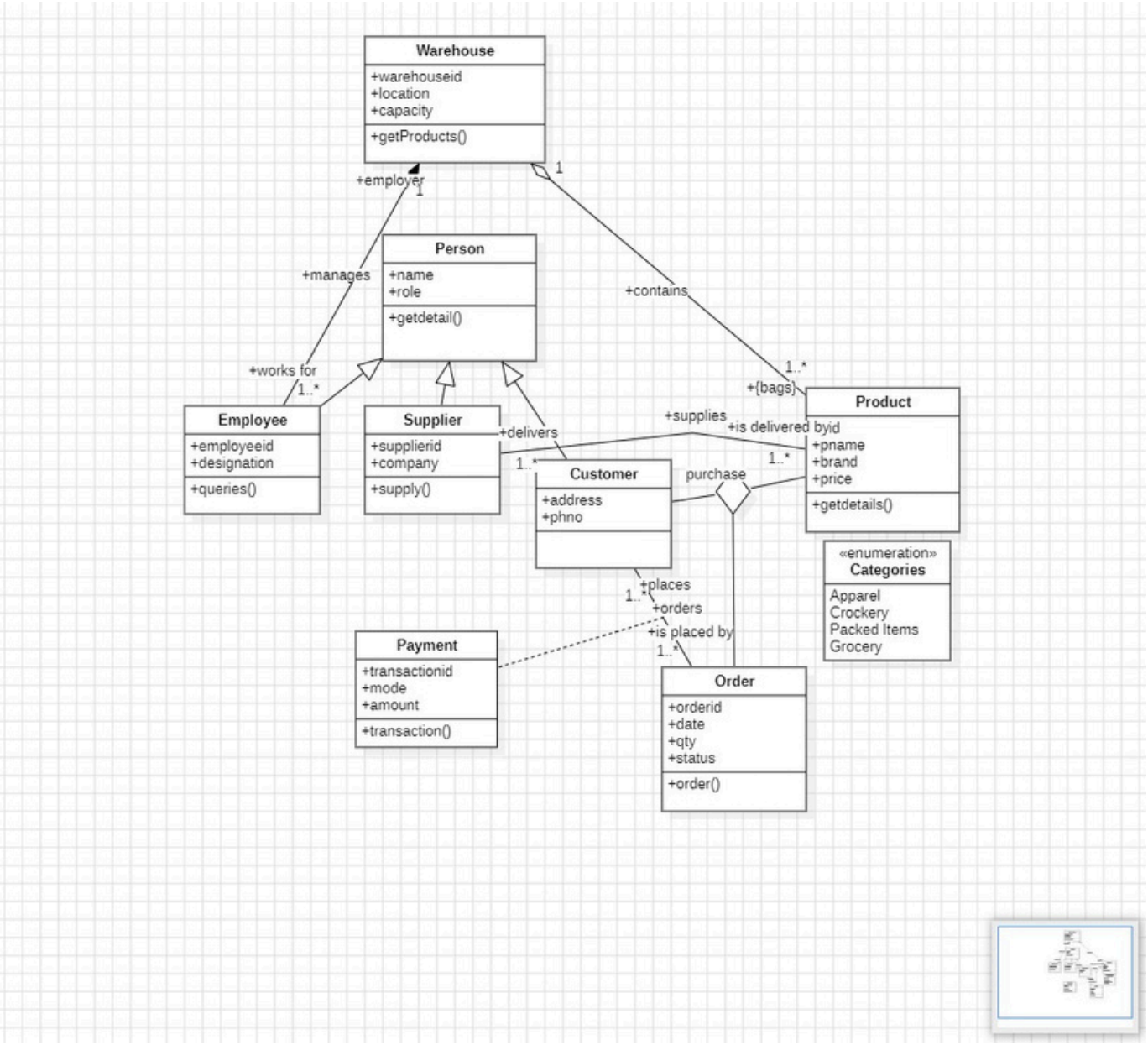
→ Budget: Development : ₹ 150000

Testing; ₹ 50000

maintanance; 50000/year

Total Cost ₹ 250000

Class Diagram

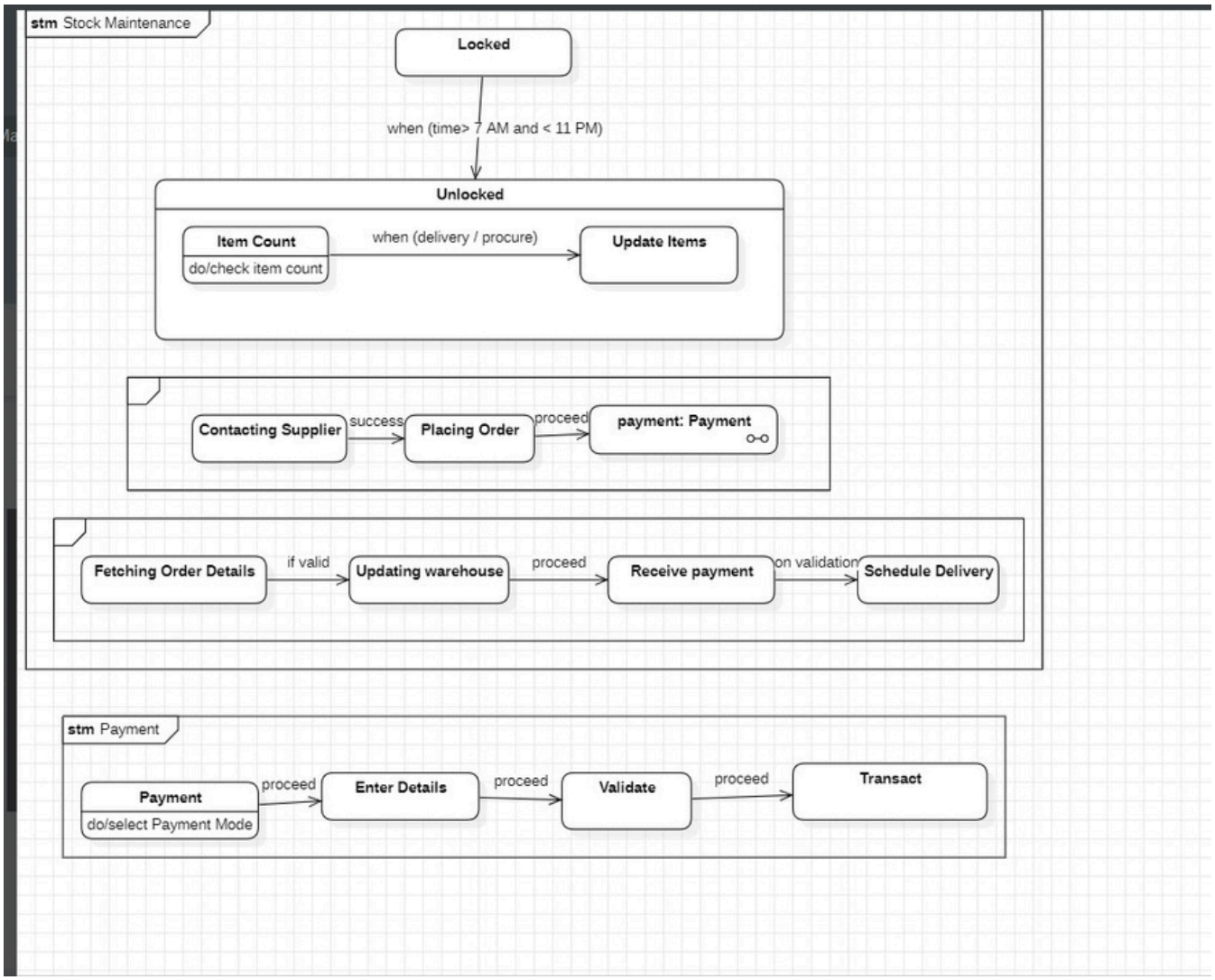


Description of Class Diagram:

- Warehouse:Representsstoragefacilitiescontainingproducts(*composition*withProduct).
- Person:Generalclass(*generalization*)inheritedbyEmployee,Supplier,andCustomer.
- Employee:Representsstaffworkingforthewarehouse(*aggregation*withWarehouse).
- Supplier:Representsentitiesdeliveringproductstothewarehouse(*association* with Product).
- Customer:Representsbuyersplacingordersforproducts(*association*withOrder).
- Product:Representsitemsstoredinthewarehouse,categorizedbytype(*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

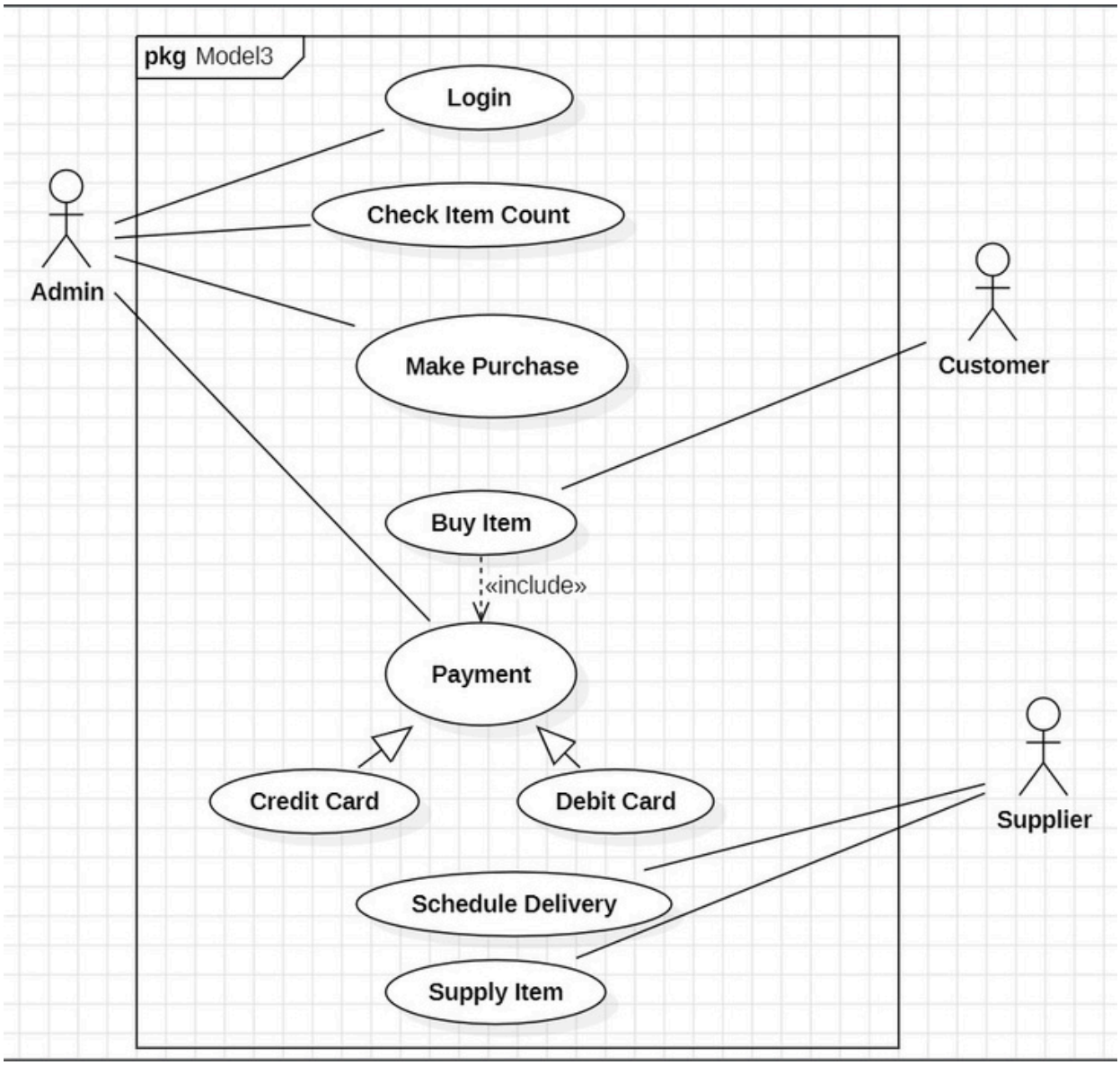


Description of State Diagram:

- ThestatediagramrepresentsaStockMaintenancesystemwithstatesformanaging inventory and deliveries.
- ThesystemstartsintheLockedstate,operationalonlybetween7AMand11PM, 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.

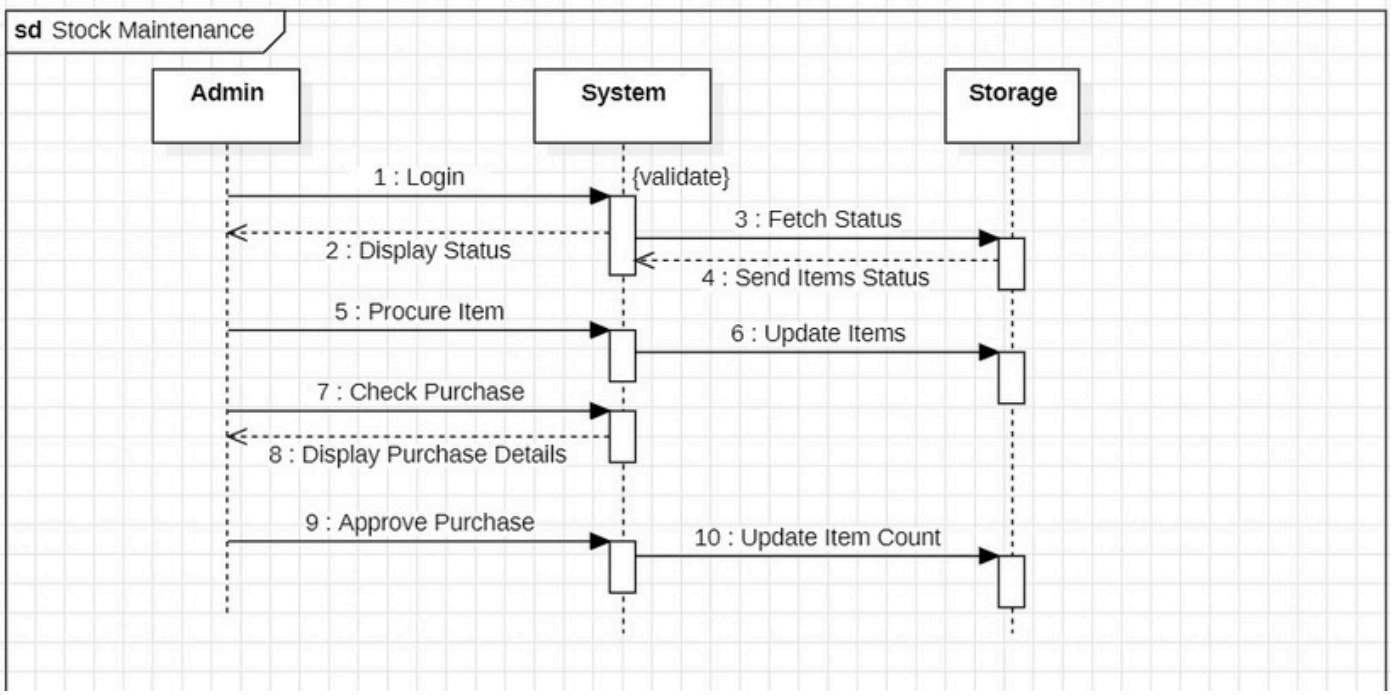
Use Case Diagram



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.

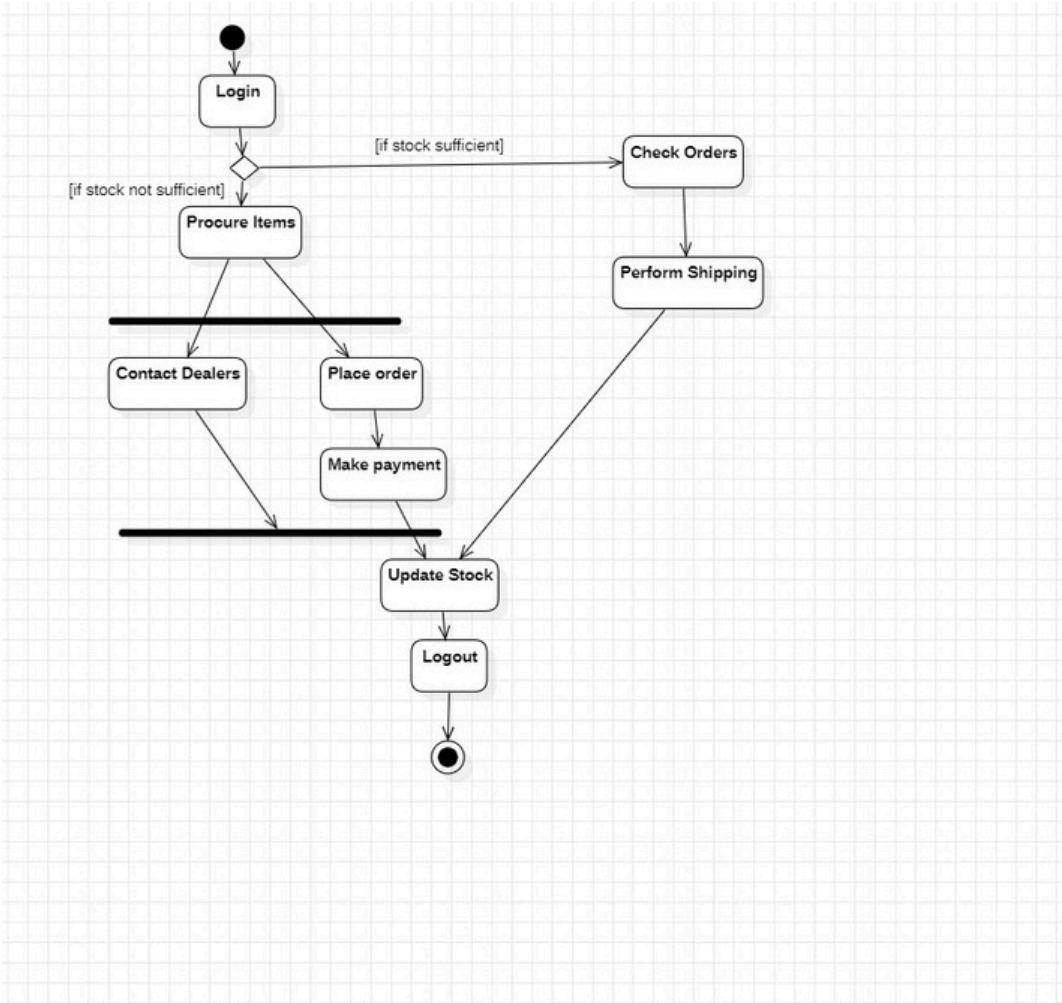
Check Purchase: The admin checks the purchase details to verify the procurement process.

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.

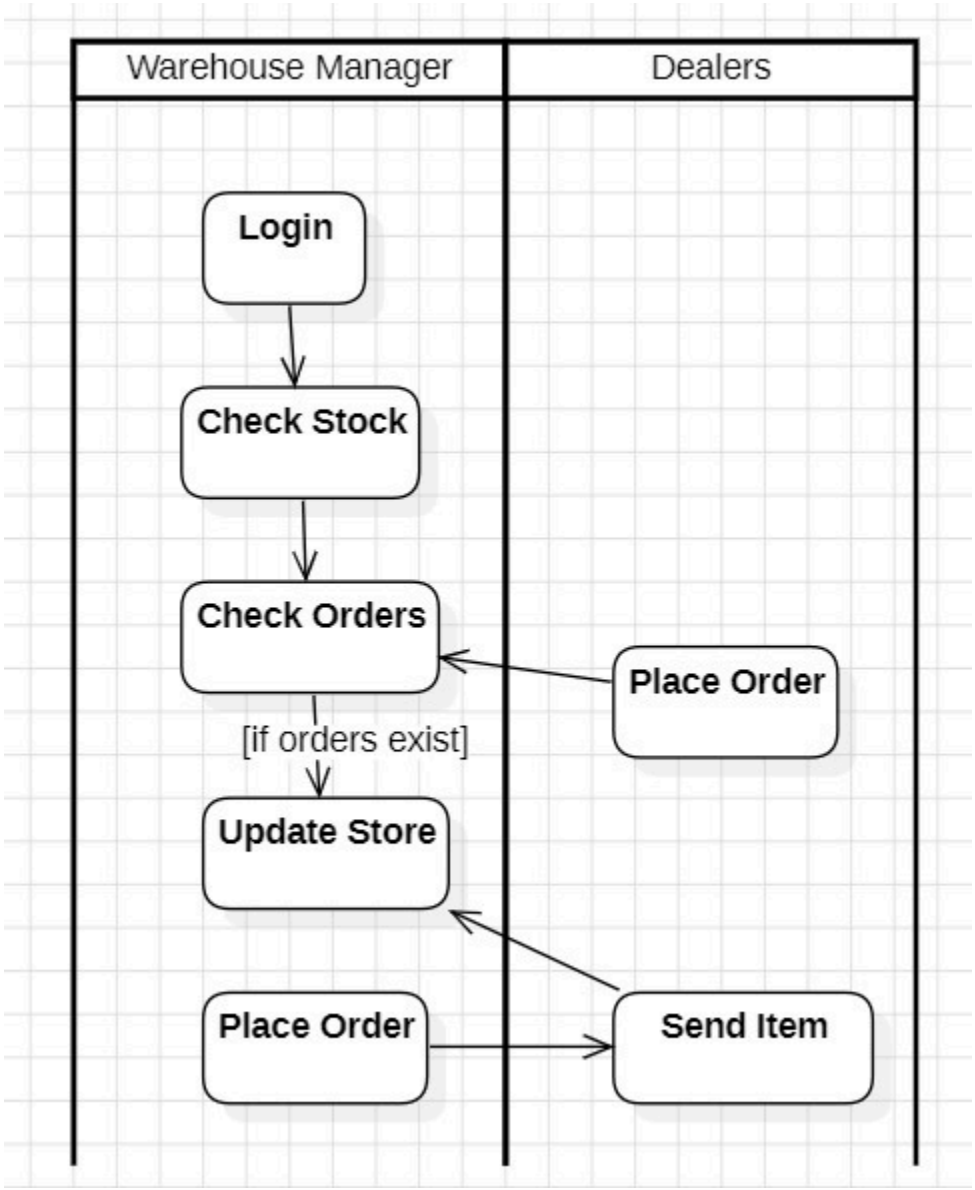
Activity Diagram



Description:

- Thewarehousemanagerlogsintothesystem.
- Theycheckifthereissufficientstock.Ifthereis,theyproceedtochecktheordersand deliver those items.
- Otherwisetheyplacetheorderbycontactingthedealersandprocuringtheitemsby making payments.

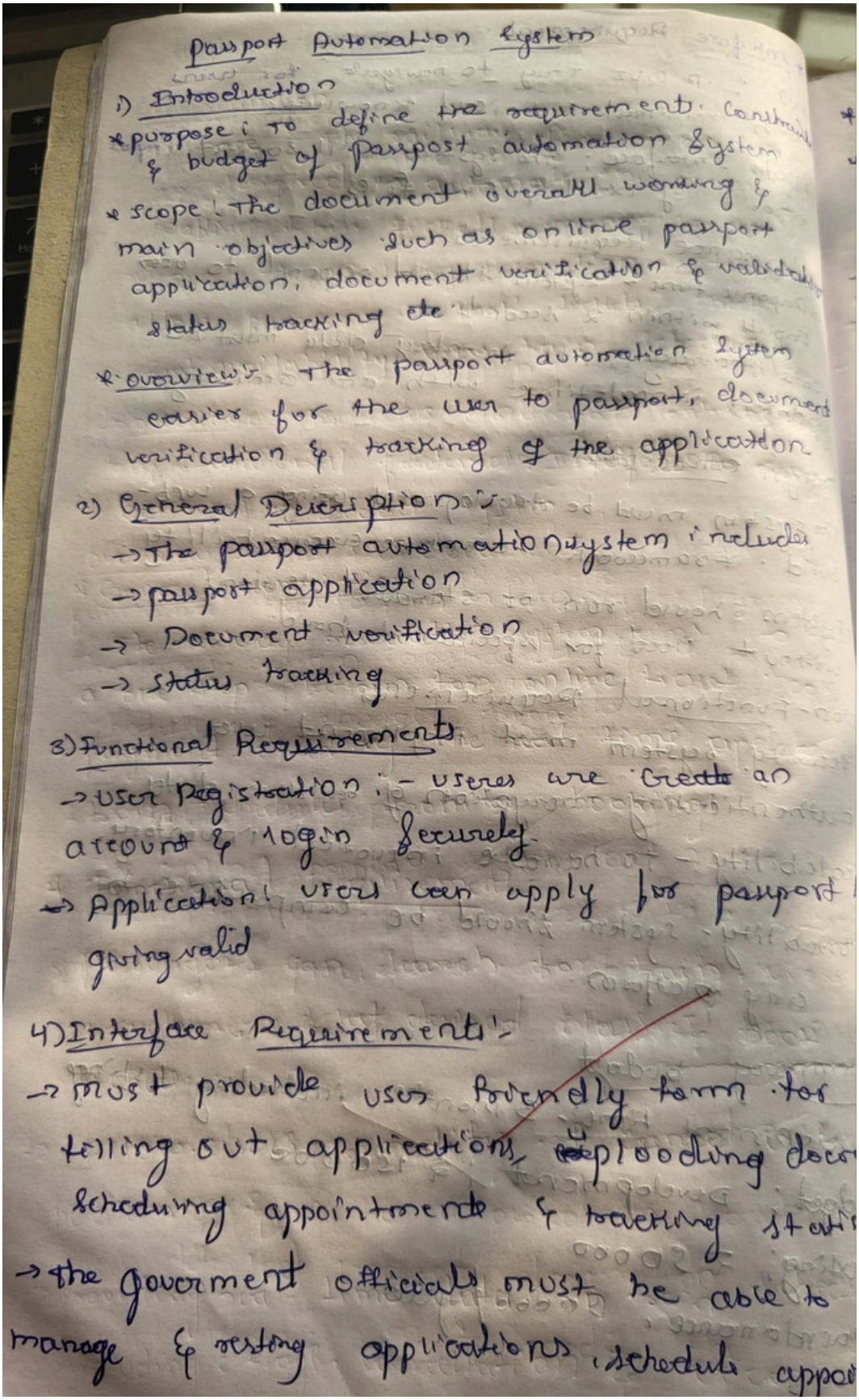
Activity Diagram using Swimlanes:



There are 2 swimlanes Warehouse Manager and Dealer.

5. Passport Automation System

SRS - Software Requirements Specification



Passport Automation System

1) Introduction

* purpose: To define the requirements, constraints & budget of passpost automation System
* scope: the document overall working & main objectives such as online passport application, document verification & validation status tracking etc
* overview: The passport automation system easier for the user to passport, document verification & tracking of the application

2) General Description:

→ The passport automation system include
→ passport application
→ Document verification
→ status tracking

3) Functional Requirements

→ User Registration: - users are Create an account & login securely.
→ Application: users can apply for passport giving valid

4) Interface Requirements:-

→ must provide user friendly form for filling out applications, uploading docu scheduling appointments & tracking status
→ the goverment officials must be able to manage & resting applications, schedule appoi

* Design constraints

* The system must obey the data protection must ensure secure storage & handling the personal data.

* the sensitive data like biometric information must encrypted & only the authorized person can be excessed

* performance Requirements

→ Scalability: It must be capable of handling large number of applications

→ must ensure high uptime for uninterrupted access to the system

* non-functional Requirements

→ Security - multifactor authentication for applicants & officials

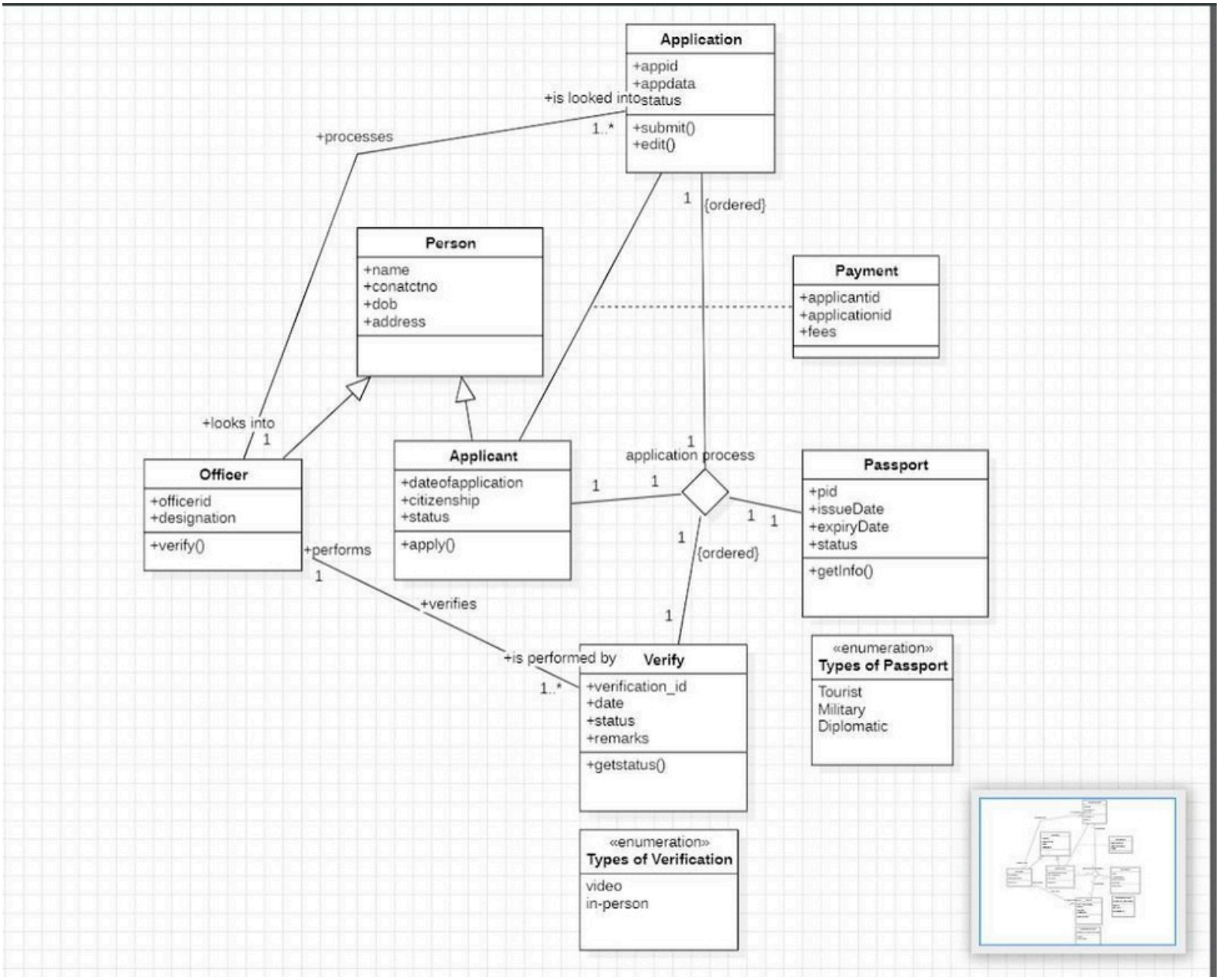
→ Usability - Easy to use interfaces for both users with minimal training req for new users

* preliminary ~~for~~ budget

→ Schedule:- Requirement Specification & analyse system Design: 6 weeks

→ Budget: 50 lakhs

Class Diagram

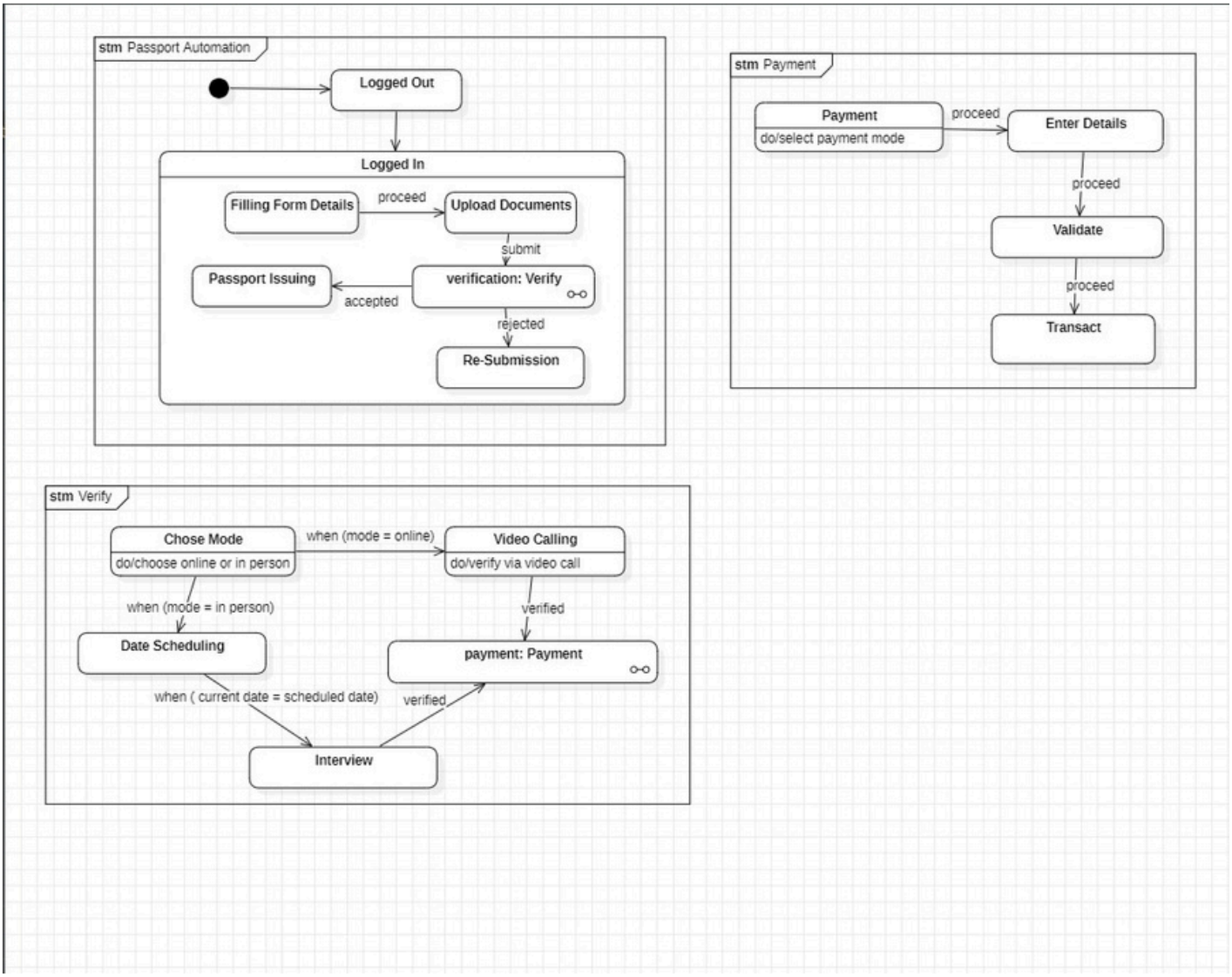


Description of Class Diagram

- Personisageneralclassassociatedwith ApplicantandOfficer,representingentities involved in the system.
- Applicantappliesforapassport,linkstoApplication,andhasattributeslikecitizenship and application status.
- Officerverifiesapplications,performingaverificationprocessrepresentedbythe 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

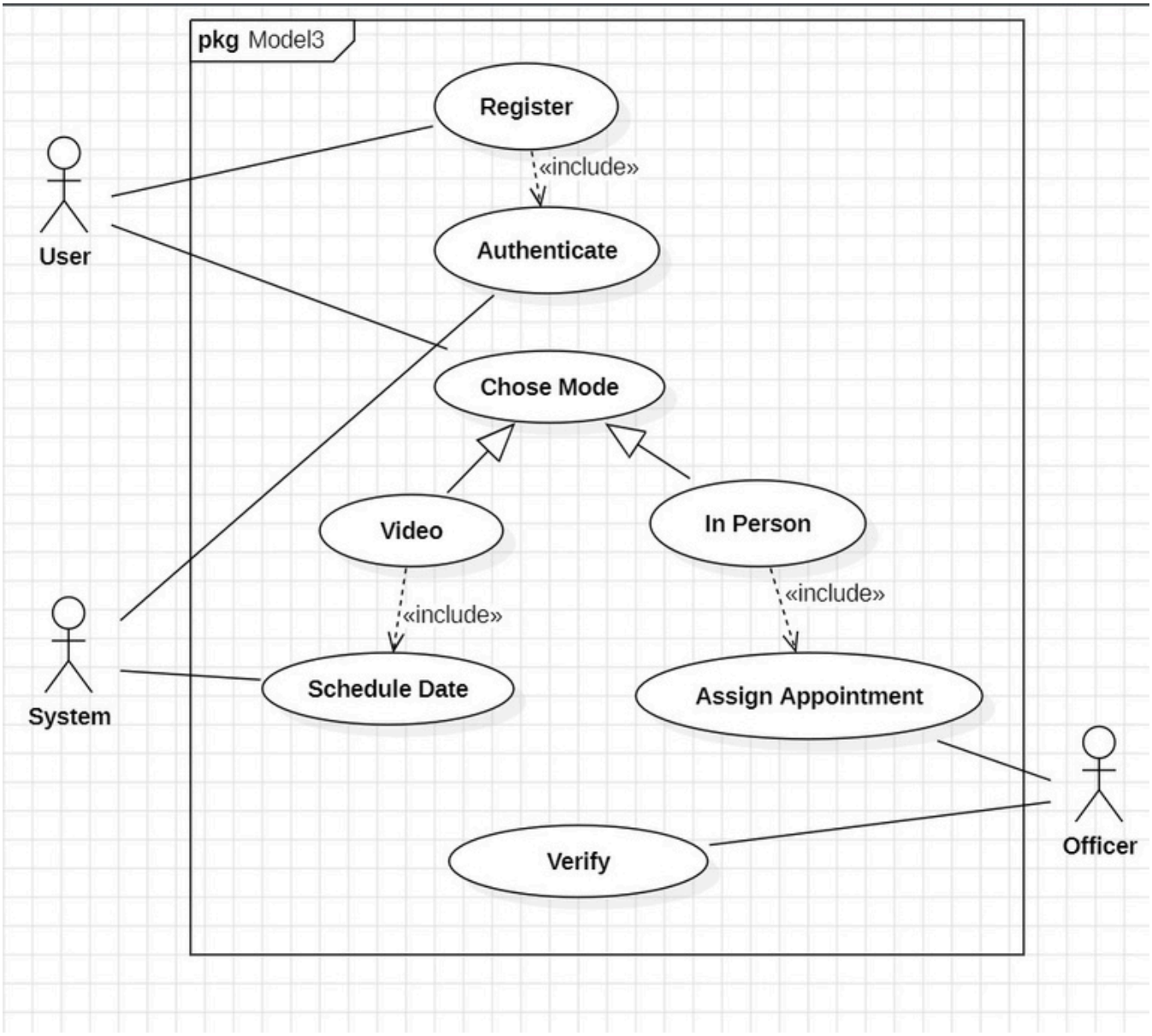


Description of State Diagram:

- ThisstatediagramrepresentsaPassportAutomationsystem.ItbeginsintheLoggedOut 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.

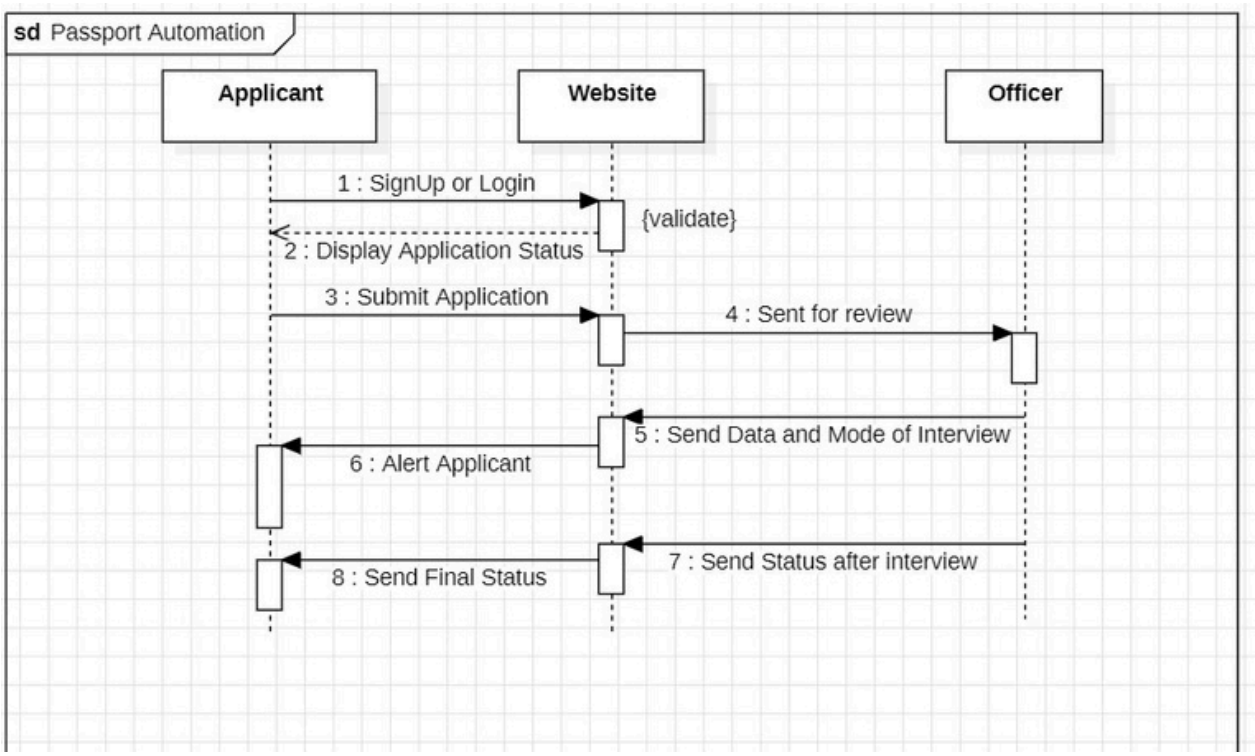
User: Registers and authenticates to access the system. They choose a mode—Video or In-Person.

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:Verificationextendstobothappointmentmodes,ensuringsuccessfulcompletionbythe 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.

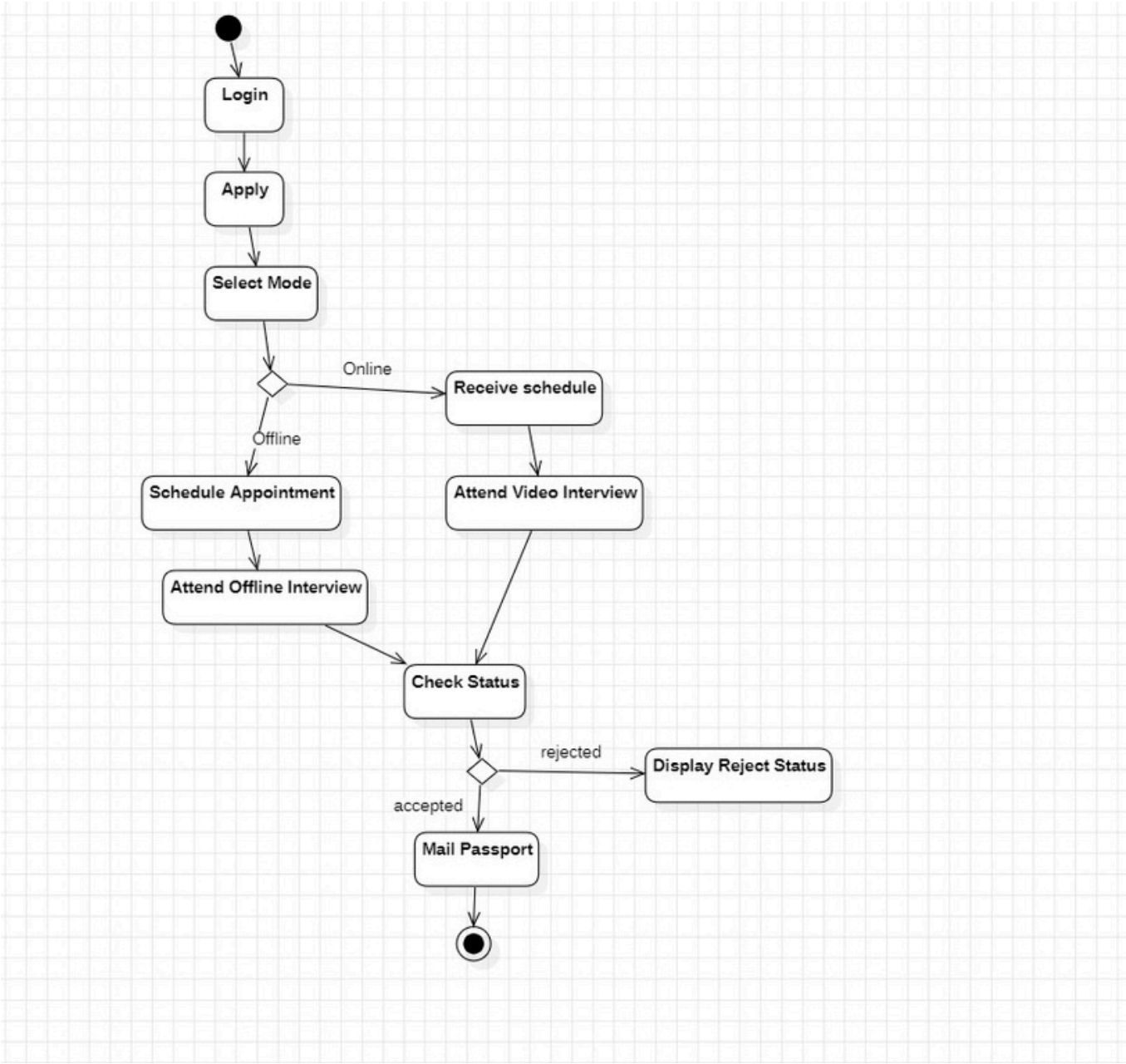
Send Data and Mode of Interview: The officer sends back the reviewed data along with the mode and schedule of the interview.

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.

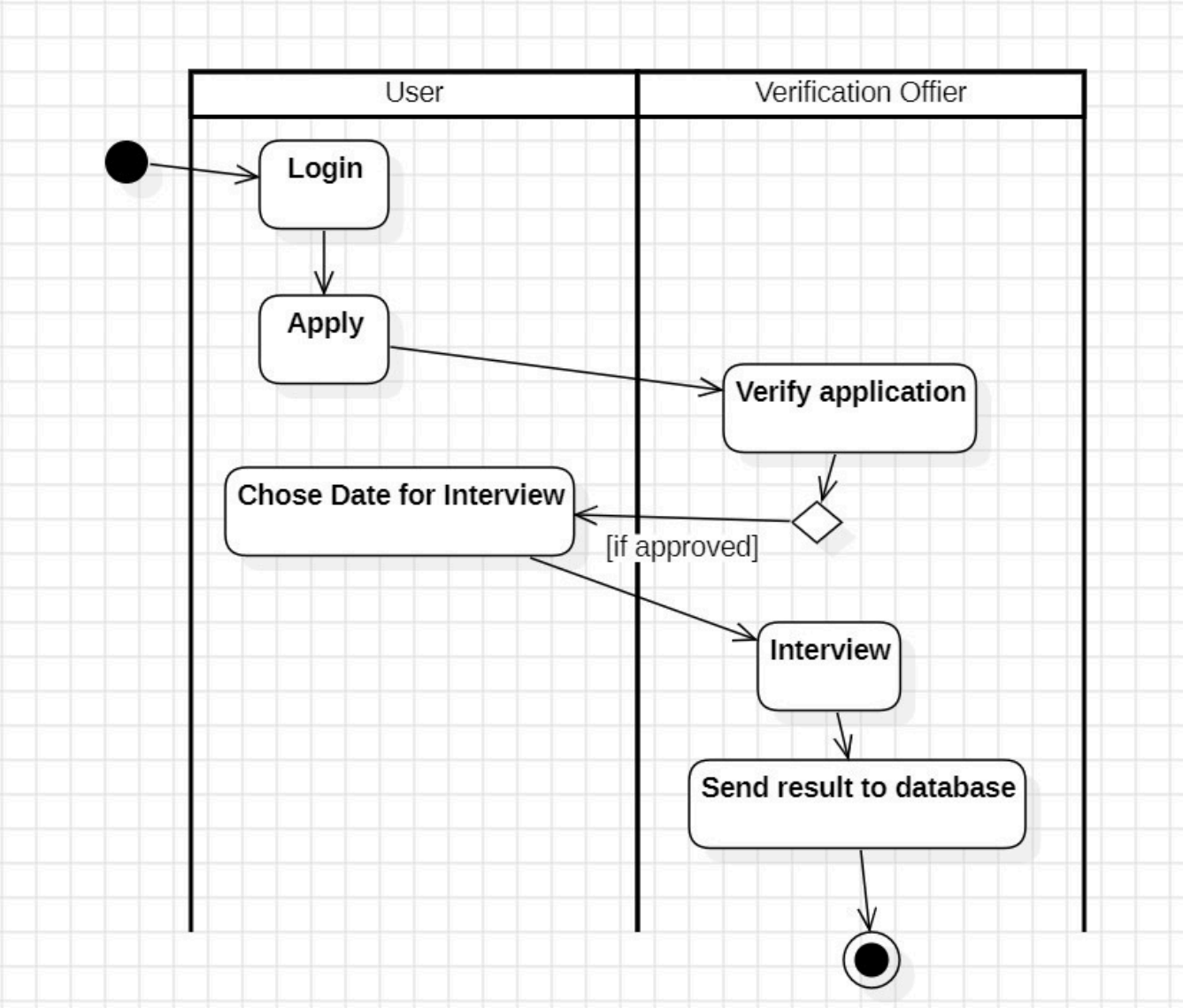
Activity Diagram



Description:

- Theuserlogsintotheapplication.Theyapplyforthepassport.
- Theyselectthemodei.eonlineoroffline.
- Ifonlinetheyreceiveascheduleandattendtheinterviewonline.
- Ifofflinetheyscheduleanappointmentandattendtheinterviewoffline.
- Postthistheycheckthestatusoftheirapplicationontheplatform.Ifaccepted the passport is mailed to them.

Activity Diagram with Swimlane



The swimlane has 2 lanes namely User and Verification Officer.