

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



OOMD Mini Project Report on

**‘One Card Management’**

*Submitted in partial fulfillment for the award of degree of*

Bachelor of Engineering  
in  
Computer Science and Engineering

*Submitted by:*

**Nachiketha (1BM21CS109)**  
**Nishant Bhat (1BM21CS116)**  
**Nithin S (1BM21CS120)**  
**P Sai Krishna (1BM21CS123)**

Department of Computer Science and Engineering  
B.M.S. College of Engineering  
Bull Temple Road, Basavanagudi, Bangalore 560 019  
2023-2024

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



***DECLARATION***

We, **Nachiketha (1BM21CS109), Nishant Bhat (1BM21CS116), Nithin S (1BM21CS120), P Sai Krishna (1BM21CS123)** students of 6<sup>th</sup> Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this OOMD Mini Project entitled "One Card Management" has been carried out in Department of CSE, B.M.S. College of Engineering, Bangalore during the academic semester March - July 2024. I also declare that to the best of our knowledge and belief, the OOMD mini Project report is not from part of any other report by any other students.

**Signature of the Candidate**

**Nachiketha (1BM21CS109)**  
**Nishant Bhat (1BM21CS116)**  
**Nithin S (1BM21CS120)**  
**P Sai Krishna (1BM21CS123)**

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



***CERTIFICATE***

This is to certify that the OOMD Mini Project titled “**One Card Management**” has been carried out by **Nachiketha (1BM21CS109), Nishant Bhat (1BM21CS116), Nithin S (1BM21CS120), P Sai Krishna (1BM21CS123)**, during the academic year 2023-2024.

Signature of the Faculty in Charge

## Table of Contents

Sl No	Title
1	Ch 1: Problem statement
2	Ch 2: Software Requirement Specification
3	Ch 3: Class Diagram
4	Ch 4: State Diagram
5	Ch 5: Interaction diagram
6	Ch 6: UI Design with Screenshots

# Chapter 1: Problem Statement

## PROBLEM STATEMENT

The One Card Management System addresses the problem of disorganization and lack of efficiency in managing multiple cards, by providing a secure, centralized platform that enables users to easily store, manage and access information about their cards in one place.

## ANALYSIS

A "one card for all" system refers to a single card that can be used for multiple purposes, such as accessing different bank accounts, paying bills, and making purchases. This type of system offers convenience and streamlines financial management by eliminating the need for multiple cards. However, it also increases the risk of fraud and loss if the card is stolen or misused.

## Chapter 2: Software Requirement Specification

### 1. Introduction

#### *1.1 Purpose of this Document*

The purpose of this document is to detail the requirements for the development of the One Card Management System. This document outlines the functional and non-functional requirements, interfaces, constraints, and other relevant information necessary for the development of this system. The intended audience includes project stakeholders, development team members, and testers.

#### *1.2 Scope of this Document*

The One Card Management System aims to provide a centralized, secure platform for users to store, manage, and access information about their multiple cards. This system addresses the disorganization and inefficiencies associated with managing multiple cards. The scope includes the design, development, testing, and deployment of the system. The development cost and time required will be detailed in the preliminary schedule and budget section.

#### *1.3 Overview*

The One Card Management System will allow users to consolidate information from various cards (such as credit cards, debit cards, membership cards, etc.) into a single, user-friendly platform. Key features include secure storage, easy access, and efficient management of card information. The system will improve user efficiency and organization.

### 2. General Description

#### *2.1 Product Perspective*

The One Card Management System is a new, standalone application that aims to enhance users' ability to manage multiple cards. It will interact with various external card issuers' systems through secure APIs to retrieve and update card information.

#### *2.2 Product Functions*

- **Card Storage:** Securely store information for multiple cards.
- **Card Management:** Enable users to add, update, and delete card information.
- **Access Control:** Provide secure access to card information using authentication mechanisms.
- **Notifications:** Send alerts and notifications for card usage, expiration dates, and other important events.
- **Reporting:** Generate reports on card usage and other metrics.

## *2.3 User Characteristics*

The target users are individuals who own multiple cards and seek an efficient way to manage them. Users should be familiar with basic digital interfaces and possess a moderate level of technical proficiency.

## *2.4 Constraints*

- Compliance with relevant data protection and privacy laws (e.g., GDPR, CCPA).
- Secure storage and transmission of sensitive card information.
- Compatibility with major card issuers' systems.

## *2.5 Assumptions and Dependencies*

- Users have access to a stable internet connection.
- Card issuers provide APIs for integration.
- Users possess devices compatible with the application (smartphones, tablets, or computers).

# 3. Functional Requirements

## *3.1 User Authentication*

- The system shall require users to register with a unique username and password.
- The system shall support multi-factor authentication (MFA) for enhanced security.

## *3.2 Card Information Management*

- The system shall allow users to add new cards by entering relevant details.
- The system shall allow users to update existing card information.
- The system shall allow users to delete card information.

## *3.3 Secure Storage*

- The system shall encrypt card information stored in the database.

## *3.4 Notifications*

- The system shall notify users of card expiration dates 30 days in advance.
- The system shall alert users to unusual card activity.

## *3.5 Reporting*

- The system shall generate monthly usage reports for each card.
- The system shall provide downloadable and printable reports.

## 4. Interface Requirements

### *4.1 User Interfaces*

- The system shall provide a responsive web interface accessible from various devices.
- The system shall offer a mobile application for iOS and Android platforms.

### *4.2 Software Interfaces*

- The system shall integrate with external card issuer APIs to fetch and update card information.

### *4.3 Hardware Interfaces*

- No specific hardware interfaces are required.

## 5. Performance Requirements

- The system shall support up to 10,000 simultaneous users without performance degradation.
- The system shall load the main dashboard within 2 seconds under standard conditions.
- The system shall handle up to 1,000 card additions/updates per minute.

## 6. Design Constraints

- The system shall use AES-256 encryption for data storage.
- The system shall be developed using a scalable architecture to support future growth.
- The system shall comply with PCI DSS standards for handling card information.

## 7. Non-Functional Attributes

### *7.1 Security*

- The system shall enforce strong password policies.
- The system shall undergo regular security audits and vulnerability assessments.

### *7.2 Reliability*

- The system shall have 99.9% uptime.
- The system shall provide data backup and recovery mechanisms.

### *7.3 Usability*

- The system shall have an intuitive user interface with easy navigation.
- The system shall provide user guides and tutorials.



## *7.4 Scalability*

- The system shall be designed to scale horizontally to accommodate increased user load.

## 8. Preliminary Schedule and Budget

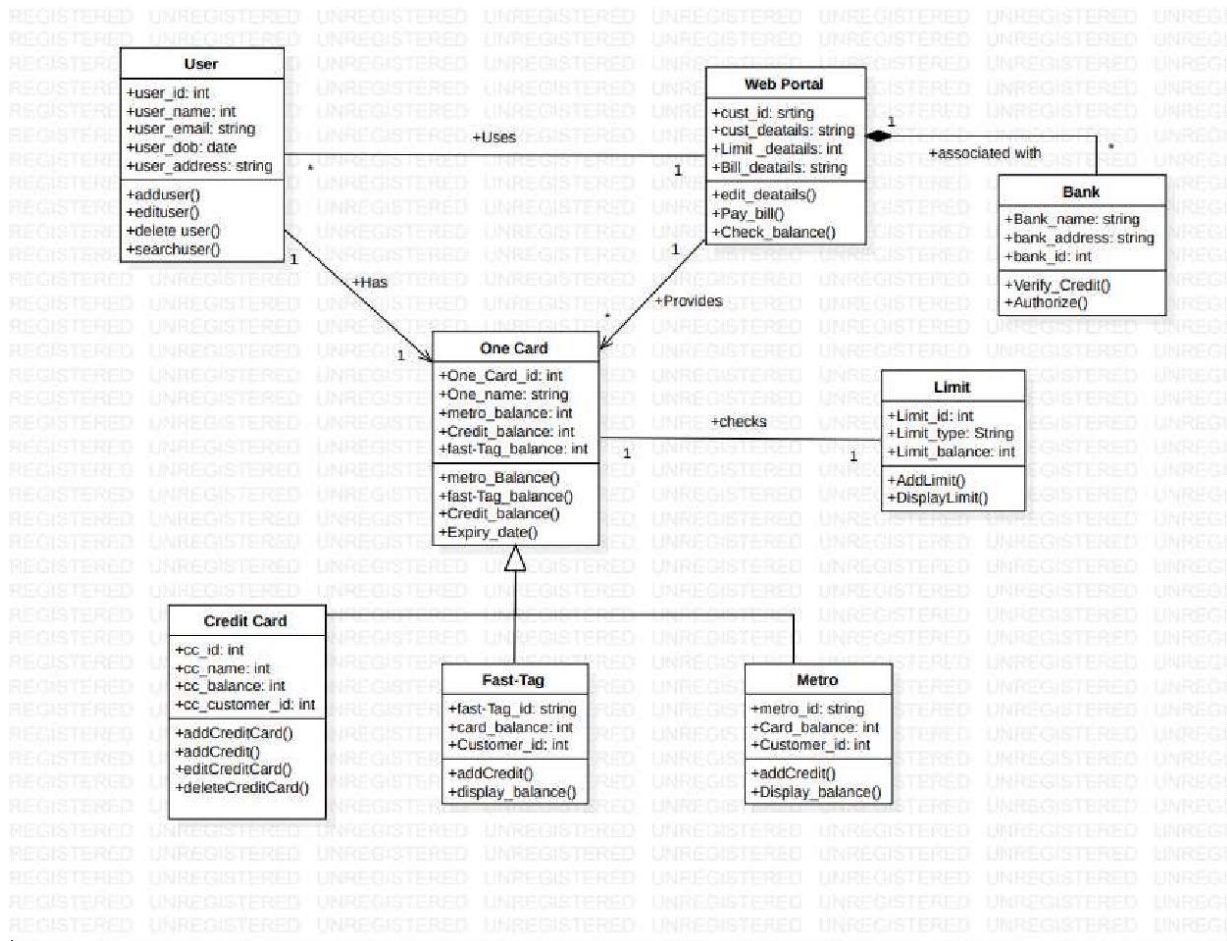
### *8.1 Schedule*

- **Requirements Gathering:** 2 weeks
- **Design Phase:** 4 weeks
- **Development Phase:** 12 weeks
- **Testing Phase:** 4 weeks
- **Deployment:** 2 weeks

### *8.2 Budget*

- **Development Costs:** \$200,000
- **Testing Costs:** \$50,000
- **Deployment Costs:** \$30,000
- **Total Estimated Cost:** \$280,000

## Chapter 3: Class Modeling



### □ User

#### • Attributes:

- user\_id: Unique identifier for the user.
- user\_name: Name of the user.
- user\_email: Email address of the user.
- user\_dob: Date of birth of the user.
- user\_address: Address of the user.

#### • Methods:

- addUser(): Method to add a new user.
- editUser(): Method to edit an existing user's information.
- deleteUser(): Method to delete a user.
- searchUser(): Method to search for a user.

- **Relevance:** This class manages user-related information and operations, such as adding, editing, deleting, and searching for users.

## □ One Card

- **Attributes:**
  - One\_Card\_id: Unique identifier for the One Card.
  - One\_name: Name of the One Card.
  - metro\_balance: Balance for metro card.
  - Credit\_balance: Balance for credit card.
  - fast\_Tag\_balance: Balance for Fast-Tag.
- **Methods:**
  - metro\_Balance(): Method to manage metro card balance.
  - fast\_Tag\_balance(): Method to manage Fast-Tag balance.
  - Credit\_balance(): Method to manage credit card balance.
  - Expiry\_date(): Method to check the expiry date.
- **Relevance:** Central class that consolidates different types of card balances and operations related to the One Card.

## □ Credit Card

- **Attributes:**
  - cc\_id: Unique identifier for the credit card.
  - cc\_name: Name of the credit card.
  - cc\_balance: Balance of the credit card.
  - cc\_customer\_id: Customer identifier linked to the credit card.
- **Methods:**
  - addCreditCard(): Method to add a new credit card.
  - addCredit(): Method to add credit to the card.
  - editCreditCard(): Method to edit credit card details.
  - deleteCreditCard(): Method to delete a credit card.
- **Relevance:** Manages operations and information specific to credit cards.

## □ Fast-Tag

- **Attributes:**
  - fast\_tag\_id: Unique identifier for the Fast-Tag.
  - card\_balance: Balance of the Fast-Tag.
  - Customer\_id: Identifier for the customer linked to the Fast-Tag.
- **Methods:**
  - addCredit(): Method to add credit to the Fast-Tag.
  - display\_balance(): Method to display the balance of the Fast-Tag.
- **Relevance:** Manages operations and information specific to Fast-Tags used for toll payments.

## □ Metro

- **Attributes:**

- metro\_id: Unique identifier for the metro card.
- card\_balance: Balance of the metro card.
- Customer\_id: Identifier for the customer linked to the metro card.
- **Methods:**
  - addCredit(): Method to add credit to the metro card.
  - display\_balance(): Method to display the balance of the metro card.
- **Relevance:** Manages operations and information specific to metro cards.

#### □ **Limit**

- **Attributes:**
  - Limit\_id: Unique identifier for the limit.
  - Limit\_type: Type of limit (e.g., daily, monthly).
  - Limit\_balance: The balance for the limit.
- **Methods:**
  - addLimit(): Method to add a new limit.
  - displayLimit(): Method to display the limit information.
- **Relevance:** Manages and sets limitations on card usage.

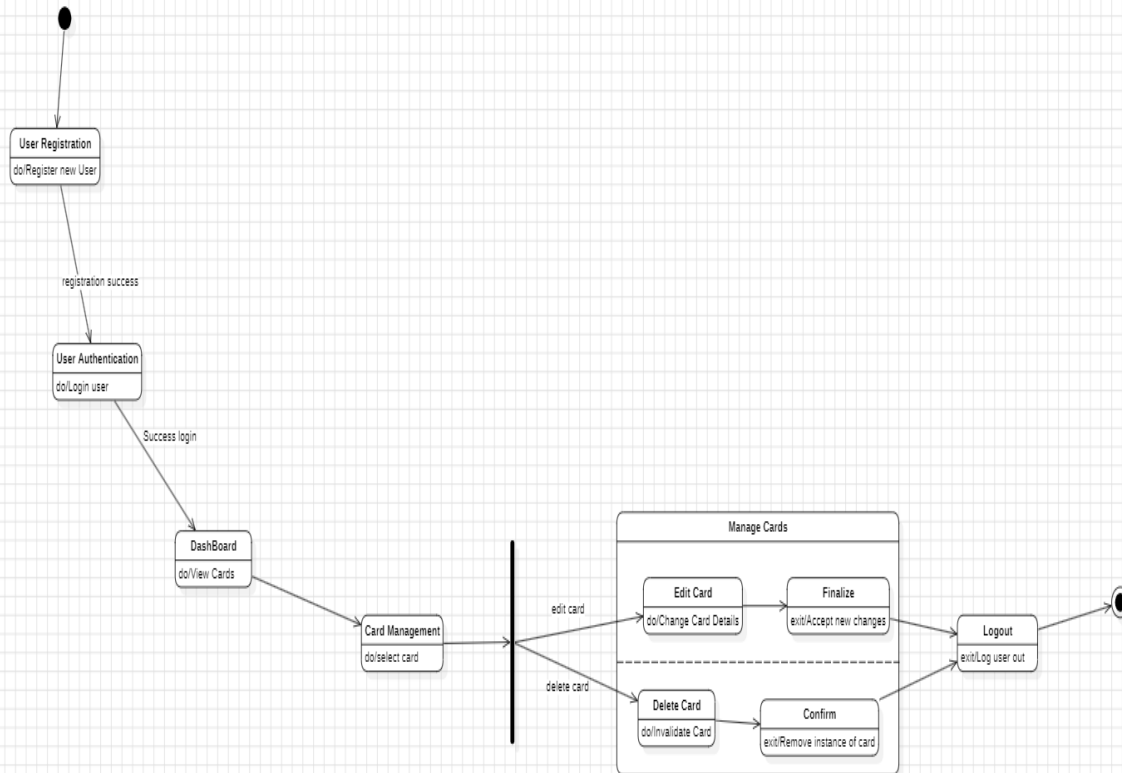
#### □ **Web Portal**

- **Attributes:**
  - cust\_id: Customer identifier.
  - cust\_details: Details of the customer.
  - Limit\_details: Details of the limits associated with the customer.
  - Bill\_details: Details of the bills associated with the customer.
- **Methods:**
  - edit\_details(): Method to edit customer details.
  - pay\_bill(): Method to pay bills.
  - check\_balance(): Method to check balance.
- **Relevance:** Provides an interface for customers to manage their cards, check balances, and pay bills online.

#### □ **Bank**

- **Attributes:**
  - Bank\_name: Name of the bank.
  - bank\_address: Address of the bank.
  - bank\_id: Unique identifier for the bank.
- **Methods:**
  - verify\_Credit(): Method to verify credit transactions.
  - authorize(): Method to authorize transactions.
- **Relevance:** Represents banks that issue cards and authorize transactions.

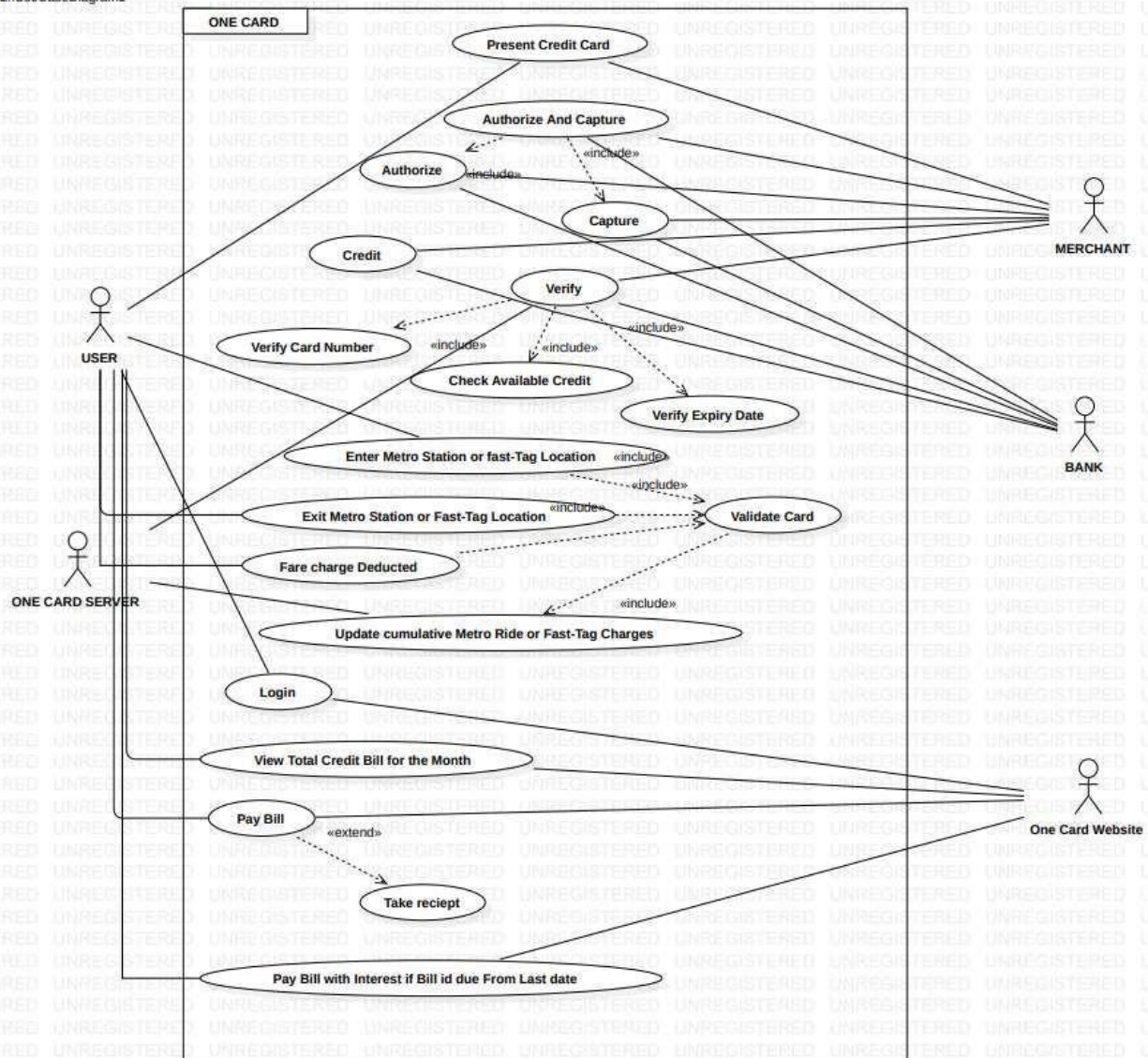
## Chapter 4: State Modeling



## Chapter 5: Interaction Modeling

### Use case diagram

Model2::UseCaseDiagram1



#### Set of actors:

The users who interact with the system as in this case, the one card operations and management, form the set of actors. The actors who play a major role in the one card system are as follows:

- Bank
- User

- One card website
- Merchant
- One card server

#### Use Cases:

1. Present Credit card- The user presents the One card to the merchant.
2. Authorize and capture- The payment Gateway is captured and authorized by the means of Bank.
3. Check available Credit- the available credit is checked before processing (checking if the purchase is not above credit limit)
4. Enter Metro Station/ Fast-Tag- The card now works on RFID technology and the credit limit for separate metro and fast-tag will be created.
5. Fare charge Deducted- The charge will be deducted and this charge will be accumulated in the users web portal account.
6. Login- The user has to login to the web portal to access his account.
7. View total bill for the month- The bill for the month will be generated at the end of the month including all the charges.
8. Pay bill with interest – The user has to pay bill with interest if the bill has not been paid before the stipulated time.

#### Use Case Description:

Use case name: Pay Bill

Participating actors: Initiated by user, validated by web portal, verified by bank.

Flow of events:

- 1) User initiates payment of bill from his/ her account.
- 2) Web portal send the user OTP, who enters it to complete transaction.
- 3) The website verifies payment and provides receipt to the user.

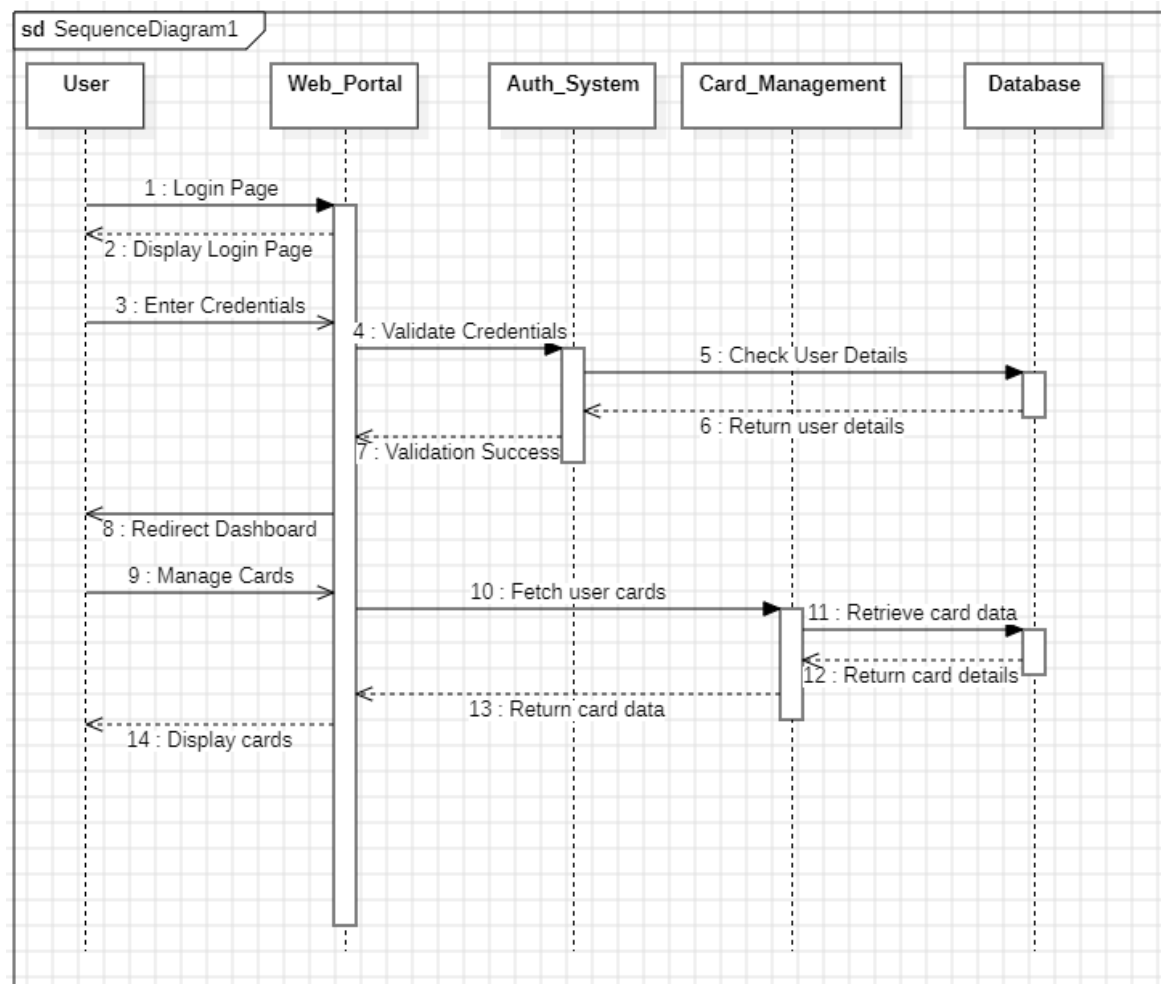
Entry condition: User initiates Bill payment.

Exit condition: The passenger has received an acknowledgment from the admin in the form of an SMS confirmation.

Quality requirements:

- 1) OTP for verification should be sent in less than 30 seconds.
- 2) The receipt should include necessary information of the user and with information that the user has paid bill before time or not which will later affect the credit score of users if bill has been defaulted.

## Sequence Diagram



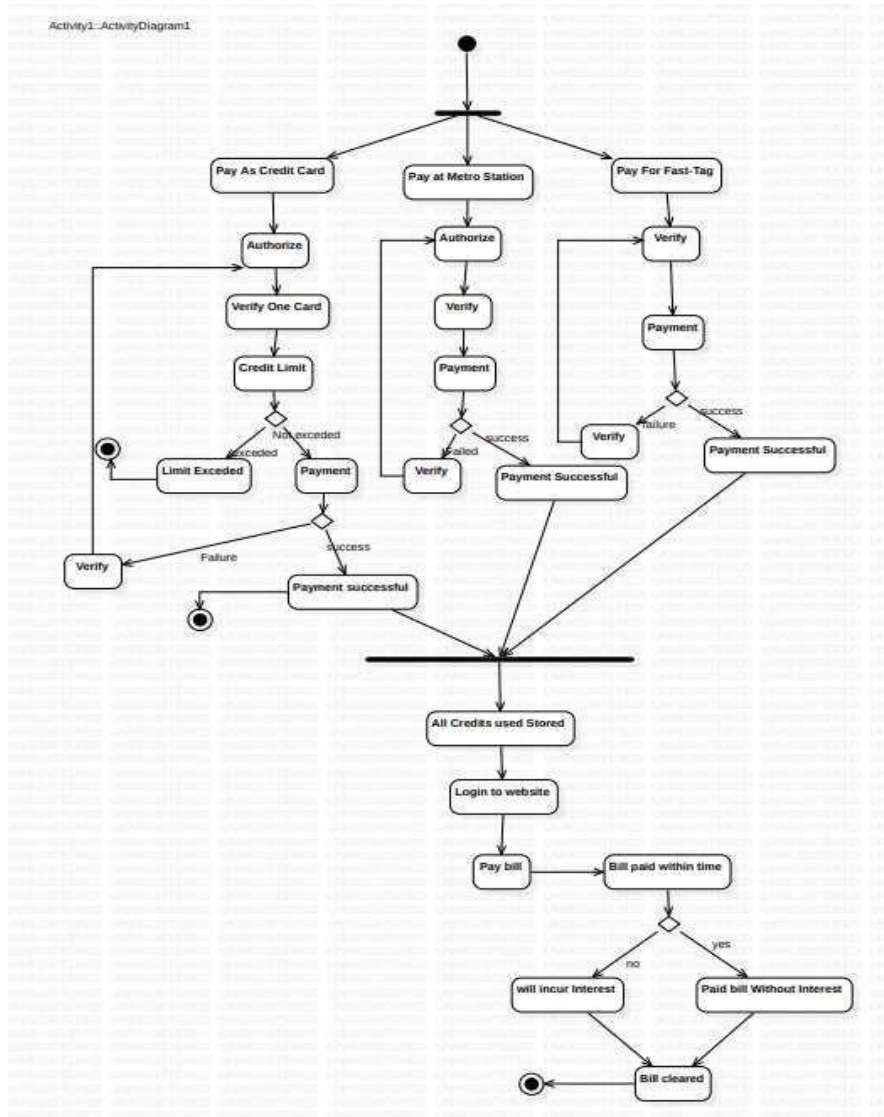
- **User Opens Login Page:** The user opens the login page, and the web portal displays the login interface.
- **User Enters Credentials:** The user enters their login credentials (username and password).
- **Web Portal Validates Credentials:** The web portal sends the credentials to the authentication system for validation.
- **Authentication System Checks Credentials:** The authentication system queries the database to verify the user's credentials.
- **User Validated:** The database confirms the credentials are correct, and the authentication system informs the web portal of the successful validation.
- **Redirect to Dashboard:** The web portal redirects the user to the dashboard after successful login.
- **User Selects Manage Cards:** The user selects the option to manage their cards.
- **Fetch User's Cards:** The web portal requests the card management system to fetch the user's cards.
- **Retrieve Cards from Database:** The card management system queries the database to retrieve the user's card information.



- ☐ **Send Card Data to Web Portal:** The card management system sends the retrieved card data back to the web portal.
- ☐ **Display Cards:** The web portal displays the user's cards.

## Chapter 6: UI Design with Screenshots

### Activity Diagram



```
Enter the name of user:
Arnold
Enter the choice, how you want to use the OneCard as:
1) Metro Card
2) FasTag
3) Credit Card
4) Return Credentials
5) TopUp
6) Exit
3
Enter the amount to be debited:
14000
Can't proceed! Card limit exceeded
Please Topup to proceed:)
Remaining Limit: 2000
```

Enter the choice, how you want to use the OneCard as:

- 1) Metro Card
- 2) FasTag
- 3) Credit Card
- 4) Return Credentials
- 5) TopUp
- 6) Exit

4

User Name: Arnold

Metro Card ID: 10042

FasTag ID: 20068

Credit Card ID: 30035

Remaining Limit: 16000

Attention! confidential data is disclosed

You want to display CW(Y/N):

y

501

Enter the choice, how you want to use the OneCard as:

- 1) Metro Card
- 2) FasTag
- 3) Credit Card
- 4) Return Credentials
- 5) TopUp
- 6) Exit

5

Remaining Limit: 2000

Enter the TopUp amount:

14000

ToppedUp Remaining Limit: 16000

Enter the amount to be debited:

12000

Enter CW:

501

Your money has been debited

Remaining Limit: 4000