

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:

SHREE SANKET
1BM22CS261

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 Modelling (23CS5PCOOM) laboratory has been carried out by **SHREE SANKET(1BM22CS261)** during the 5th Semester Oct24-Jan2025.

Signature of the Faculty Incharge:

Sunayana S
Assistant Professor
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

Software Requirement Specification

Hotel Management System	
1)	<u>Introduction:</u>
1.1)	<u>Purpose:</u> To Get familiar with Software requirement specifications of Hotel Management System.
1.2)	<u>Scope:</u> document provide General description functional requirements, non functional requirements, Performance requirements of Hotel management system.
1.3)	<u>Overview:</u> The hotel management systems provides facilities of room, and other inventories to user for temporary usage.
2)	<u>General description:</u> The Product includes bunch of rooms for customers to stay, and use other inventories. The customers can stay in the hotel rooms for reserved time / days. Customer should have some identification proof and needs to pay in Room. During Traveling Hotels plays a major role for shelter for travellers. For those who don't have their own home in that area Hotels are important shelters for those customers.
3)	<u>Functional requirements:</u>
*	<u>Room management:</u> A clean and cheap Room Should be available for customer. The room should be maintained properly and a bunch of rooms should be

available for customers to choose. A proper toilet should be available.

- *.) Room Reservation System: customer should be able to book the rooms easily by any Govt. Proof of customers.
- *.) Staff management system: A proper staff management should be there in a hotel for the help & support of customers for their good experience.
- *.) Billing & payment: A billing or the payment should be given to user/customers. and many payment methods like UPI, cash, or card payment system should be provided to user.
- *.) Room service: A phone/telephone is given to customers for the room service facility to customer's room.
- *.) Feedback system: A feedback form is given to users for their feedback about the room experience.
- *) Interface Requirements: A proper software should be developed for the entry of customers data on counter and a website can be developed for online room booking and ease payment. methods should be provided to users. Interface should have facility to choose their room and reserve the room advance.
- *.) A room booking cancellation option should be given & refund options will be given to.

users.

5) Performance Requirements: A proper response given to customers for their requests for help & support or any essential inventories.

* The Online website for Room reservation should have a proper User interface & account management of customers.

* The staff should work properly in the hotel for the requests of customers.

6) Design constraints: The design team should be having limitations for website design / software design like the user interface limitations, Payment limitations, Reservation/cancellation, Authentication limitation

* The hotel should have a customer number limitations for the respective Rooms.

7) Non-functional Requirements:

* Security: The security for the customers room and the language in the room. The website should be provided with security and privacy with user authentication management.

* Account management: The website should have proper user account management system.

- * Cleanliness of rooms: The rooms should be maintained properly clean for better experience of users.
- * Responsive staffs: The staff members should be responsive & should behave nicely to users.

8) Preliminary schedule & Budget:

Hotel construction: 10 lac cr

website development: 1 cr

software development: 50 lac

Software testing: 25 lac

~~Software Validation: 25 lac~~

Class Diagram

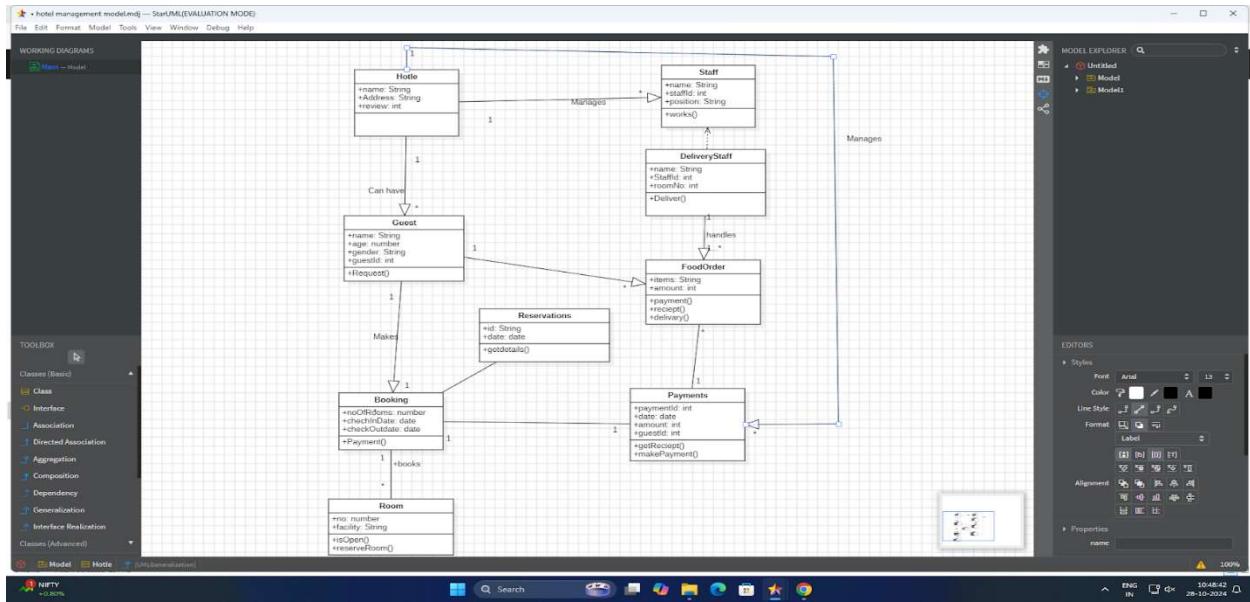


Fig1.1 Hotel Management System - Class Diagram

The diagram represents a hotel management system. It showcases the relationships between various entities such as Customer, Hotel, Room, Reservation, Payment, Service, and Staff. The diagram defines the attributes and operations associated with each entity, such as making a reservation, checking in/out guests, processing payments, etc. It also depicts the relationships between these entities, including one-to-one, one-to-many, and many-to-many relationships. For example, a customer can make multiple reservations, each reservation is associated with a specific room, and different types of staff members can be involved in various services. The diagram provides a comprehensive overview of the system's structure and interactions.

State Diagram

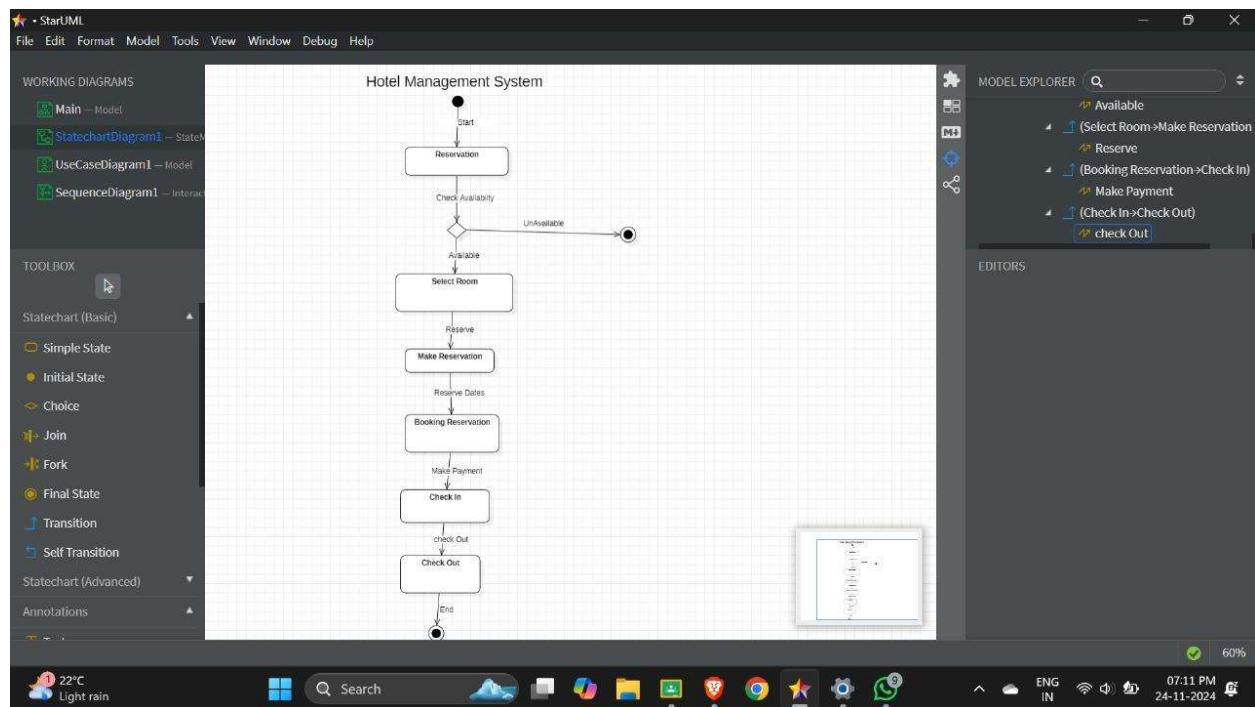


Fig1.2 Hotel Management System - State Diagram

The hotel management system state machine models the operational workflow of a hotel, transitioning through distinct states triggered by specific events. It begins in an Idle state, waiting for booking requests. Upon receiving a request, it transitions to Check_Availability to verify room availability. If rooms are available, it moves to Book_Room, confirming the booking; otherwise, it returns to Idle. Once booked, the customer proceeds to Checkin, after which they can avail services in Avail Room Services. The CheckOut state initiates upon the customer's departure, followed by Generate Bill to prepare their bill. In Payment Method, the customer selects how to pay, leading to Transact, where the payment is processed. Each state and transition ensures smooth and sequential operation of the system, ensuring efficiency and clarity in hotel management.

Use Case Diagram

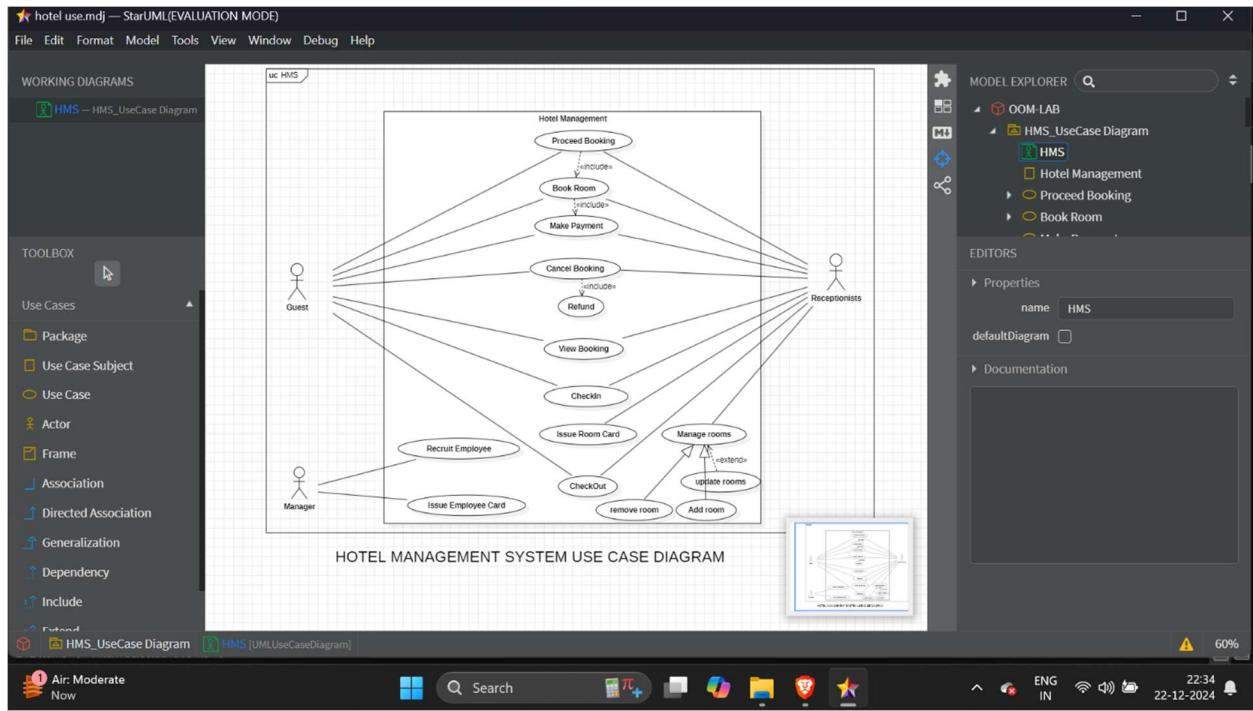


Fig1.3 Hotel Management System - Use Case Diagram

The diagram represents a Use Case Diagram for a Hotel Management System, showcasing various interactions between users (actors) and system functionalities. The primary actors include Customer, Manager, Receptionist, and Hotel Staff. Key use cases are grouped under the system, such as Authentication (which includes login and logout), Check Room Availability, Reserve Room (extended by Cancel Booking and including Make Payment), Check-In, and Room Services (further including cleaning, placing orders, and other hotel services). The diagram emphasizes the relationships and interactions among actors and system processes, demonstrating how each user contributes to the system's operations. For example, the Manager and Receptionist oversee updates and reservations, while Hotel Staff handle room services.

Sequence Diagram

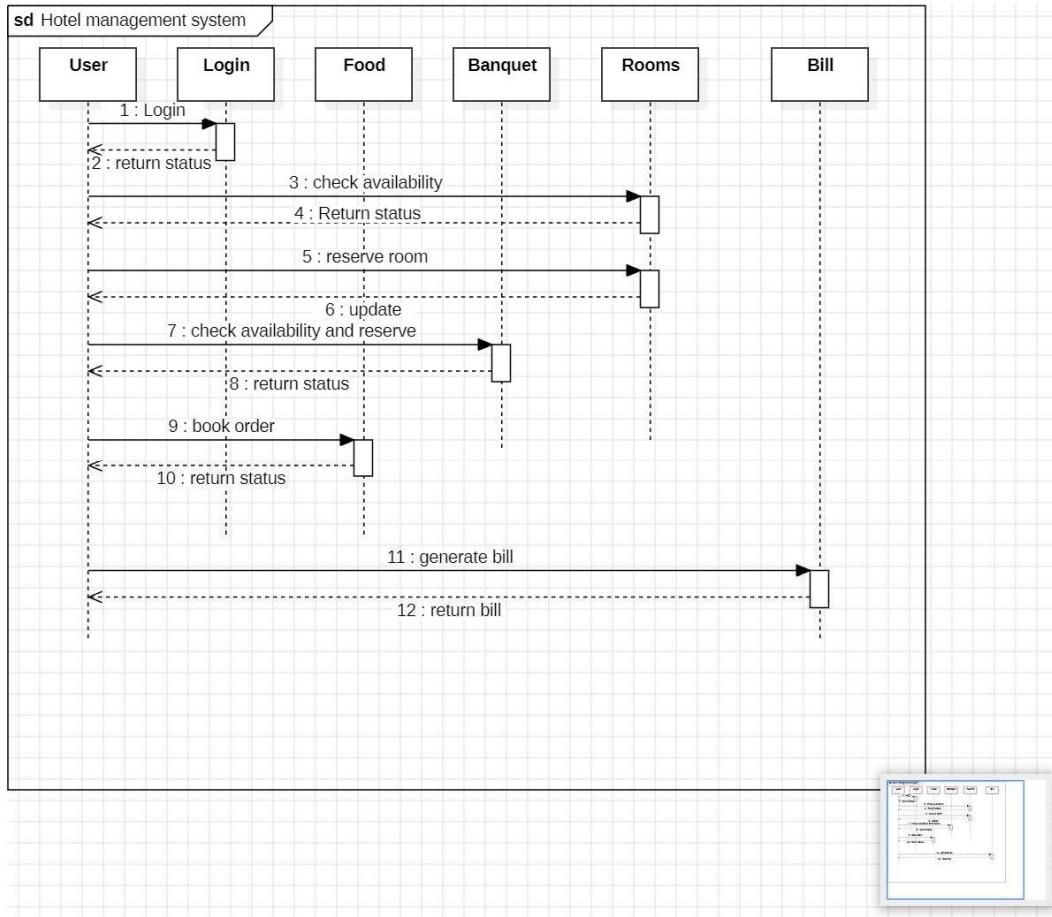


Fig1.4 Hotel Management System - Sequence Diagram

The sequence diagram illustrates the process of booking a room at a hotel. The customer initiates the process by requesting a room. The receptionist then checks the availability of the room in the database and returns the status to the customer. If the room is available, the customer can book the room. The receptionist sends the booking details to the database and confirms the booking status to the customer. The customer then makes the payment, and the transaction is processed by the database. Finally, the customer receives a receipt, and the database records the transaction details.

Activity Diagram

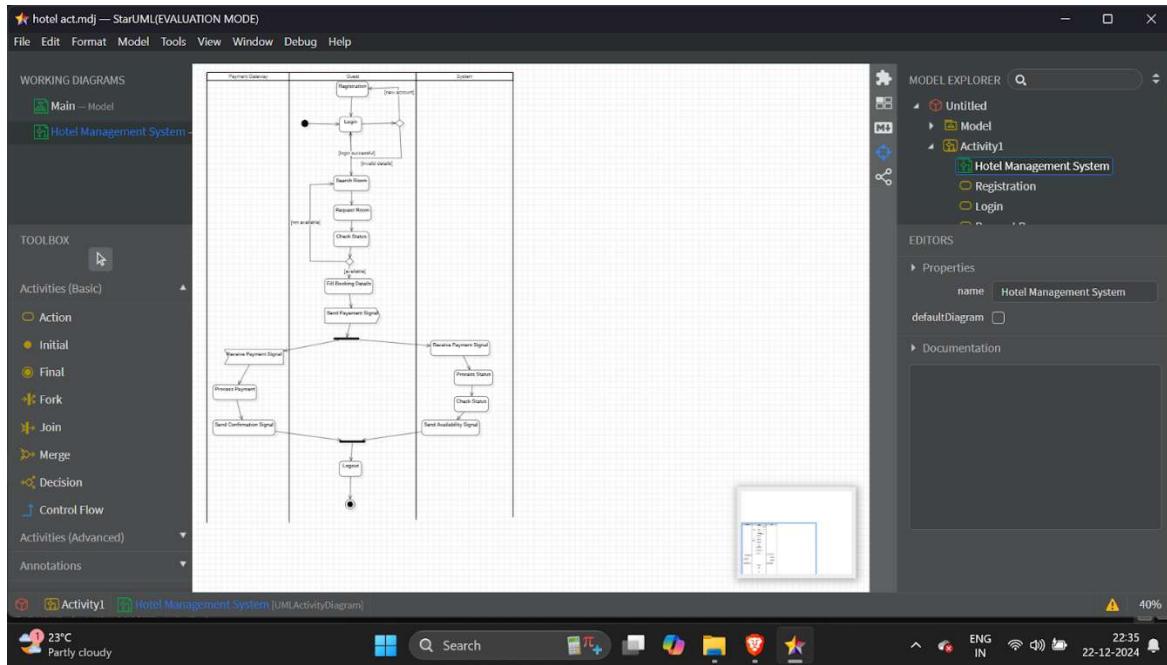


Fig1.5 Hotel Management System - Activity Diagram

The activity diagram illustrates the process of booking a room at a hotel. The customer starts by requesting to select rooms, and the system displays available rooms and their information. The customer then provides their details and selects a room. The admin approves the booking, and the system allocates the room and updates the payment. The customer pays the amount, and the system updates the room availability and balance. Finally, the customer receives a receipt, and the system updates the balance.

2.Credit Card Processing System

Software Requirement Specification

SRS Document

1) Introduction:

- 1.1. Purpose: To get familiar with software requirement specifications of credit card system.
- 1.2. Scope: Document provide a detailed description of software requirement of credit card system. It will provide transaction making service and Non-functional requirements to users.
- 1.3. Overview: Credit card system is used for making transactions and other services will be provided to users. Non functional requirements like security, privacy etc... will be provided to users.

2) General description: User's objective is to make secure transactions from one account to another account. For which user should have an account in respective bank, to use credit card services.

Features of credit card system is making secure transactions, and ~~get~~ store their money in account and use when ever they want.

Benefits: Users can save their money securely and gets many offers and rewards from the bank.

6) Design constraints:

- * Designers should have constraints about card size width when designing the card
- * They should design an algorithm for card chip handling
- * Designers should be knowing about constraints of login / authentication algorithm.
- * Should implement efficient no charges & balance handling algorithms.

7) Non-functional Requirements:

- * Security: An efficient security system should be required with secure authentication services.
- * Efficiency: An efficient and complete transaction is required.
- * Privacy: The user will be provided with privacy about their account. The data will be shared between the user and bank only.
- * Backup and recovery: The user will be provided with a service where he can enable or disable his card at any time.
- * Data management and integrity: The users data will be maintained efficiently.

3) Functional requirements:

- 1) Efficient Transaction
- 2) Pin validation / Authentication service
- 3) Notification/Alert services
- 4) Card Issuance
- 5) customer support
- 6) Billing Service
- 7) Offers and rewards.

4) Interface requirements:

- * Apps/website applications: Applications where users can enter their credit card details and make payments.
- * Payment Gateway using credit card & communication with other systems like banks, to process payments.
API's like
- * Payment API
- * Balance management API
- * Authentication API. are required.

5) Performance requirements:

- * Firstly users needs to make authentication using pincode Then the user will be able to make transactions efficiently.
- * The user will Get offers and rewards.
- * The user will Get notifications to his registered mobile number about the transactions he makes.
- * User will also receive notifications to his mobile number about the transactions he makes.

Preliminary Schedule and Budget:

Schedule: Requirement analysis - 2 weeks

Software development - 4 weeks

Validation - 1 week

Testing - 1 week

Budget - 50 lakhs. →

Requirement analysis - 10 L.

development: 25 L

Validation: 5 lakhs

Testing : 10 L

Functional Requirements:

1) Transaction: user will be allowed to make Transaction to any Other account.

2) Card Issuance: The user will be issued with a credit card and allowed to set it's pincode.

3) Billing Services: The user will Get Billing services to his mobile Through notification services

4) Realtime Account management: The user Transactions, and balance will be immediately updated in his account after Every Transaction.

5) Customer support: The user will be provided with 24x7 customer support services and help for efficient usage.

30/9/2024

Class Diagram

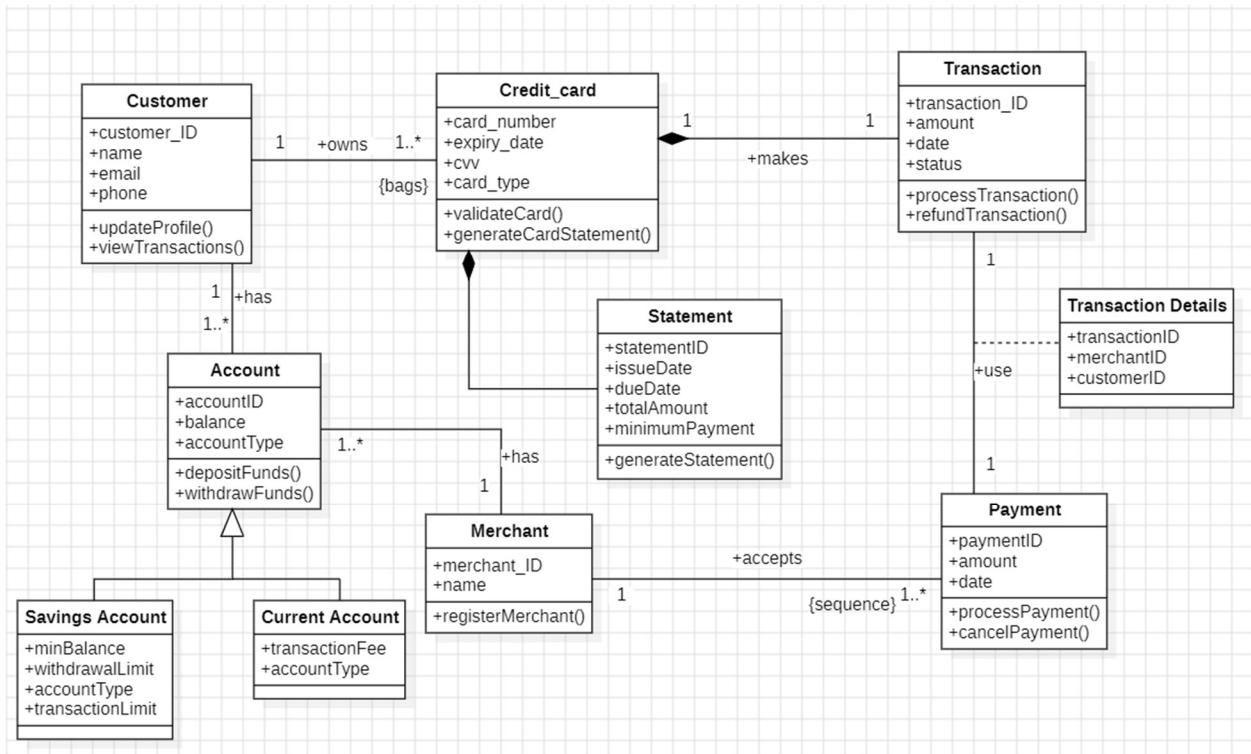


Fig 2.1 Credit Card Processing System - Class Diagram

The class diagram represents a credit card processing system. Customers own one or more Credit Cards, which are used to perform Transactions. Each credit card is validated and associated with a Statement that includes payment details like total amount and due date. Accounts (Savings or Current) store the customer's funds and enable deposits and withdrawals. Merchants register to accept payments, and payments are linked to Transaction Details, specifying the customer and merchant involved. Key functionalities include processing and refunding transactions, validating credit cards, generating statements, and updating customer profiles.

State Diagram

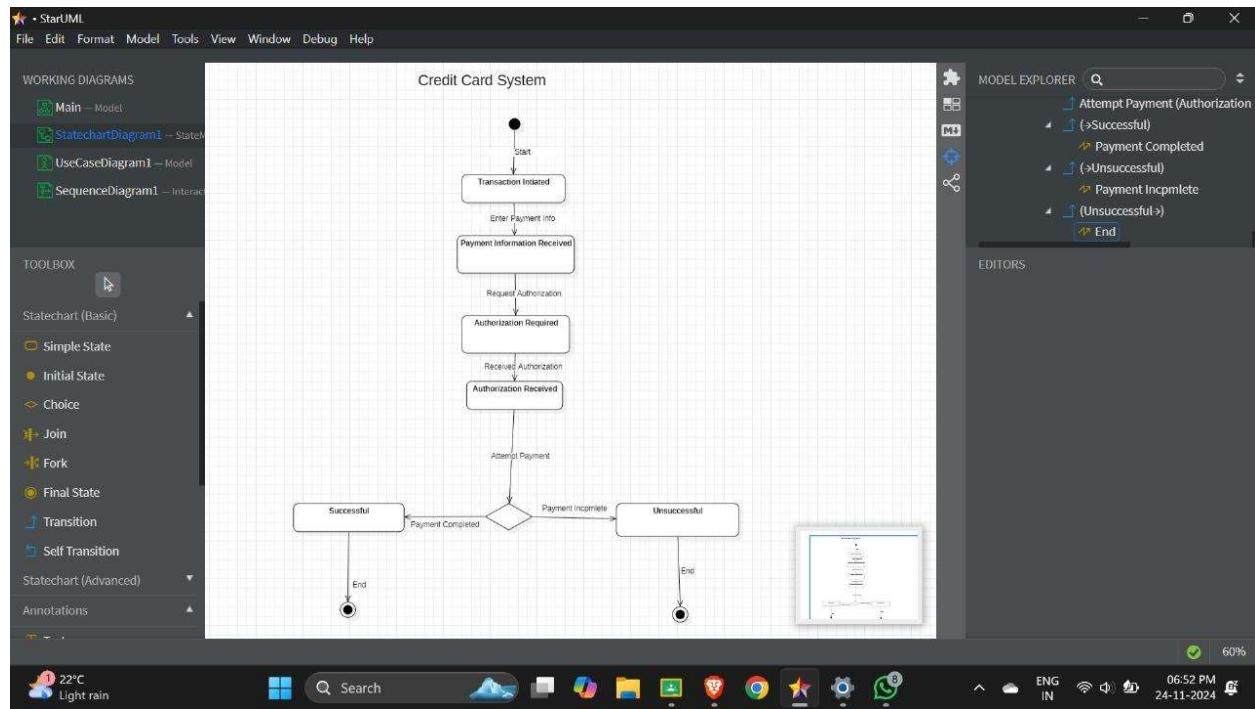


Fig 2.1 Credit Card Processing System - State Diagram

The state diagram illustrates the process of a credit card transaction. The system starts in an idle state and transitions to the "Read Card" state when a transaction is initiated. The card is then validated, and if it is invalid, the transaction is rejected. If the card is valid, the system moves to the "Authorization" state and verifies the card with the bank. If the card is verified and the funds are available, the system proceeds to the "Process Transaction" state and completes the transaction. If the card is not verified or there are insufficient funds, the transaction fails.

Use Case Diagram

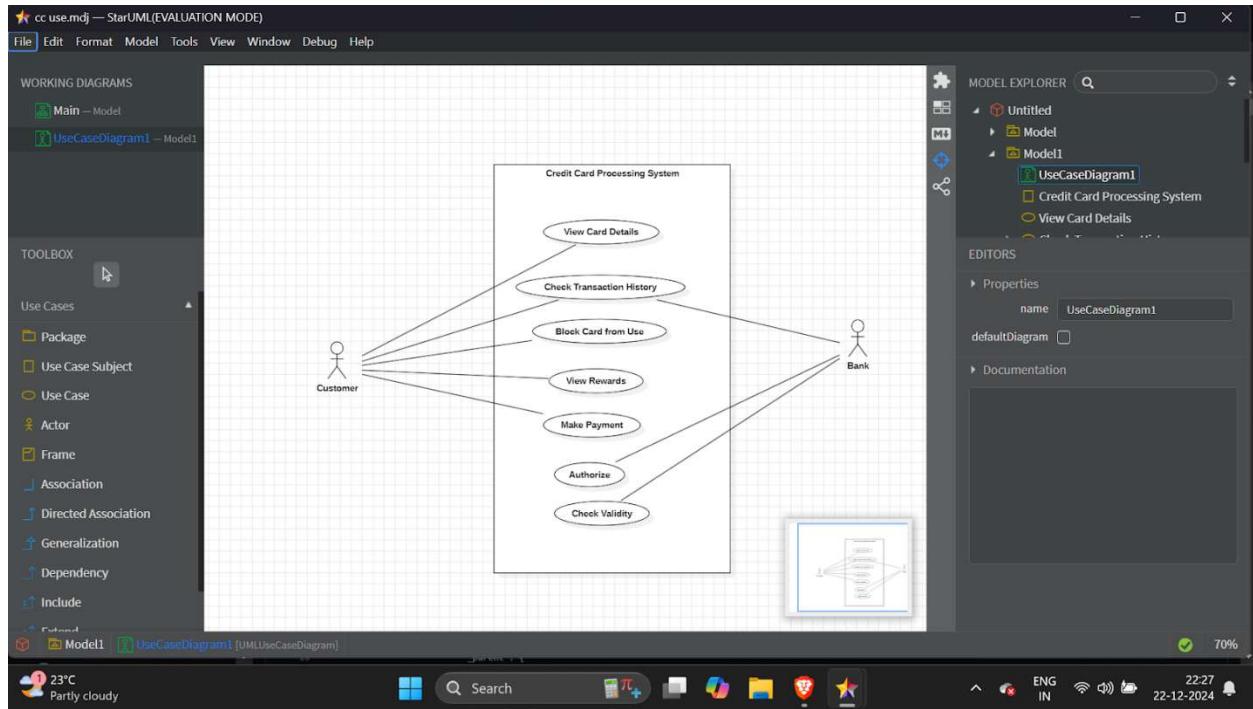


Fig 2.1 Credit Card Processing System - Use Case Diagram

The diagram depicts a Use Case Diagram for a Credit Card Processing System, highlighting the roles of the primary actors: Customer, Bank, and Merchant. The Customer interacts with the system to make payments (which includes verifying card details and approving payments), track transactions, generate statements, block or unblock cards, and pay bills. The Bank is responsible for approving payments, monitoring transactions, updating customer information, adjusting credit limits, and managing users. Additionally, merchants can request refunds, which involve approval and subsequent processing by the system. This diagram effectively illustrates the interactions and responsibilities within the credit card processing workflow.

Sequence Diagram

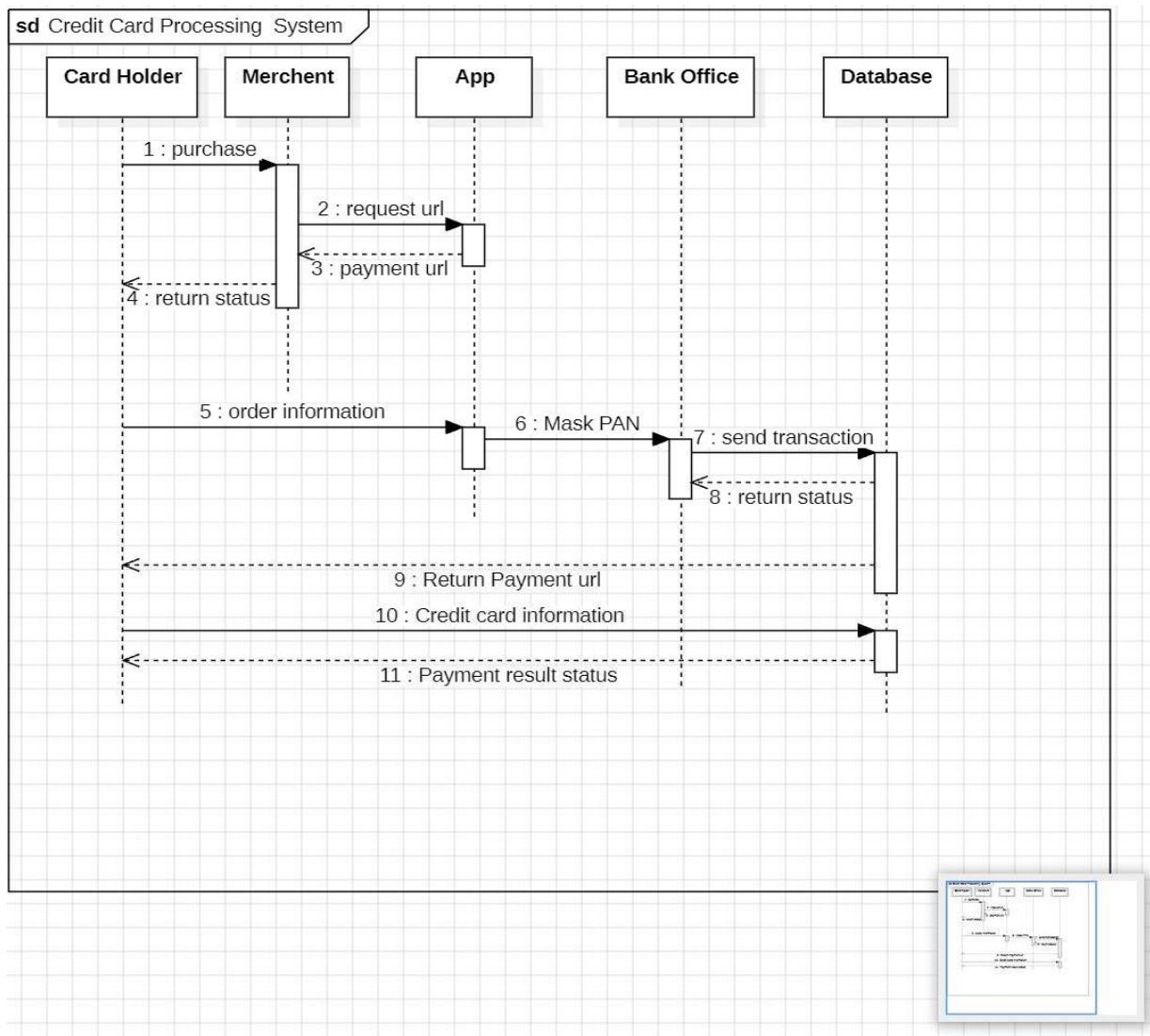


Fig 2.1 Credit Card Processing System - Sequence Diagram

The sequence diagram illustrates the process of a credit card transaction. The customer initiates the payment, and the merchant sends the payment details to the payment gateway. The payment gateway requests the customer to enter their PIN for authentication. Once the PIN is entered, the gateway authenticates the card with the bank. If the card is authenticated, the payment gateway processes the payment and confirms it to the merchant. Finally, the customer receives a notification of successful payment. If the card authentication fails, the payment is not approved.

Activity Diagram

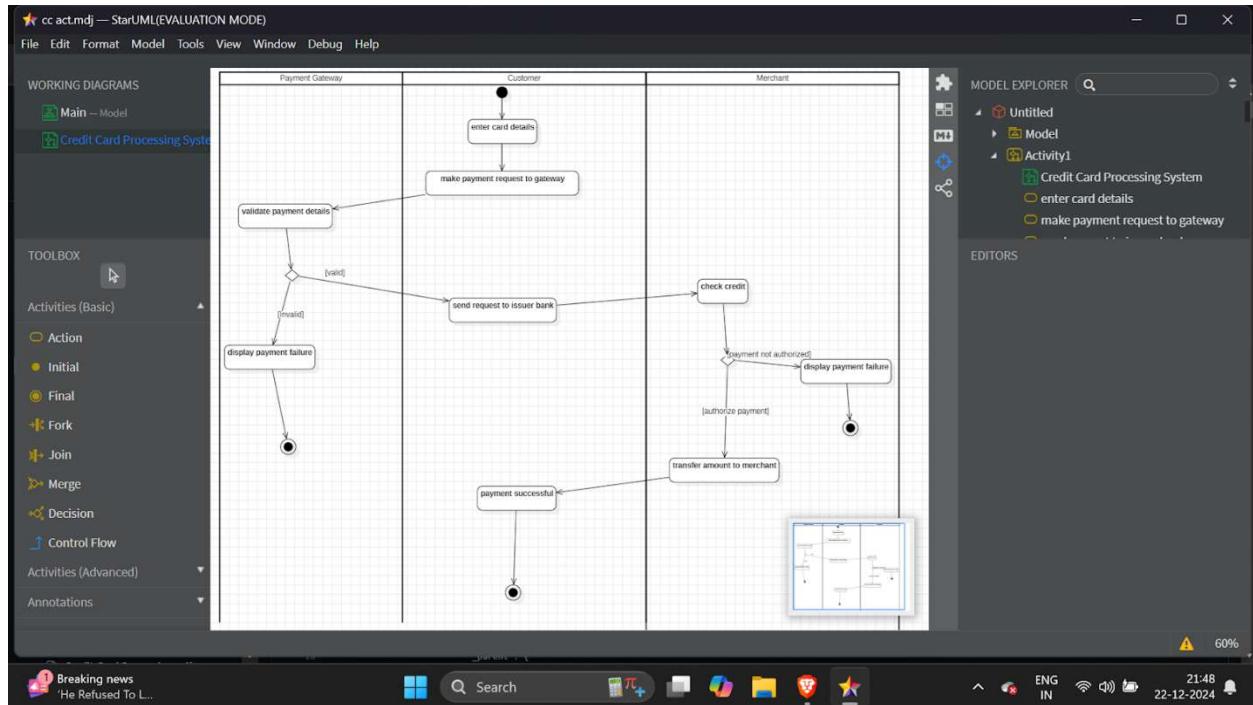


Fig 2.5 Credit Card Processing System - Activity Diagram

The activity diagram illustrates the process of a credit card transaction. The customer starts by entering their card details and making a payment request to the gateway. The gateway validates the payment details. If the details are valid, the gateway sends a request to the issuer's bank to check the credit. If the credit check is successful, the bank authorizes the payment and the gateway transfers the amount. The customer then receives a notification of successful payment. If the payment details are invalid or the credit check fails, the transaction is rejected, and the customer receives a notification of payment failure.

3. Library Management System

Software Requirement Specification

- 3) Library Management system.
- ① 1) Introduction:
- 1.1) Purpose: To Get familiar with the SRS of library management system.
 - 1.2) Scope: The objective of document is to provide a description of details of Requirements.
 - 1.3) Overview: The document gives information about book management requirements, Staff management system requirements, catalogue system requirement, searching of book req system. etc.
- 2) General description: The library management will be having ease of book search system, ease borrow & return management. The customers will be having restriction on the no. of books he borrows. The us.
- * The user will be Given a Identity library card for the reference of librarians.
 - * A catalog / registration of borrowing & return of the Books. will be there in library.
- 3) Functional Requirements:
- * Authentication: user will be Given a Library card which he should bring whenever he needs to borrow or return a book in library.
 - * Books management: The books in the library should be maintained in order of category for the eas.

- of access.
- * Library staff management help/support:
The library staff should help customer to find required book.
 - * Software for data management: The Librarian should be provided with a perfect software for the entry of customers, borrowed or returned books data for tracking.
- 4) Interface Requirements: The software for book management should be proper for the Librarian to enter the user's data.
- * The software should have a good interface and database management to store data.
 - * Software should have options to borrow, store & return data and the books name of books.
- 5) Performance Requirements:
- * The library should contain all types of books for users and should have proper management of books.
 - * The staff should be very supportive for the borrowers for their better experience.
 - * The Data Entry website should perform perfectly for long time and large data entry should be possible.

6) Design Constraints:

- * There should be a limitation for no. of books a user can borrow from 3-5.
- * at a time.
- * The library card should have unique library Id number for each user.

7) Non-functional requirements:

- * Availability: Each book should be available for users. for their Good Experience.
- * Book maintenance: The books should be maintained properly & Arranged in Order for Easy Search
- * Efficiency: The data entry should be Efficient to store the data. Easily In Software.
- * Performance: The librarian should be very fast to enter the data in Software & staff should be active. To manage the library

8) Preliminary Schedule and Budget.

Library construction: 5cr

website development: 75 lacks

website maintenance: 25 lacks

Staff maintenance: 2 cr

Book Purchase: 5 cr.

Class Diagram

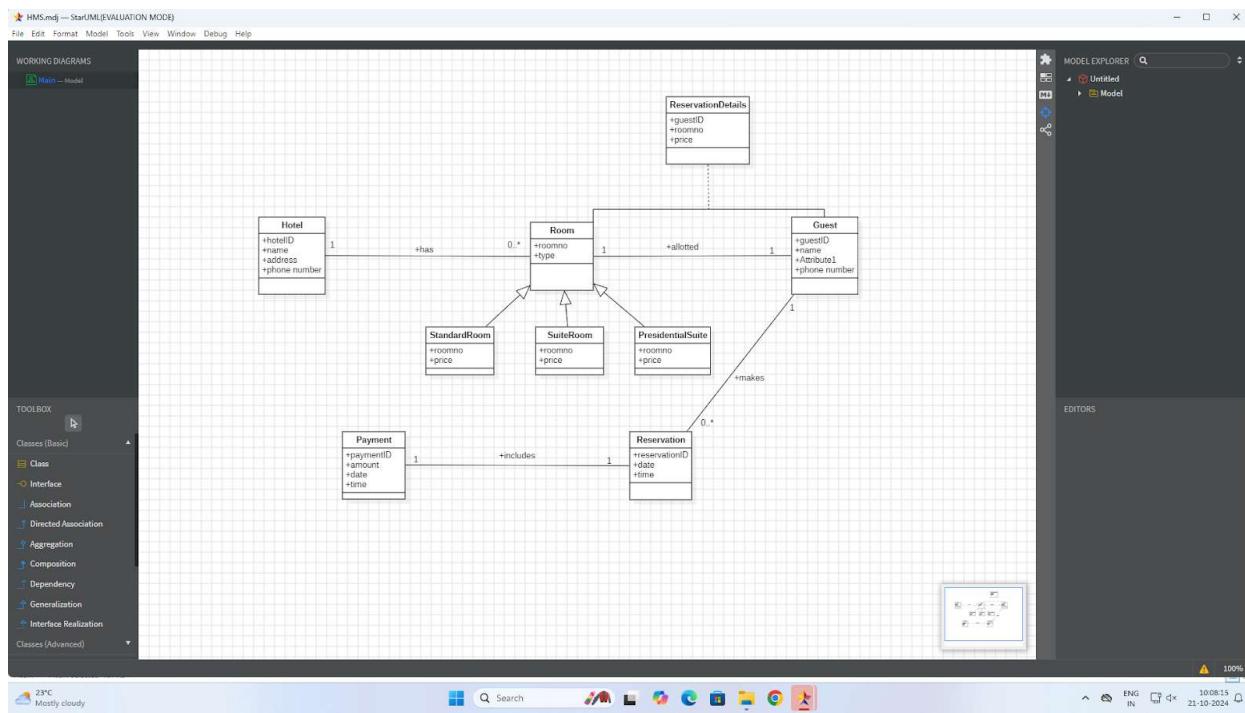


Fig 3.1 Library Management System -Class Diagram

The class diagram represents a library management system, showcasing entities like Library, Librarian, User, Book, Account, and Transaction. The Library handles the system's operations, managed by a Librarian who adds, updates, and deletes books. Users are divided into Students and Faculty, each with functionalities like issuing, renewing, and returning books, managed via their respective Accounts that track borrowed books and fines. Books store details like title, author, and status, while Transactions record borrowing and returning activities. The relationships between these entities ensure seamless management of books, users, and transactions.

State Diagram

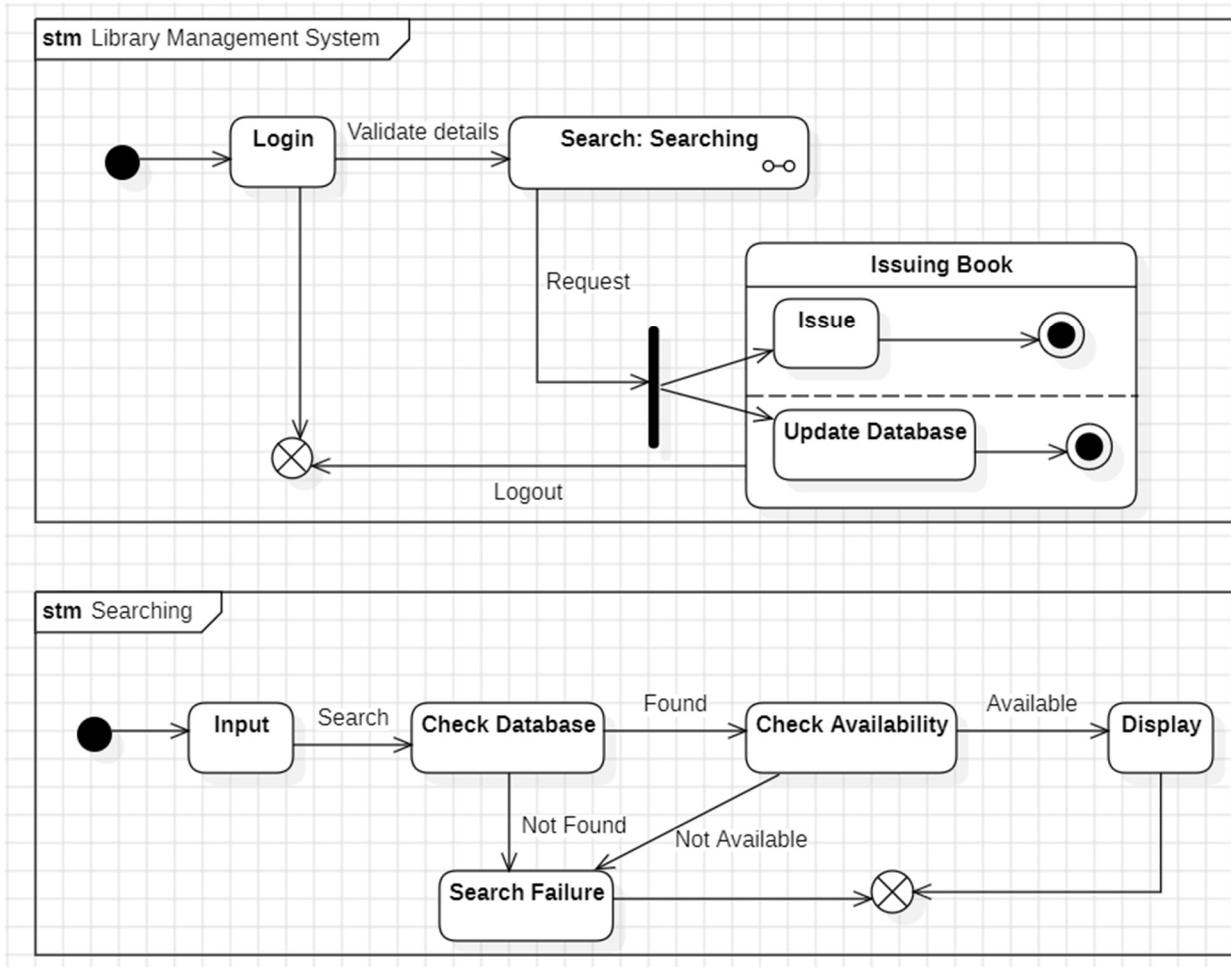


Fig 3.2 Library Management System - State Diagram

The state diagram illustrates the workflow of a library management system. It begins with user login, followed by credential validation. Upon successful login, the system enters the "Searching" state, where the user can search for books. The search process involves checking the database for matches and then checking availability. If a book is available, its details are displayed to the user, who can then request to issue it. The system updates its database accordingly. If the search yields no results or the book is unavailable, the system transitions to the "Search Failure" state. At any point, the user can log out of the system.

Use Case Diagram

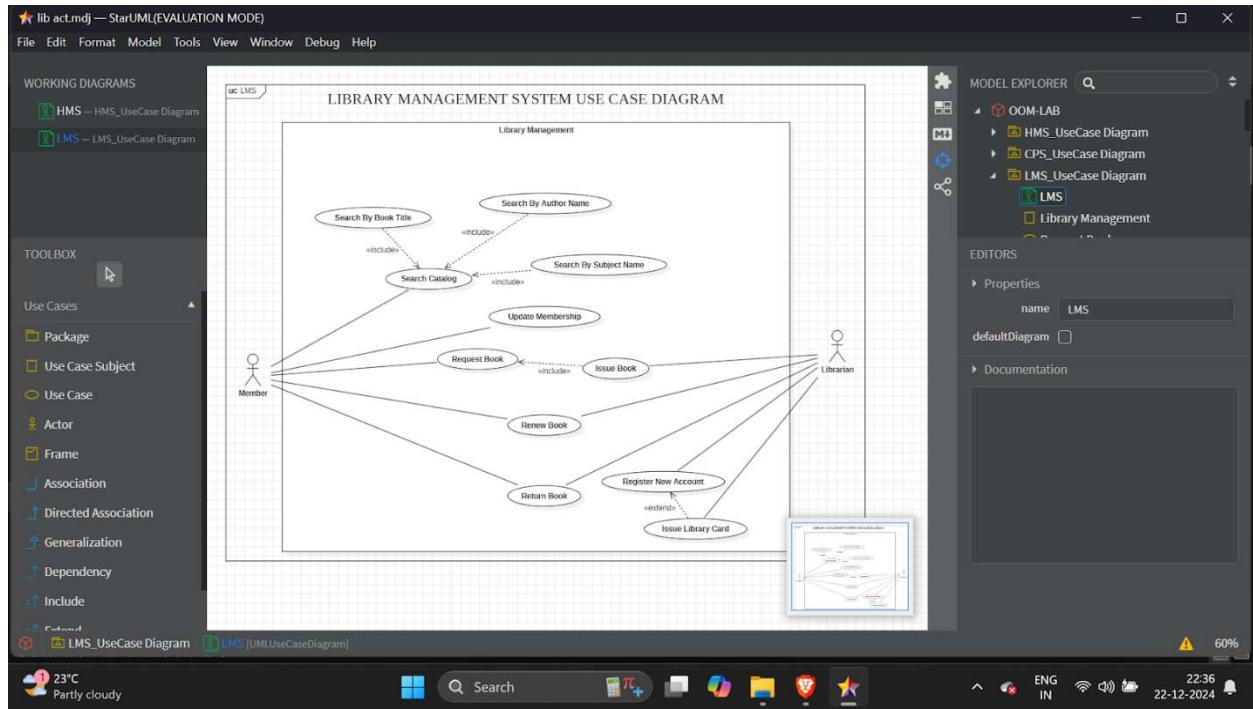


Fig 3.3 Library Management System - Use Case Diagram

The Library Management System is designed to manage the library's resources and user interactions. The system has three main actors: User, Librarian, and Database. The User can reserve books, request new books, pay fines, and register as a new user. The Librarian can add books to the catalog, update the catalog, delete books, and send overdue notifications. The Database stores and manages all the information related to the library, users, and books. The system includes use cases for authentication, filling up forms, and getting library IDs, which are further elaborated by the "include" relationships. This system aims to streamline library operations and provide a convenient experience for users.

Sequence Diagram

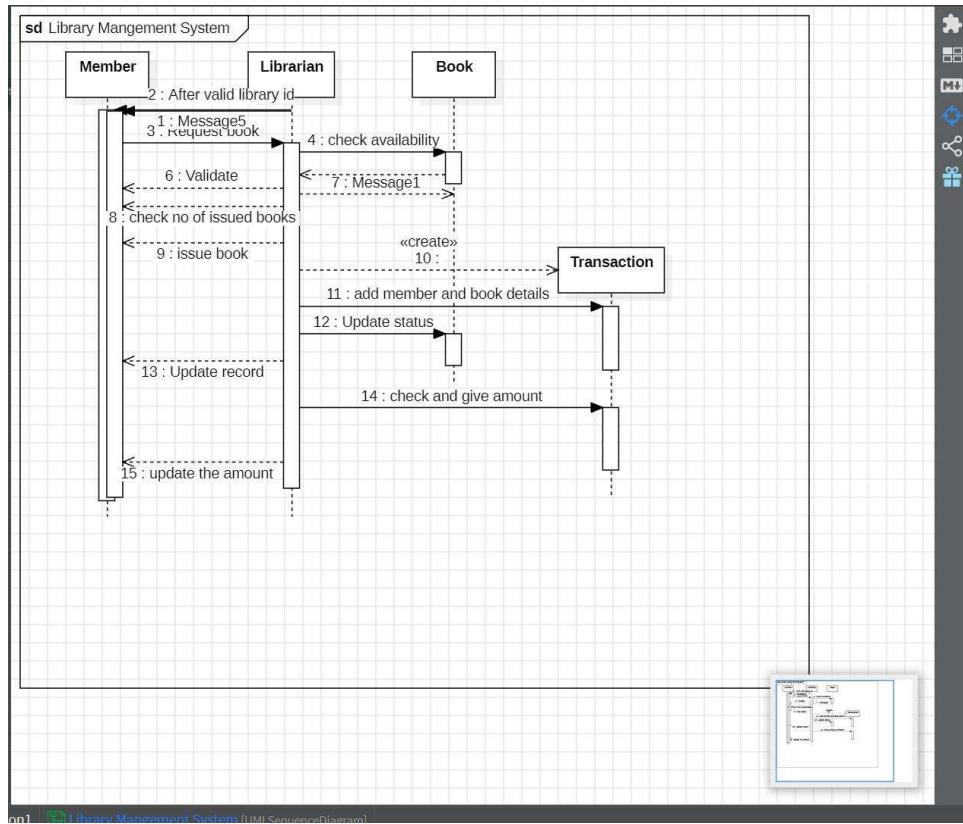


Fig 3.4 Library Management System - Sequence Diagram

The sequence diagram illustrates the process of a user borrowing a book from the library. The user begins by searching for a book in the library catalog. The catalog then searches for the book and returns the results to the user. The user then requests to borrow the book, and the library system checks its availability. If the book is available, the system issues the book to the user and updates its records. Finally, the user receives a receipt confirming the checkout. This diagram highlights the automated steps involved in the process and the interactions between the user and the library system.

Activity Diagram

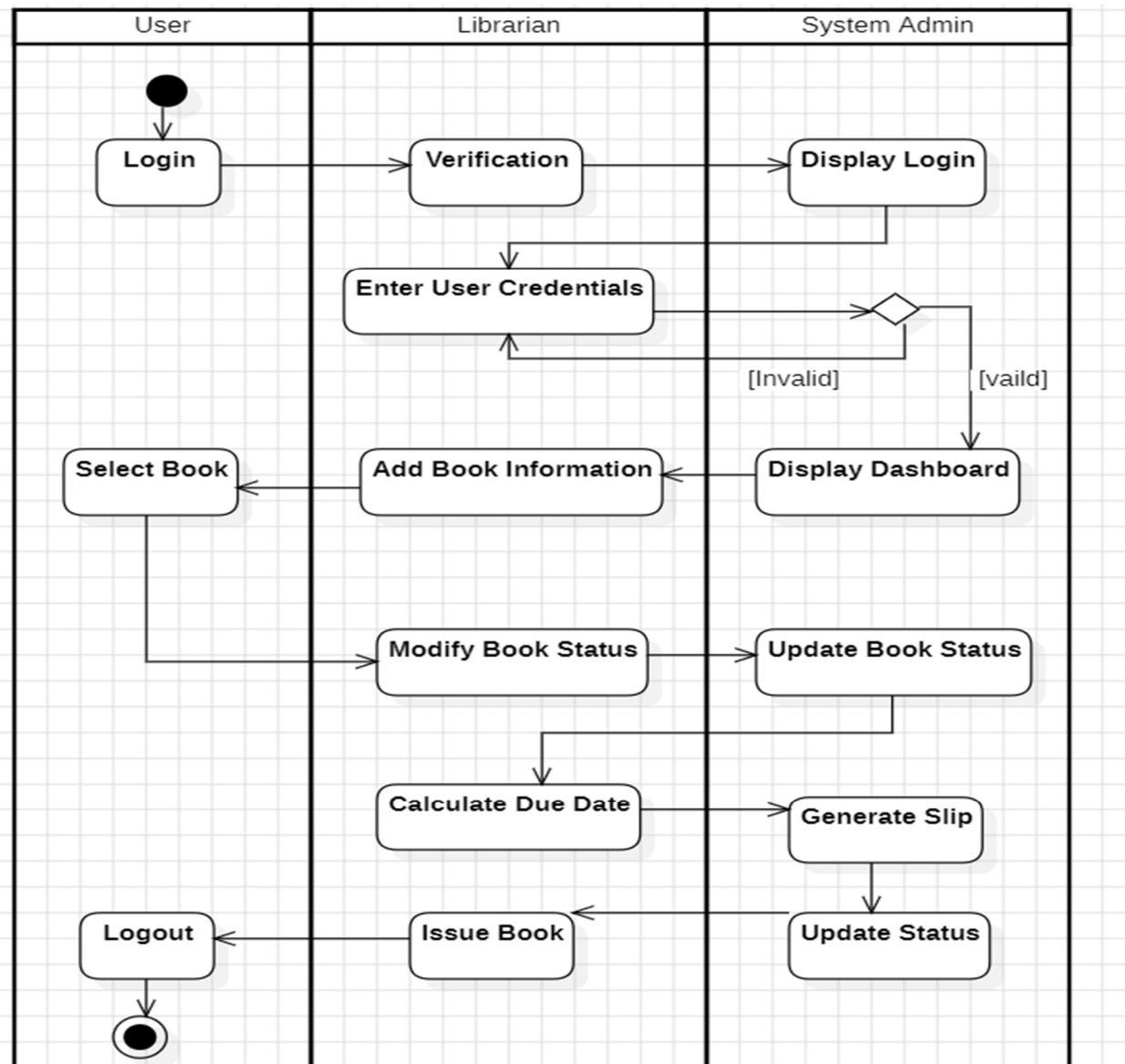


Fig 3.5 Library Management System - Activity Diagram

The activity diagram outlines the workflow of a library management system. It starts with a user logging in, followed by credential verification. Successful login grants access to book selection for the user. Simultaneously, librarians can add new books or modify existing book information. System administrators possess the authority to update book statuses and generate slips related to book transactions. Upon book selection, the system calculates the due date and issues the book to the user, updating the database accordingly.

4. Stock Maintenance System

Software Requirement Specification

5) Stock maintenance system	
1) Introduction:	<p>1.0.1) Purpose of this document: The purpose of this document is to provide detailed specification for a stock maintenance system that helps business track inventory, manage product stock levels.</p> <p>1.0.2) Scope of this document: The system is designed to manage stock for business of all sizes, providing functionalities for stock tracking, order processing & report generation.</p> <p>1.0.3) Overview: This document provides requirements of stock maintenance systems. functionalities user instructions, system architecture. The system will help business manage their inventory efficiently & reduce manual processes.</p>
2) General description:	<p>Stock maintenance system is a stand alone application designed to automate stock management tasks such as tracking inventory levels, managing product orders.</p> <p>→ The system will be web based enabling access from multiple devices including desktop, laptops & mobile devices.</p>
3) Functional requirements:	<p>→ The system will allow users to create, view and update product information.</p>

including SICV, product name, a category & stock levels

- users can log stock movements into the system when receiving new inventory or when products are sold to other locations.
- The system will provide real-time stock tracking
 - g allowing users to monitor inventory levels, stock ageing & usage trends.

4) Interface requirements:-

- Interface should be very informative with lots of graphs showing the recent trends in respective stocks
- Interface must have user profile where user can see, and maintain his stocks.

5) Design constraints:

- The system should handle growing number of products, suppliers & stock movements of business expands.
- Role-based access control will be implemented to ensure that only authorised users can access certain data or perform sensitive stock.
- The system should be able to process stock updates, generate reports & handle multiple users.

6) Performance requirements:

- The system should support up to 50000 concurrent users without noticeable delay in loading.
- All stock & released transactions should be processed within 2-3 seconds.
- It should be able to show live stock value anytime.

7) Non-functional requirements:

- All user data including stock information & supplier information should be stored securely.
- The system must be easy to use with an intuitive user interface designed to minimize errors.
- The system should maintain Real Time management with Regular back-ups of database to prevent data loss.

8) Preliminary schedule & budget

- The development of stock maintenance system should take up to 6 months
- website application development - 20 lacs
- Application verification - 5 lacs
- application maintenance - 5 lacs

Class Diagram

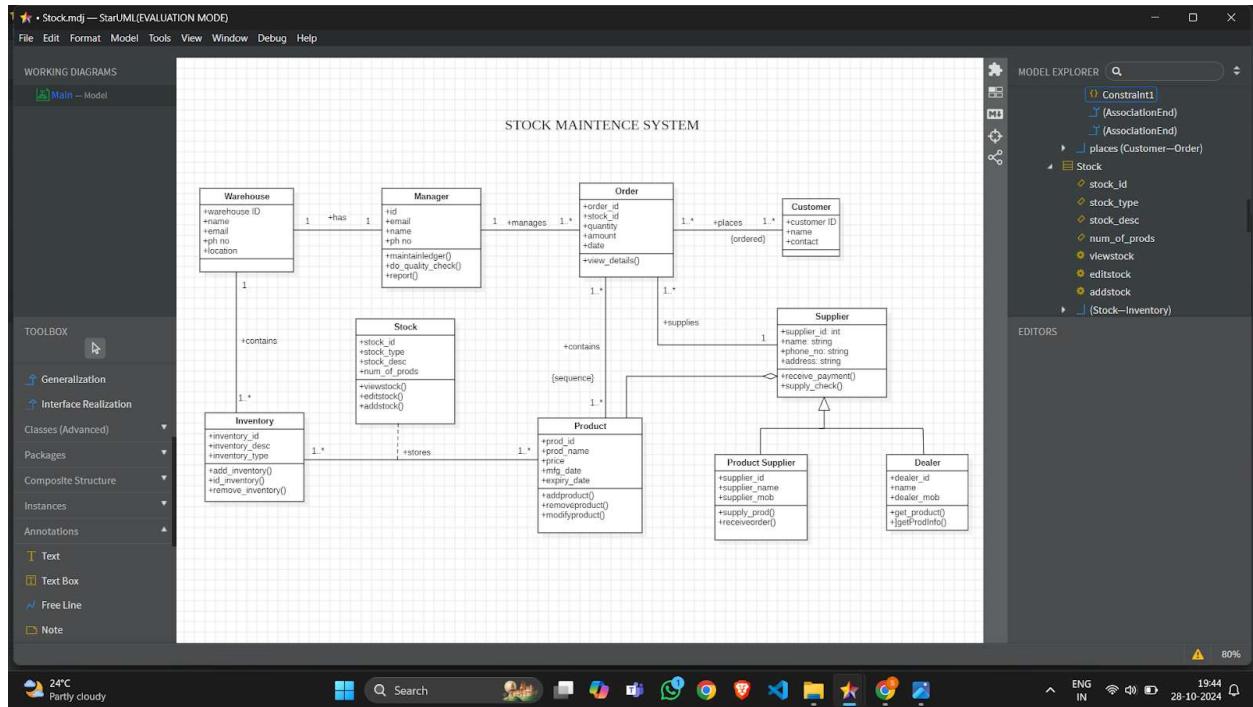


Fig 4.1 Stock Maintenance System - Class Diagram

The class diagram illustrates a warehouse inventory and order management system. The Warehouse contains multiple Inventory items, managed by a Manager who oversees operations like reporting and quality checks. Stock stores details about products, which are managed with functionalities like addition and modification. Orders placed by Customers link products to quantities and amounts, while Suppliers, including Product Suppliers and Dealers, handle the supply of products to the warehouse. The diagram highlights the interactions between inventory, stock, orders, and suppliers within the system.

State Diagram

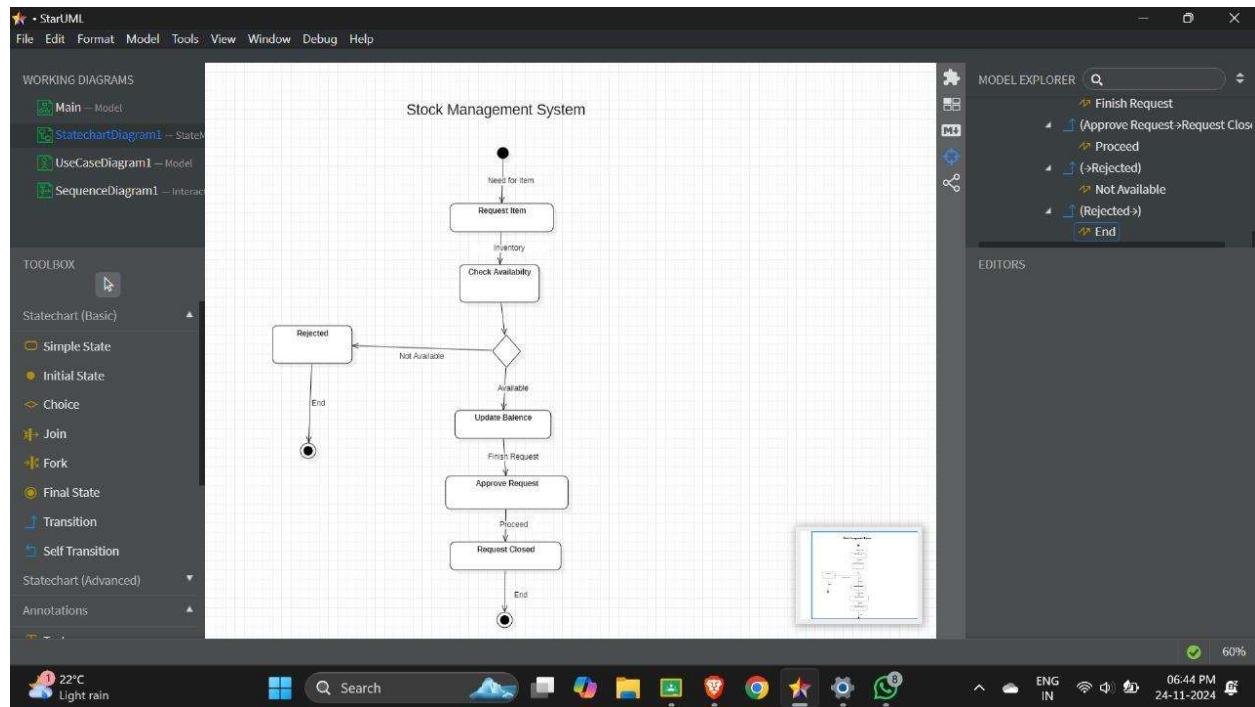


Fig 4.2 Stock Maintenance System - State Diagram

The state diagram illustrates the stock maintenance system's workflow. It starts with a user logging in. The system then checks current stock and inventory levels. If stock is low, the system transitions to the "Order Placement" state, where order details are confirmed and supplier details are obtained. After the order is confirmed, the system moves to the "Order Processing" state, where the order is processed and dispatched. During order processing, stock is deducted and inventory is updated. Finally, the system transitions to the "Payment" state, where payment is authorized and verified. Upon successful payment, the system returns to the "Check Stock" state to monitor inventory levels.

Use Case Diagram

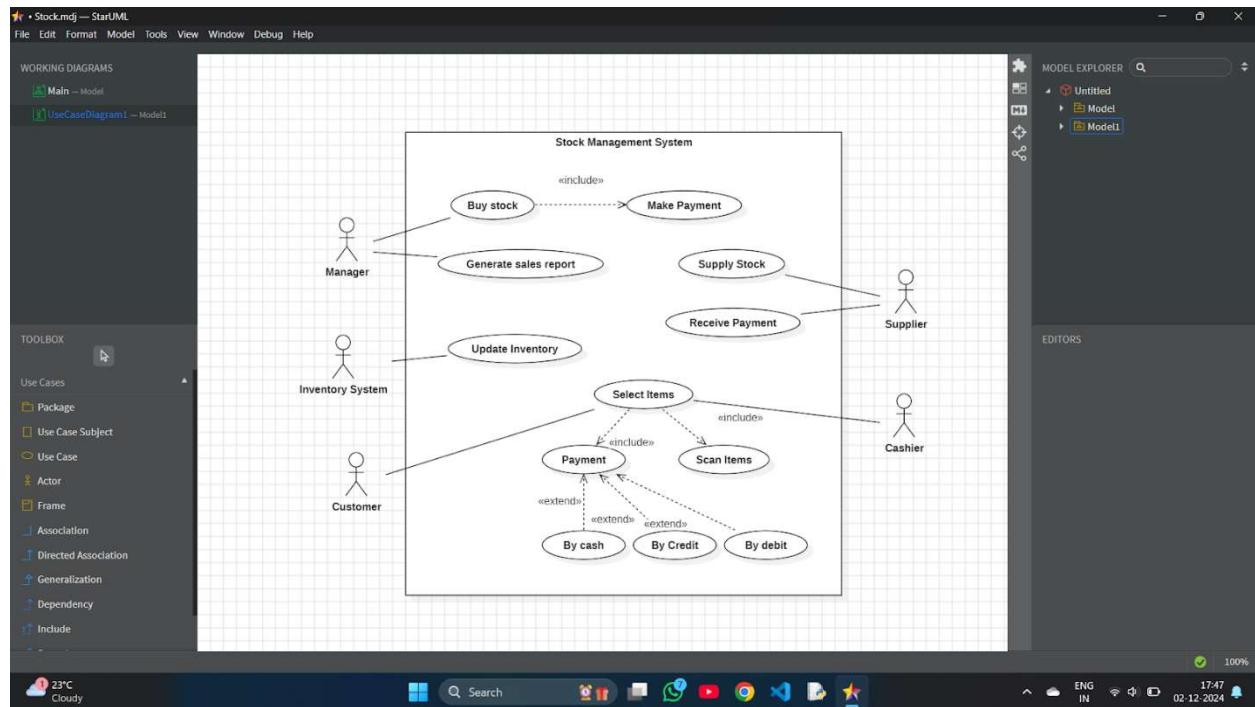


Fig 4.3 Stock Maintenance System - Use Case Diagram

The Stock Maintenance System is designed to manage inventory and customer interactions for a business. The system has three main actors: Stock Manager, Warehouse Manager, and Customer. The Stock Manager can process customer details, check customer records, and identify regular customers. They can also check the expiry of stocks and flag expired products. The Warehouse Manager can add new items to the inventory and check flagged products. The Customer can browse the catalog, place orders, and make payments. The system includes use cases for order details and payment, which are further elaborated by the "include" relationships. This system aims to streamline inventory management and provide a seamless experience for customers.

Sequence Diagram

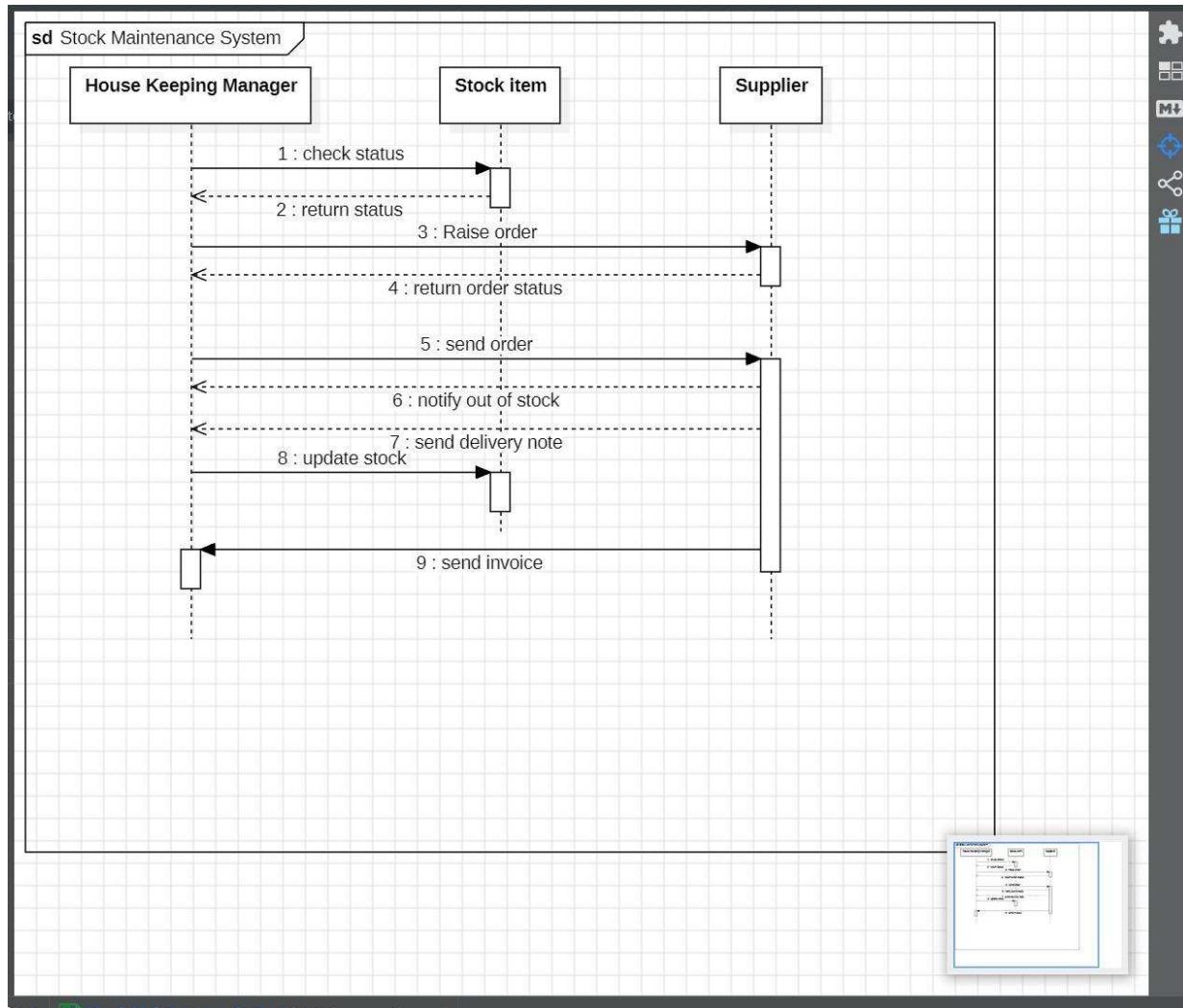


Fig 4.4 Stock Maintenance System - Sequence Diagram

The sequence diagram outlines the order fulfillment process in a stock maintenance system. It begins with the customer placing an order. The manager then checks stock availability, and the system verifies stock levels in the database. If stock is sufficient, the order is created and processed. If stock is insufficient, the system checks with suppliers, places orders, and updates stock levels once the replenishment is complete. The system communicates order status updates to the customer throughout the process. This diagram illustrates the interactions between the customer, manager, database, and suppliers, highlighting the steps involved in fulfilling an order effectively.

Activity Diagram

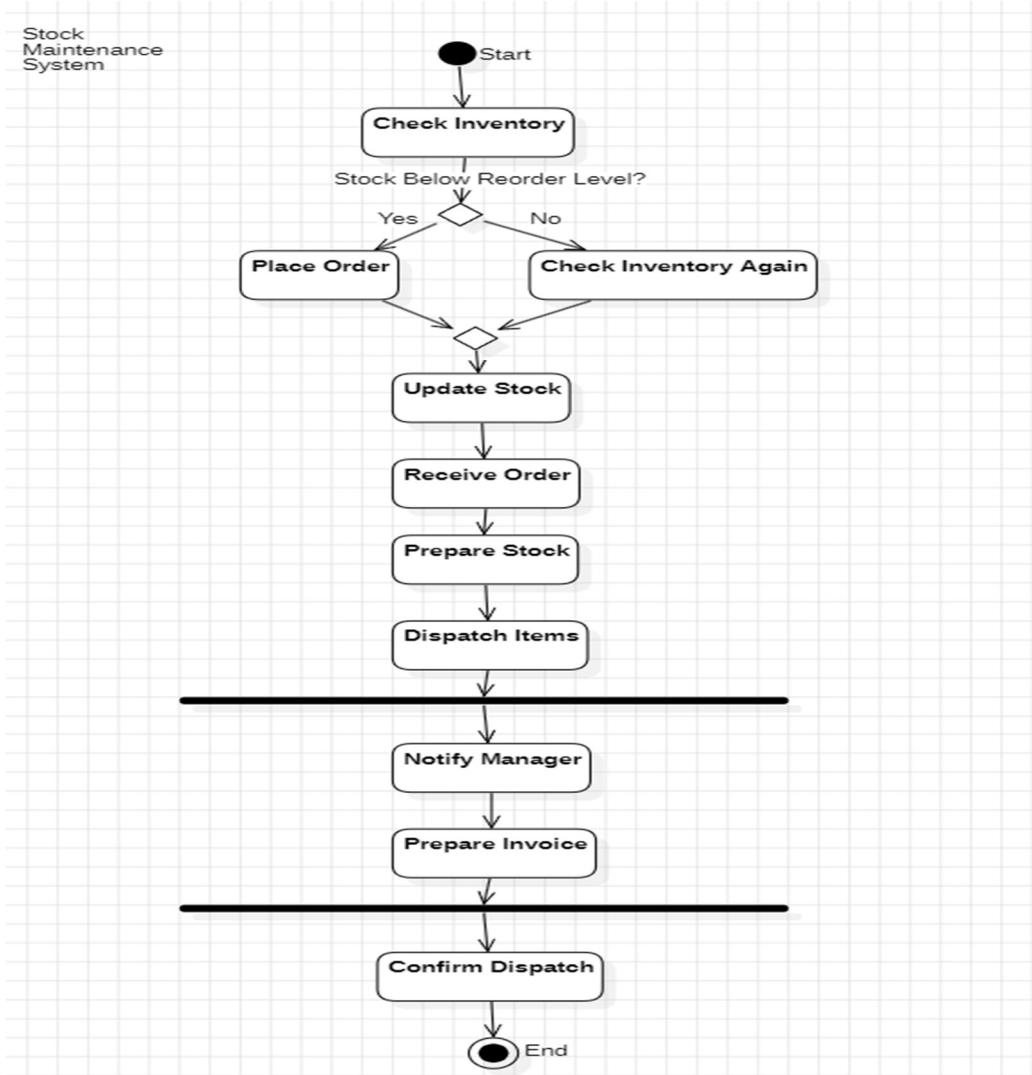
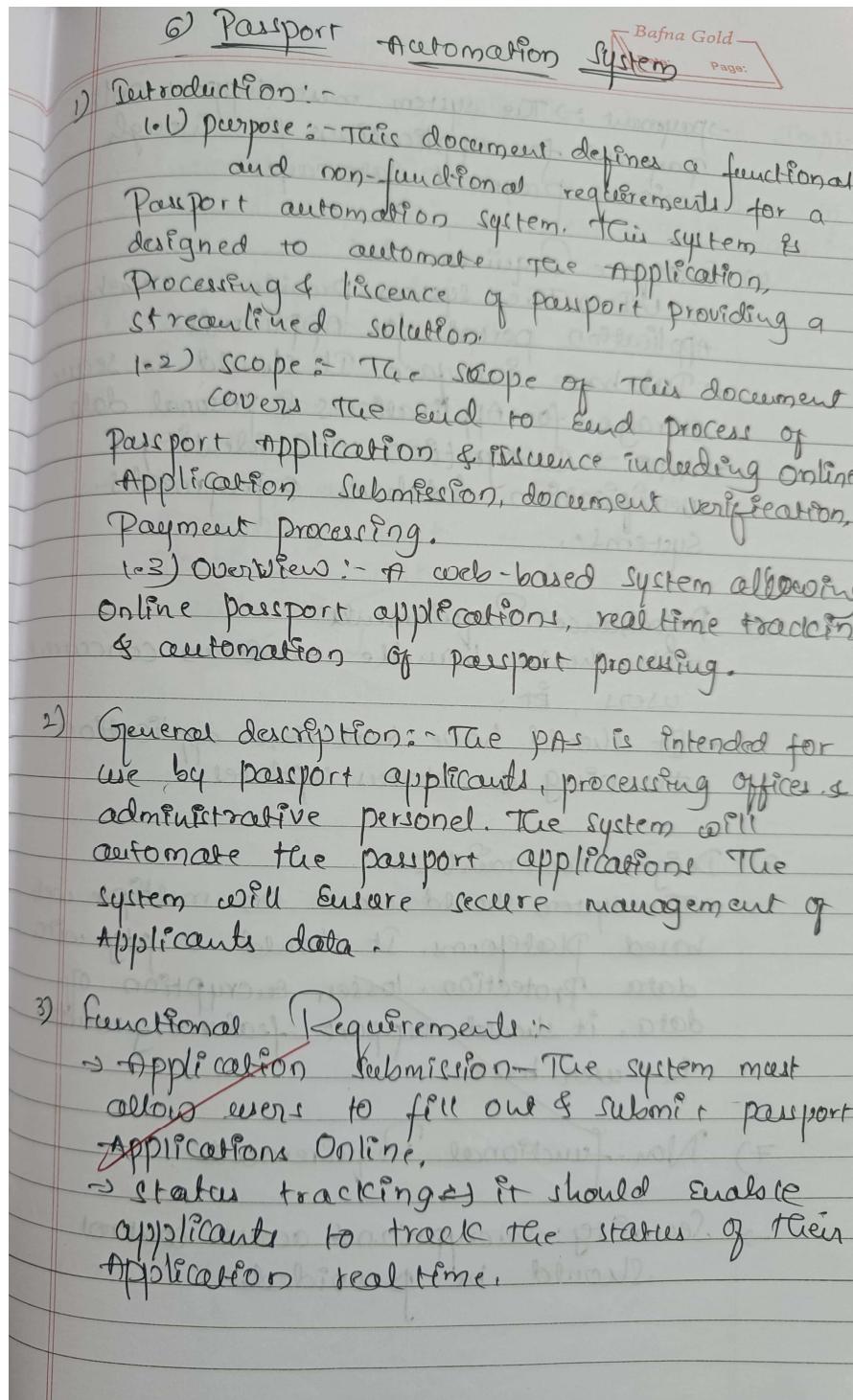


Fig 4.5 Stock Maintenance System-Activity Diagram

This activity diagram represents the workflow of a stock maintenance system. It begins with checking inventory levels, followed by a decision point to assess if stock is below the reorder level. If stock is low, an order is placed, otherwise, inventory is rechecked. Upon receiving the order, the stock is updated, prepared, and dispatched. The process then involves notifying the manager, preparing the invoice, and confirming the dispatch, concluding the workflow.

5. Passport Automation System

Software Requirement Specification



→ payment :- The system must support online payment for applications for through various payment gateways.

4) Interface Requirements:

- UI → The system provide a web-based application portal for applicants.
- Database interface: It should be secure storage for applications & personal data
- API → Integration with identity verification databases & payment systems.

5) Performance Requirements:

- It should handle maximum concurrent users, etc.
- It should have real time application tracking & status update.

6) Design constraints:-

- The system should work on multiple web based platforms. It should comply with data protection laws, encryption of data, it should support feature growth in user base.

7) Non-functional Requirements:

- Security:- Encryption & access control should be provided.

→ **usability**: The application should be simple, & user friendly.

→ **scalability**: It should be capable of handling multiple users & their data.

8) **preliminary schedule & budget**:-

→ Requirement analysis (3 weeks)

Design (3 weeks)

Development → 10 weeks.

Testing → 2 weeks

→ Development cost - 5 lacks

Designing cost - 2 lacks

Testing cost - 1 lack

Maintaining cost - 4 lacks

Class Diagram

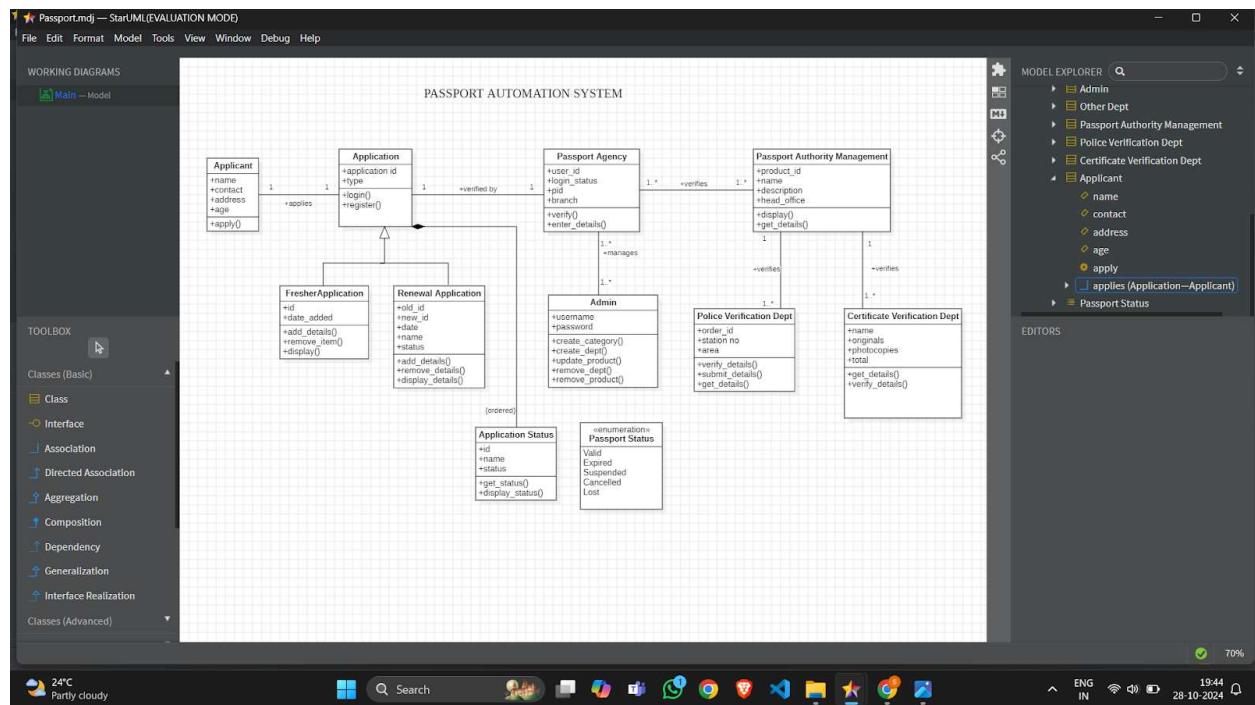


Fig 5.1 Passport Automation System - Class Diagram

The class diagram depicts the structure of a passport application and verification system. It illustrates various entities, such as Applicant, Application, and its specialized forms: FresherApplication and RenewalApplication. The Application class is associated with Applicant, who can apply and register for passport services. The system includes a Passport Agency and its management under Passport Authority Management, which oversees verification processes through departments like Police Verification and Certificate Verification. The diagram also involves an Admin class responsible for managing categories, departments, and products. Key features include status tracking through Application Status and Passport Status enumeration. Relationships between classes are depicted with multiplicity, inheritance, and composition, highlighting functionalities like verifying details, managing applications, and updating statuses.

State Diagram

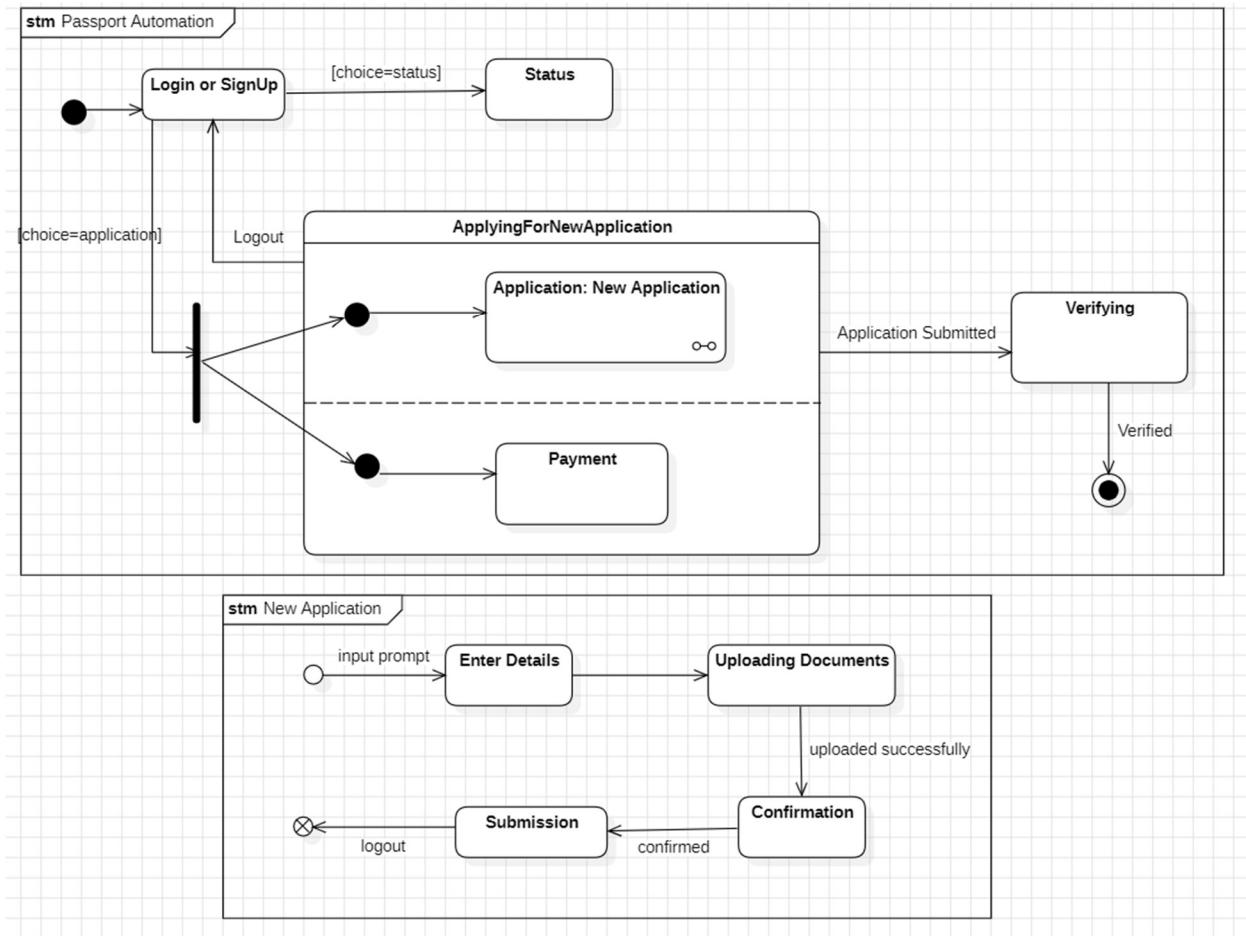


Fig 5.2 Passport Automation System - State Diagram

The state diagram illustrates the passport automation system. The system starts with the user logging in or signing up. After login, the user can choose to check the status of their application or apply for a new one. If the user chooses to apply, they enter the "ApplyingForNewApplication" state. Within this state, the user fills out the application form, uploads documents, and submits the application. Once submitted, the application enters the "Verifying" state. If the application is verified successfully, the user receives a confirmation. The user can also log out at any point during the process.

Use Case Diagram

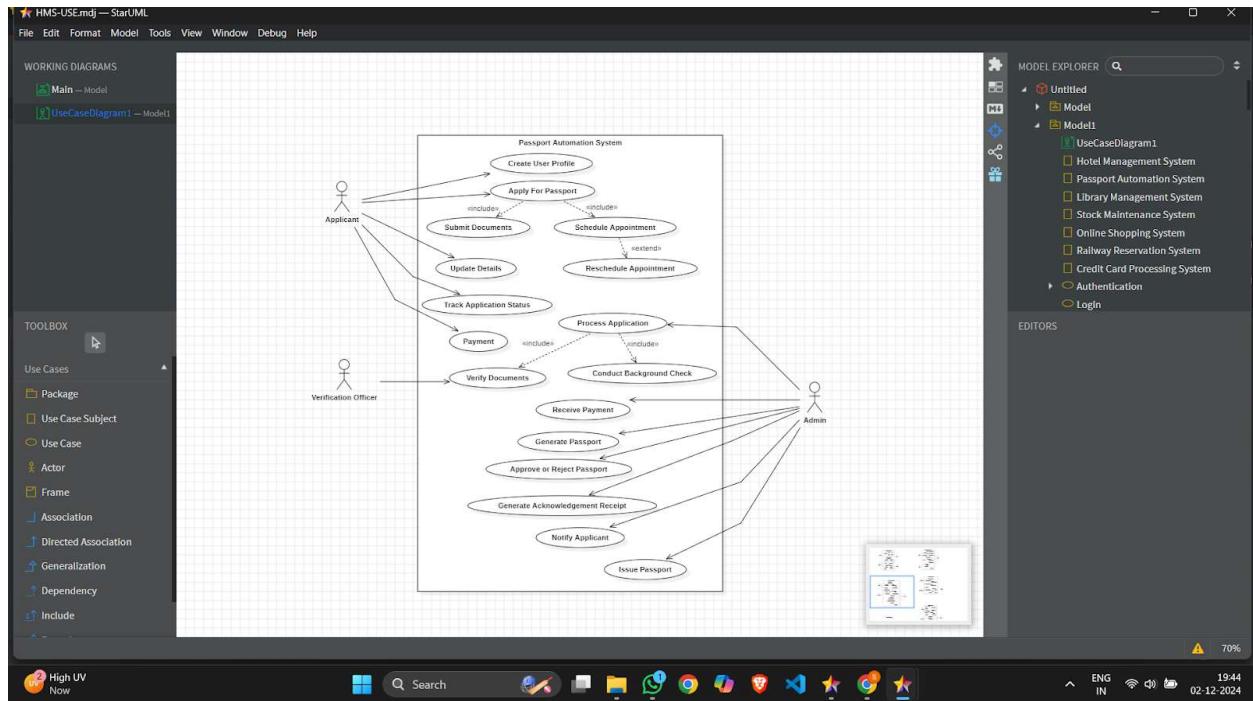


Fig 5.3 Passport Automation System - Use Case Diagram

The diagram illustrates a Use Case Diagram for a Passport Automation System, outlining the interactions between the system and its primary actors: Applicant, Verification Officer, and Admin. The Applicant begins by creating a user profile and applying for a passport, which includes submitting documents, scheduling (or rescheduling) appointments, making payments, and tracking application status. The Verification Officer is responsible for verifying documents and supporting the application processing. The Admin plays a key role in processing applications, conducting background checks, receiving payments, approving or rejecting passport requests, and issuing passports. Additional use cases include generating acknowledgments and notifying applicants of the application's status. This diagram effectively demonstrates the workflow and responsibilities of each actor in the passport issuance process.

Sequence Diagram

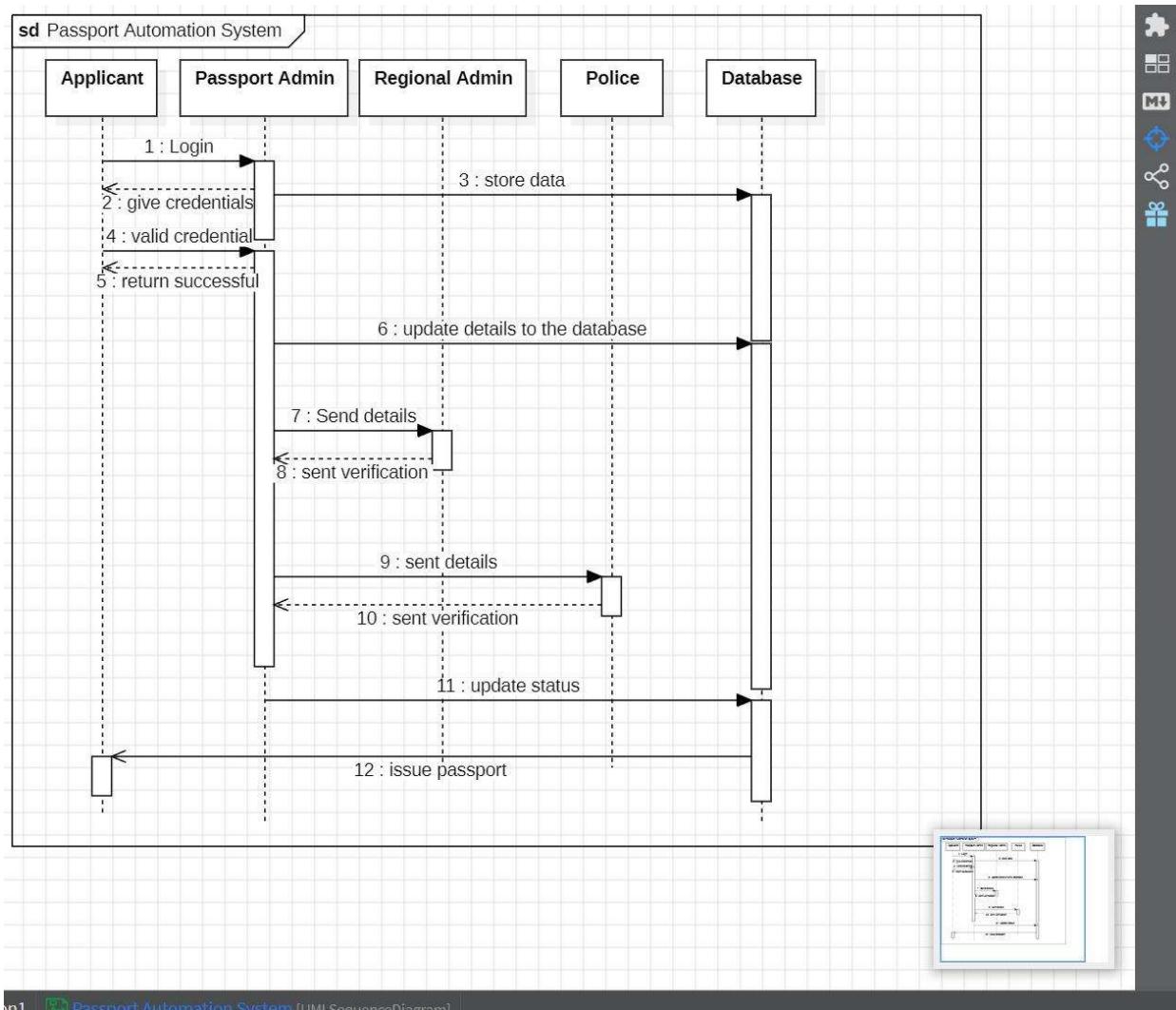


Fig 5.4 Passport Automation System - Sequence Diagram

The sequence diagram illustrates the process of applying for a passport. The applicant starts by submitting an application and then pays the application fee. The payment gateway processes the payment and updates the status. The applicant then uploads the required documents, which are verified by the document verification system. Once the documents are verified, the passport is issued to the applicant. This diagram shows the interactions between the applicant, passport admin, payment gateway, document verification system, and database throughout the passport application process.

Activity Diagram

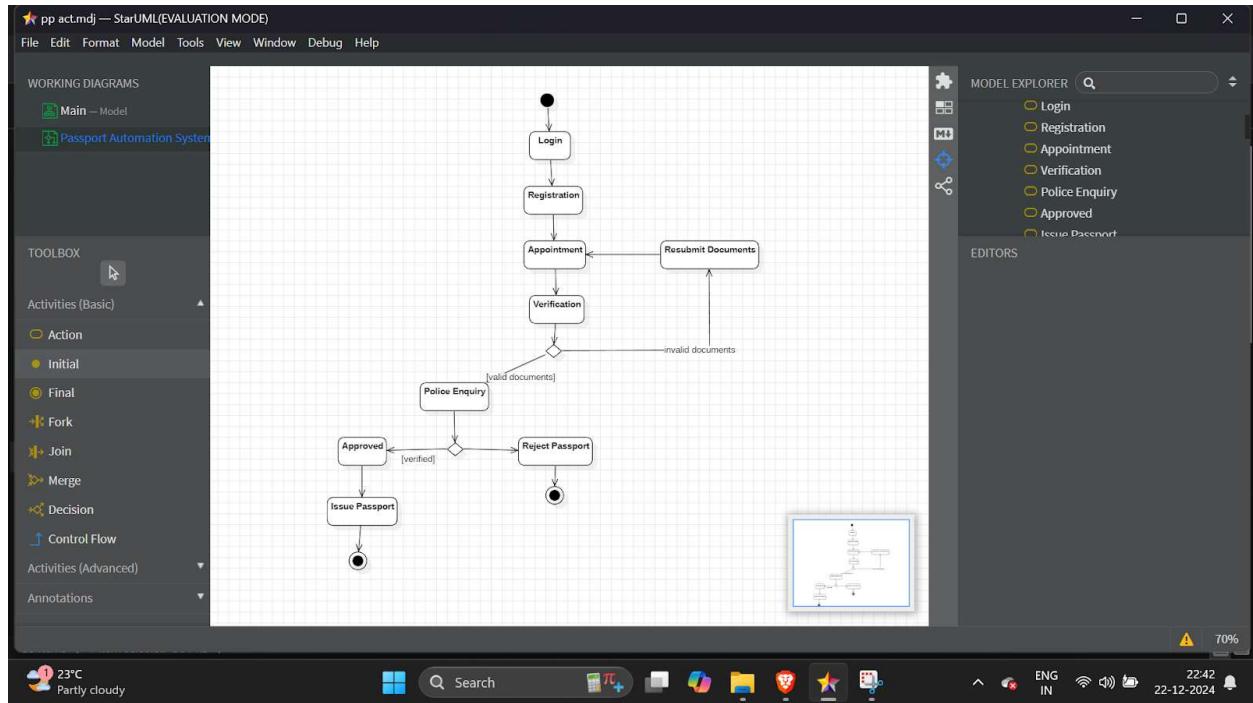


Fig 5.5 Passport Automation System - Activity Diagram

The activity diagram illustrates the passport application process. It starts with the applicant submitting an application. The application system reviews the application. If the application is complete, it proceeds to document verification. If invalid documents are found, the application is rejected. If valid, a background check is conducted. If the background check is clear, the verification is successful, and the passport is issued. If any stage fails, the application is rejected. The applicant can receive the passport once it's issued.