VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT on

Object-Oriented Modelling and Design

Submitted by

Samarth C Shetty (1BM19CS141)

in partial fulfilment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING BENGALURU-560019 May-2022 to July-2022

(Autonomous Institution under VTU)

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

<u>CERTIFICATE</u>

This is to certify that the Lab work entitled "LAB COURSE **Object-Oriented Modelling and Design**" carried out by **Samarth C Shetty(1BM19CS141),** who is Bonafede student of **B. M. S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of Object**-Oriented Modelling and Design - (20CS6PCOMD)** work prescribed for the said degree.

Sheetal V A   **Dr. Jyothi S Nayak**
Assistant Professor   Professor and Head
Department of CSE   Department of CSE
BMSCE, Bengaluru   BMSCE, Bengaluru

`

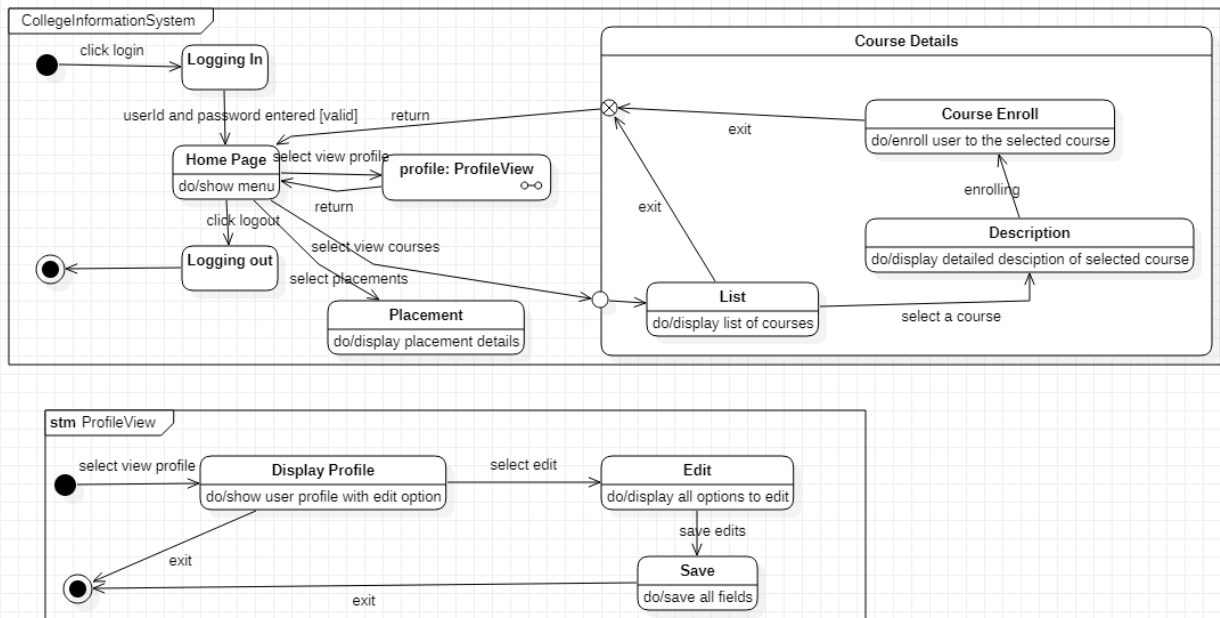## Index Sheet

1. College Information System

1.1   Problem Statement

The College Information System is a system that maintains student, staff and department information. It maintains the courses taught by teachers and students enrolled in them. Admission records of student and Examination details and other important information related to college management is maintained.

1.2 Class Diagram

1.3   State Diagram

CollegeInformationSystem

- click login → **Logging In**
- **Logging In** — userId and password entered [valid] → **Home Page**
- **Home Page** (do/show menu)
- select view profile → **profile: ProfileView**
- **profile: ProfileView** — return → **Home Page**
- return → **Home Page**
- **Home Page** — click logout → **Logging out**
- **Logging out** → (final state)
- select view courses
- select placements → **Placement** (do/display placement details)

Course Details
- **Course Enroll** (do/enroll user to the selected course)
- exit
- enrolling → **Description** (do/display detailed desciption of selected course)
- **List** (do/display list of courses)
- select a course → **Description**
- exit

stm ProfileView

- select view profile → **Display Profile** (do/show user profile with edit option)
- select edit → **Edit** (do/display all options to edit)
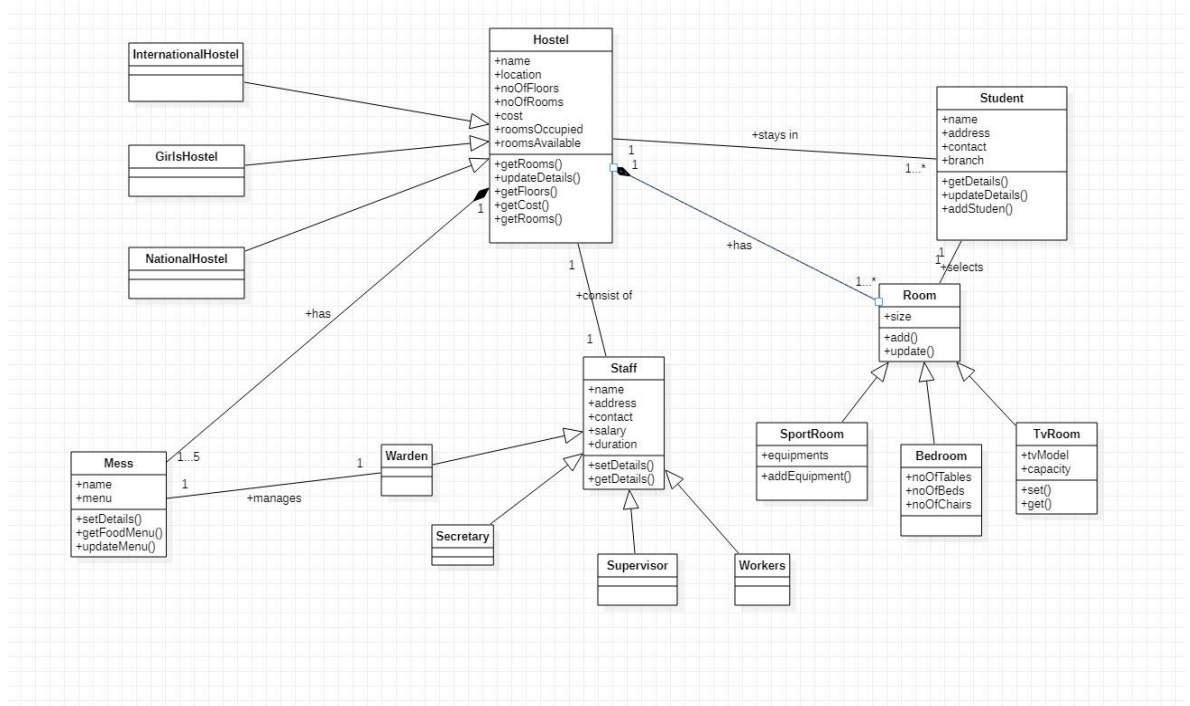- save edits → **Save** (do/save all fields)
- exit
- exit

2. Hostel Management System
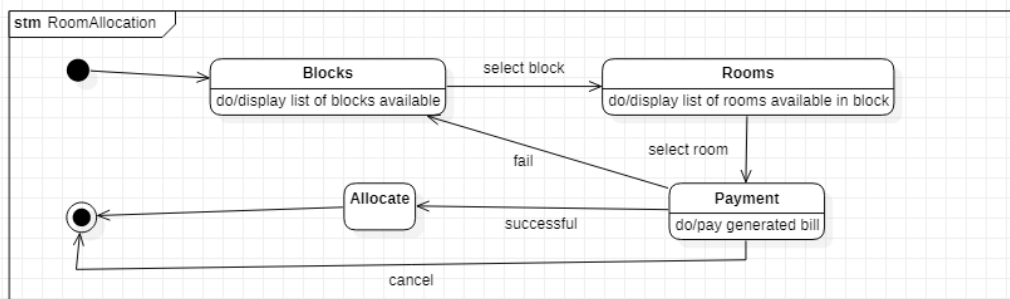
2.1 Problem Statement

The hostel management system is to provide college students accommodation to the university hostel more efficiently. This project also keeps details of the hostelers and applied students. It is headed by Warden. He will be the administrator. This document is intended to minimize human works and make hostel allocation an easier job for students and hostel authorities by providing online application for hostel.

2.2 Class Diagram

2.3
State Diagram

HostelManagementSystem

- (start) → **click login** → **Logging In**
- **Logging In** → userId and password [valid] → **Home Page**
- **Home Page**
  do/display options to apply to hostel and mess details
- **Home Page** → [admin credentials] → **Mess**
- **Home Page** → click logout → **Logging Out** → (end)
- **Home Page** → apply → **room: RoomAllocation** → return → **Home Page**

Mess

- **Mess List**
  do/list all mess details
- **Mess List** → select mess → **Menu**
  do/display menu
- **Menu** → select edit → **Edit Menu**
  do/add or delete item
- **Mess List** → bill generation → **Expenditure**
  do/display detailed expenditure for that month
- **Expenditure** → bill generated → (exit) → exit → **Home Page**
- **Edit Menu** → updation successful → (exit)

stm RoomAllocation

- (start) → **Blocks**
  do/display list of blocks available
- **Blocks** → select block → **Rooms**
  do/display list of rooms available in block
- **Rooms** → select room → **Payment**
  do/pay generated bill
- **Payment** → successful → **Allocate** → (end)
- **Blocks** → fail → (end)
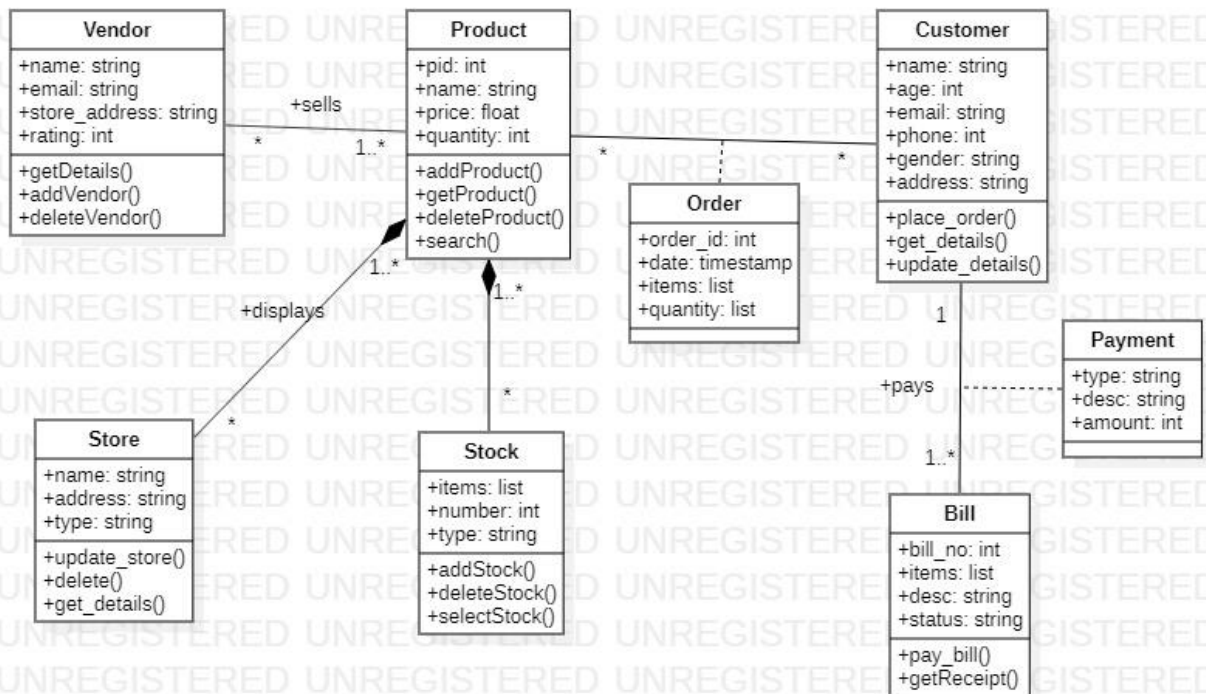- **Payment** → cancel → (end)

7

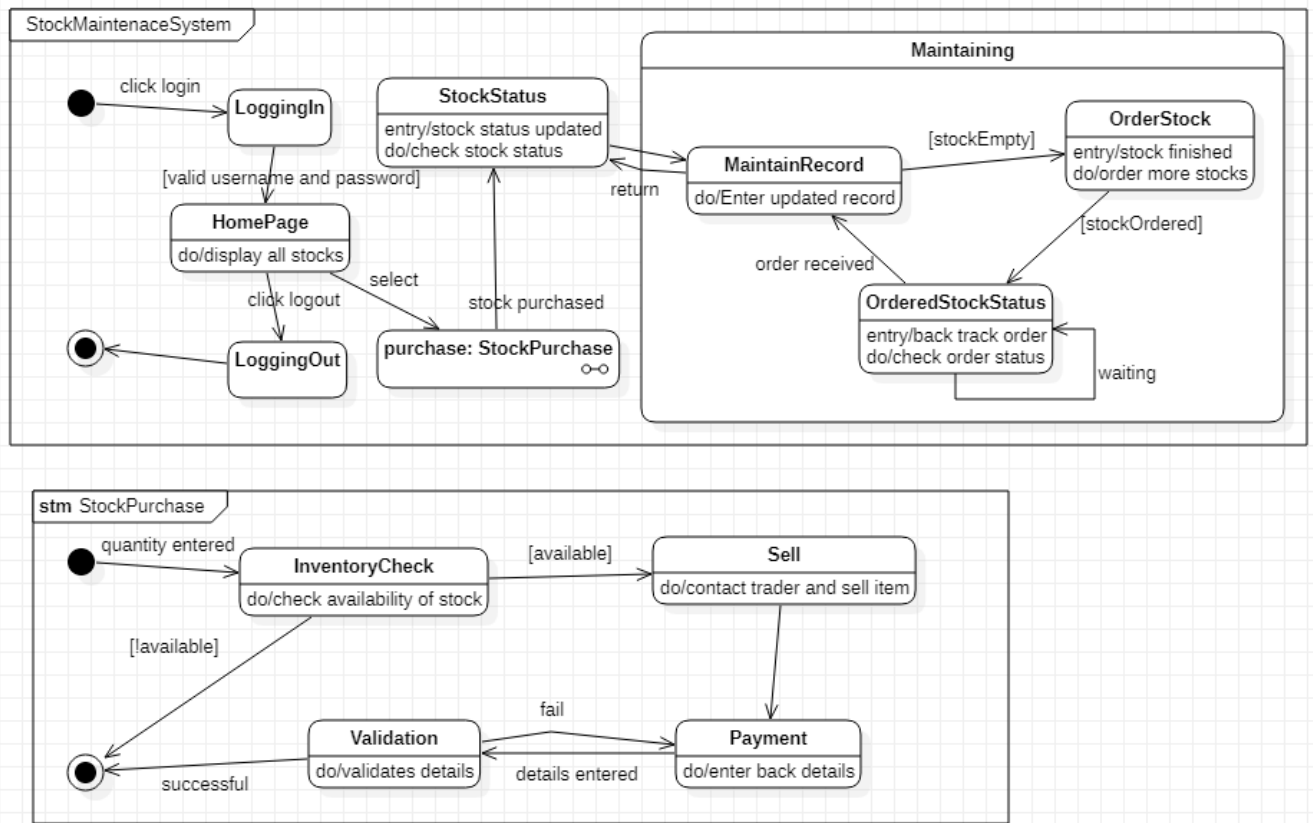### 3. Stock Maintenance System

3.1 Problem Statement

The stock maintenance system is basically for the customers who access the information about the stock and retrieves the information. The stock maintenance system is to replace the existing maintenance system which is in efficient. The new stock maintenance system will allow the employee to record information of the products available in the store .The vendor deals with the information about the details of the suppliers giving product to the organization.

3.2 Class Diagram

3.3   State Diagram

StockMaintenaceSystem

click login → **LoggingIn**

[valid username and password]

StockStatus
entry/stock status updated
do/check stock status

Maintaining

MaintainRecord
do/Enter updated record

[stockEmpty] → **OrderStock**
entry/stock finished
do/order more stocks

return

order received

[stockOrdered]

OrderedStockStatus
entry/back track order
do/check order status

waiting

HomePage
do/display all stocks

select

click logout

stock purchased

LoggingOut

purchase: StockPurchase

---

stm StockPurchase

quantity entered → **InventoryCheck**
do/check availability of stock

[available] → **Sell**
do/contact trader and sell item

[!available]

Validation
do/validates details

fail

details entered
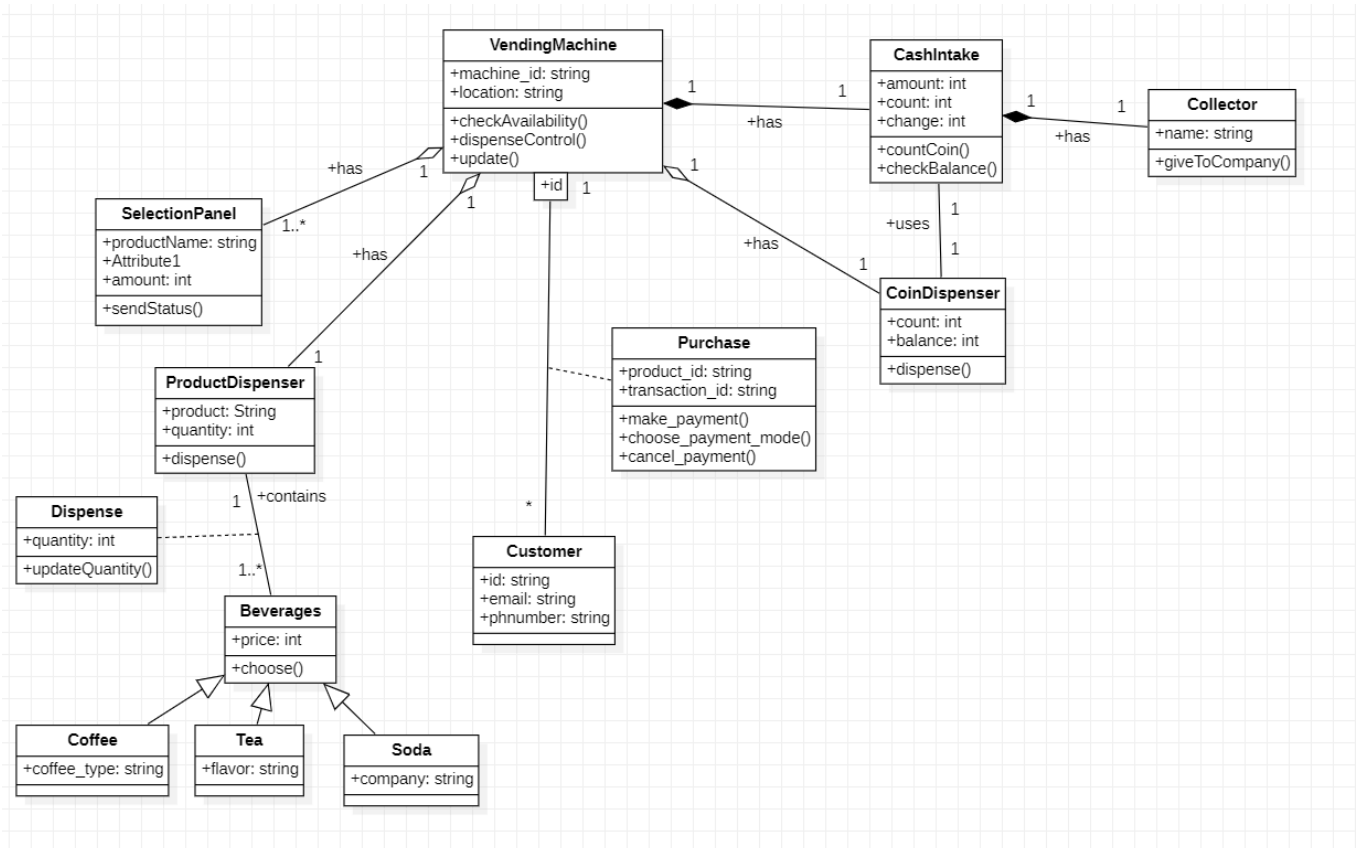
Payment
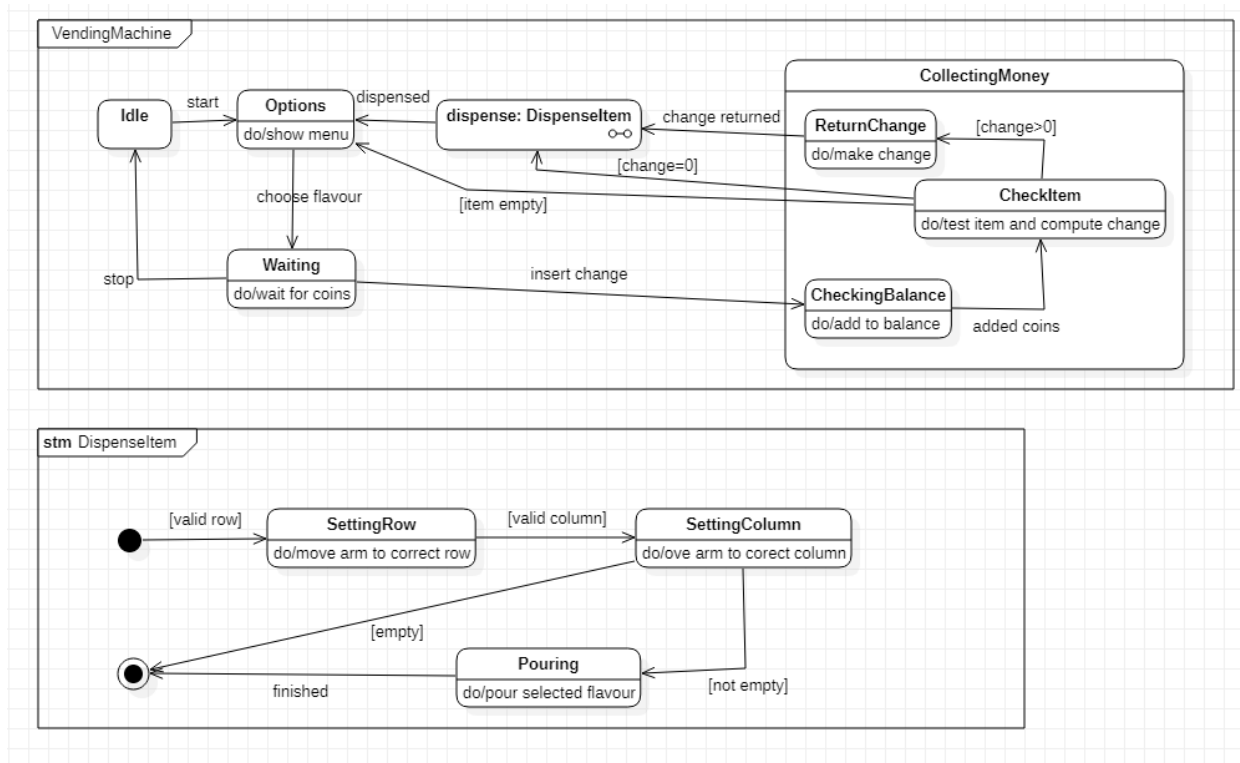do/enter back details

successful

4. Coffee Vending Machine

4.1 Problem Statement

The coffee vending machine is basically for the customers to buy coffee by themselves without any third person being involved. A coffee vending machine sells different types of coffee such. Each type of coffee has a price and a name. A customer can buy their choice of coffee by selecting the button of their coffee and paying for the same through the coin box.

4.2 Class Diagram

4.3 State Diagram

VendingMachine

Idle → start → **Options** do/show menu → dispensed → **dispense: DispenseItem** ○-○

CollectingMoney

← change returned ← **ReturnChange** do/make change ← [change>0]

choose flavour

[change=0]

[item empty]

CheckItem do/test item and compute change

stop → **Waiting** do/wait for coins → insert change → **CheckingBalance** do/add to balance ← added coins

stm DispenseItem

● → [valid row] → **SettingRow** do/move arm to correct row → [valid column] → **SettingColumn** do/ove arm to corect column

[empty]

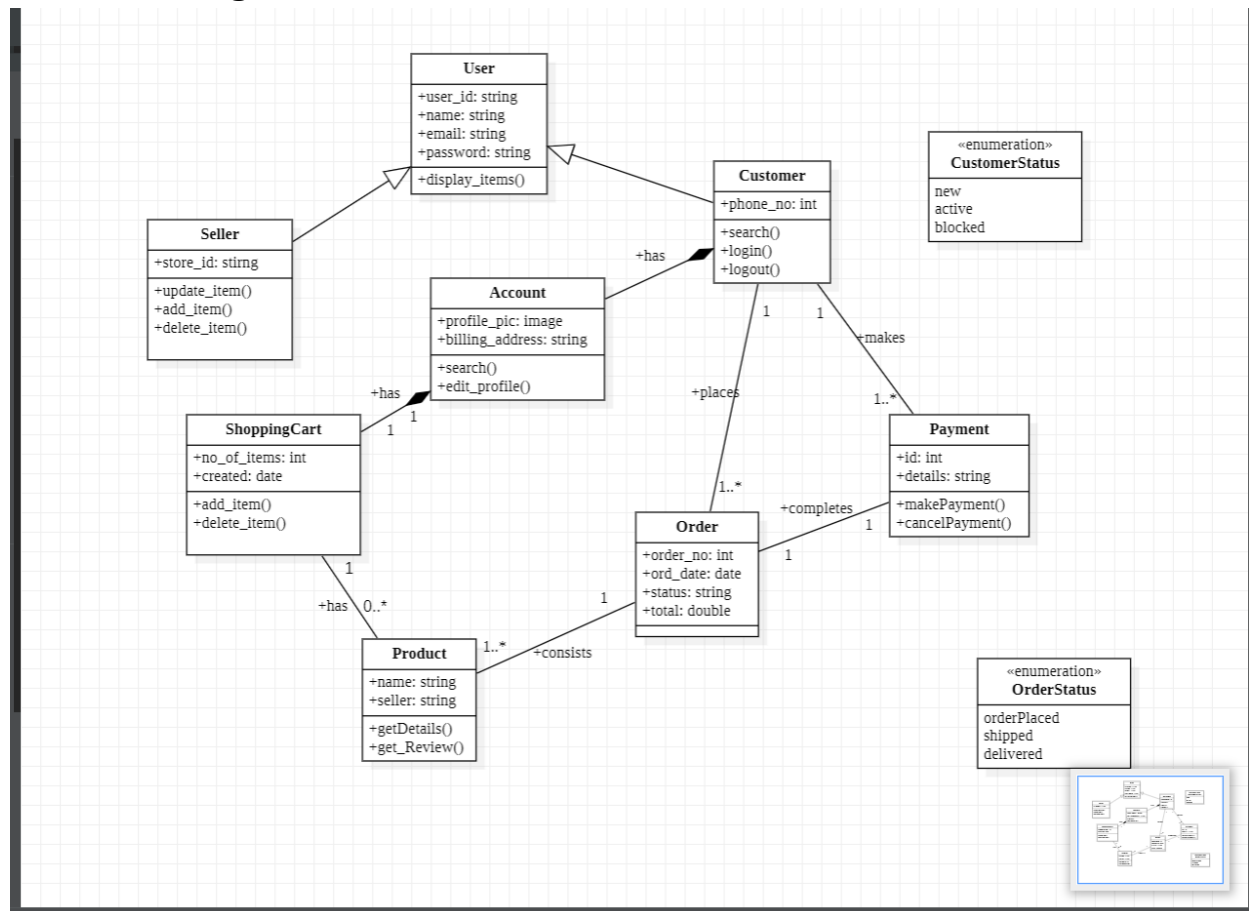◉ ← finished ← **Pouring** do/pour selected flavour ← [not empty]
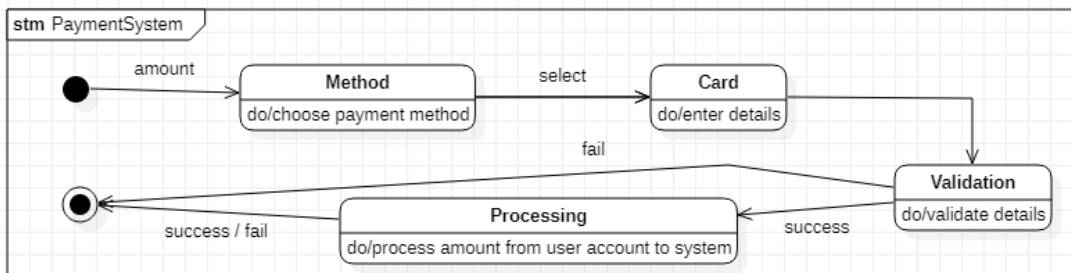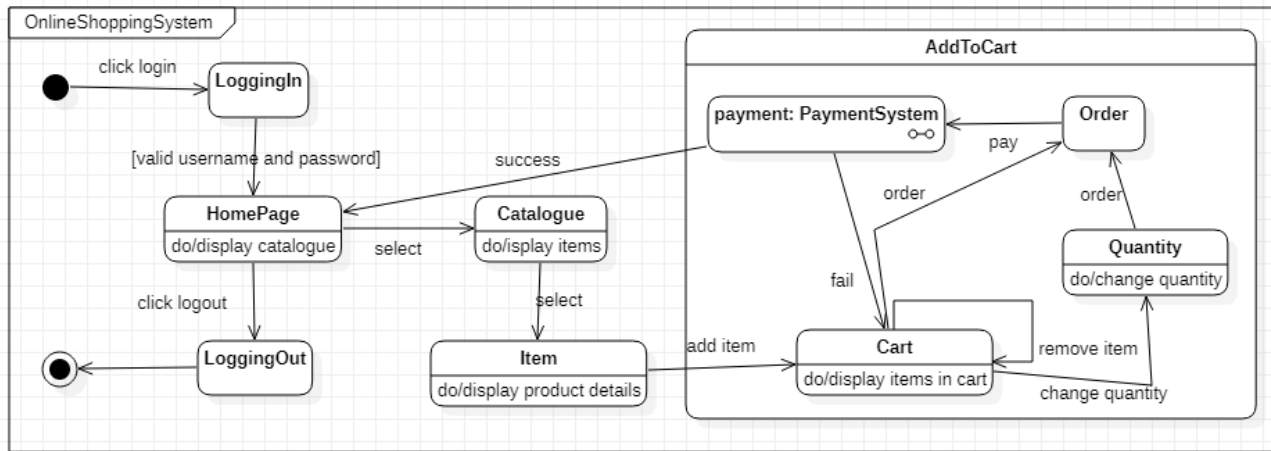
11

5. Online Shopping System

5.1 Problem Statement

The Online Shopping System for all kind of products web application is intended to provide complete solutions for vendors as well as customers through a single get way using the internet. It will enable vendors to setup online shops, customer to browse through the shop and purchase them online without having to visit the shop physically. The administration module will enable a system administrator to approve and reject requests for new shops and maintain various lists of shop category. This system allows the customer's to maintain their cart for add or remove the product over the internet.

5.2 Class Diagram

5.3   State Diagram

OnlineShoppingSystem

click login → **LoggingIn**

[valid username and password]

HomePage
do/display catalogue

select → **Catalogue**
do/isplay items

success

click logout → **LoggingOut**

select → **Item**
do/display product details

add item → **Cart**
do/display items in cart

remove item / change quantity

AddToCart

payment: PaymentSystem ○—○

order / fail

pay → **Order**

order → **Quantity**
do/change quantity

stm PaymentSystem

amount → **Method**
do/choose payment method

select → **Card**
do/enter details

→ **Validation**
do/validate details

fail

success → **Processing**
do/process amount from user account to system
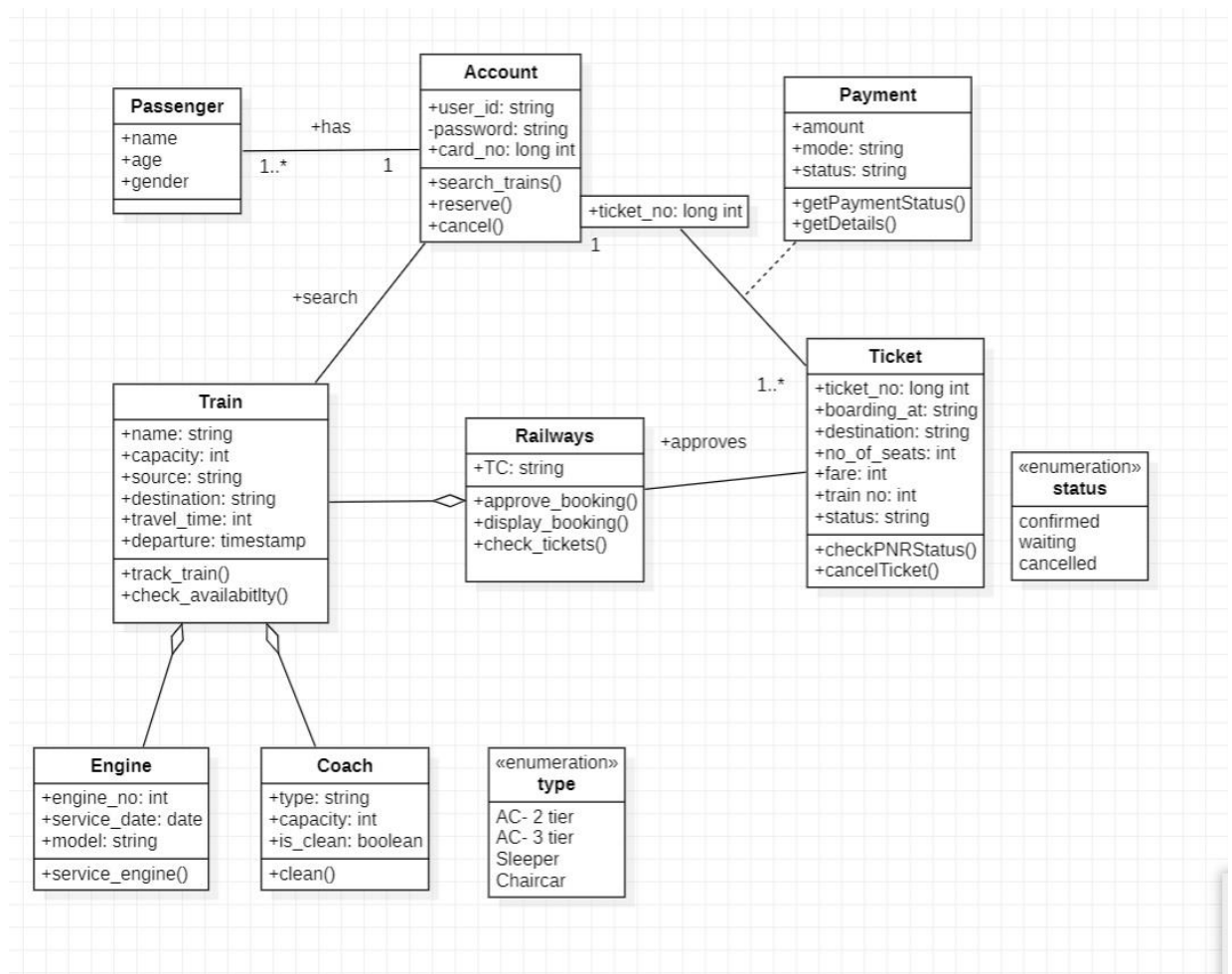
success / fail

13

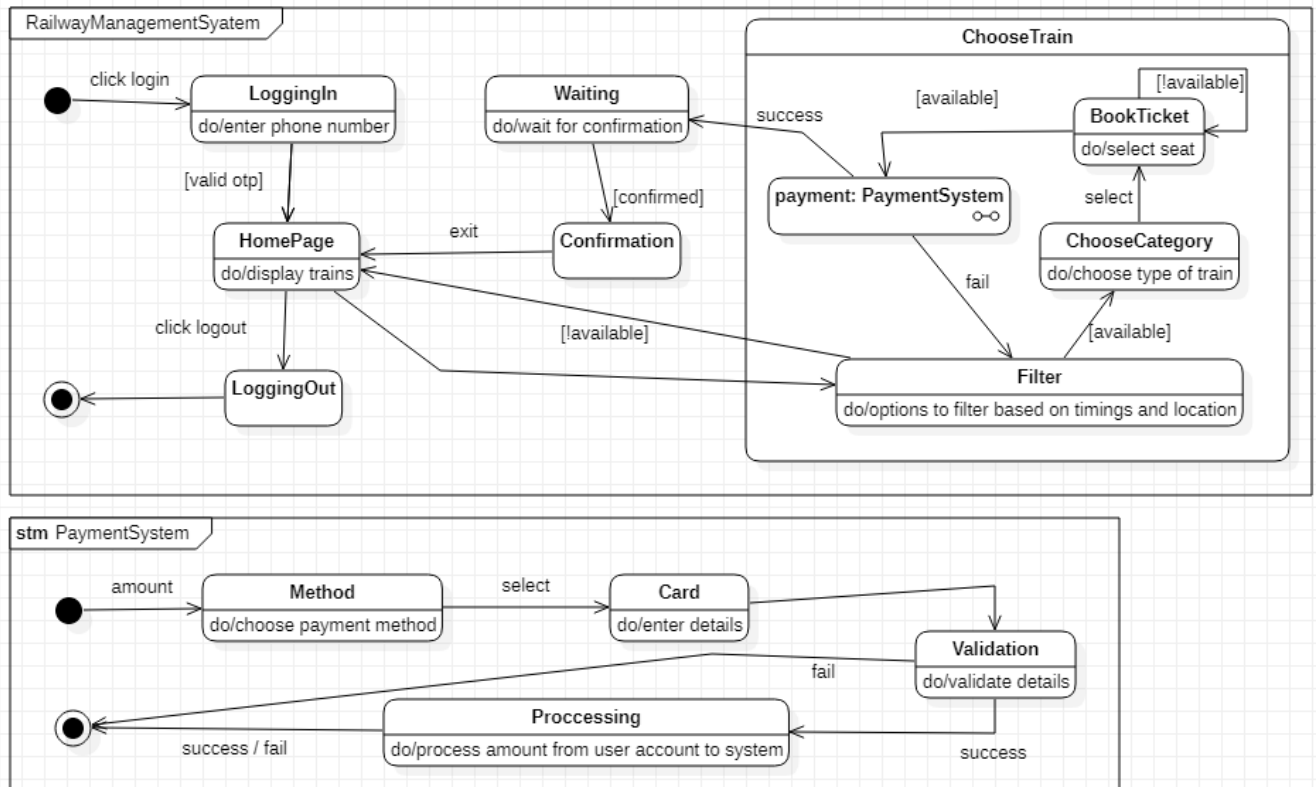## 6. Railway Reservation System
6.1 Problem Statement

Railway Reservation System is a system used for booking tickets over internet. Any Customer Can book tickets for different trains. Software has to be developed for automating the manual reservation system of railway. The system should be standalone in nature. It should be designed to provide functionalists like booking of tickets in which a user should be able to applied for tickets of any train and of any class. The software takes the current system date and time as the date of issue and calculates the amount to be paid by the user. It also provides the functionality of cancellation of tickets.

6.2 Class Diagram

6.3   State Diagram

RailwayManagementSyatem

click login → **LoggingIn** do/enter phone number

Waiting do/wait for confirmation

ChooseTrain

[valid otp]

[available] → **BookTicket** do/select seat [!available]

success

HomePage do/display trains

exit

Confirmation

[confirmed]

payment: PaymentSystem

select

ChooseCategory do/choose type of train

click logout

[!available]

fail

LoggingOut

Filter do/options to filter based on timings and location

[available]

---

stm PaymentSystem

amount → **Method** do/choose payment method

select → **Card** do/enter details

Validation do/validate details

fail

Proccessing do/process amount from user account to system
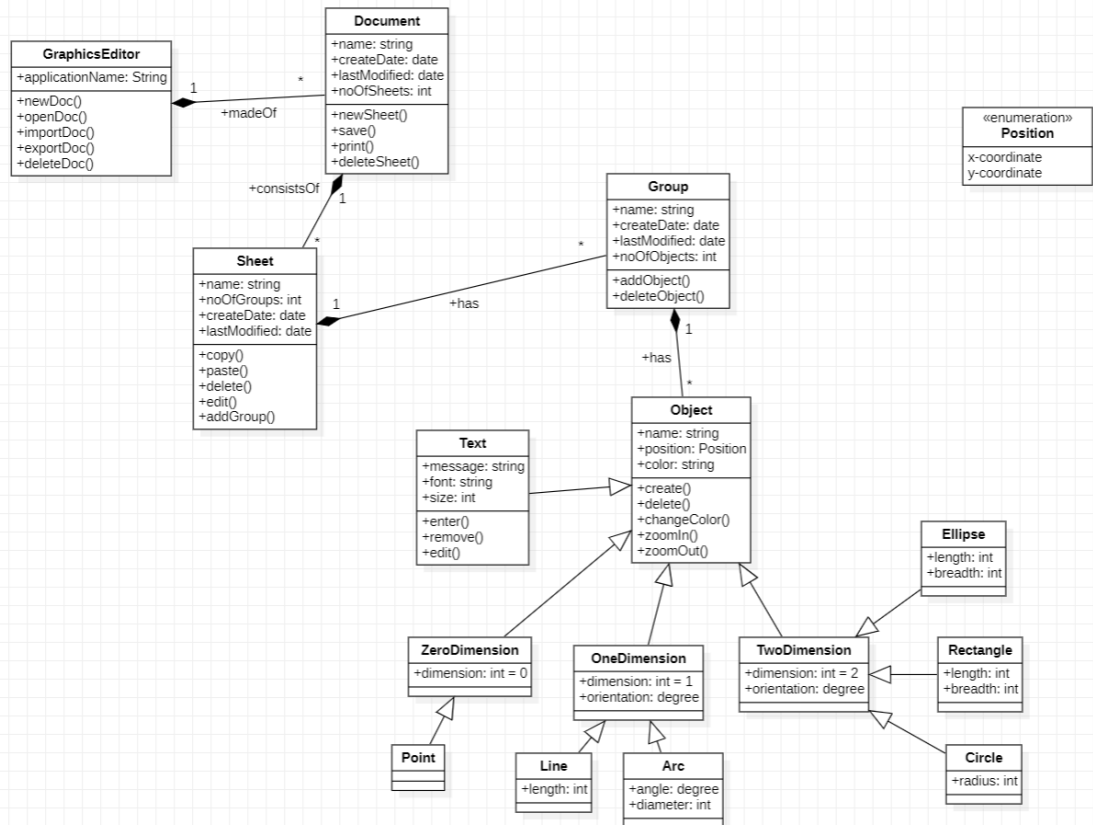
success / fail

success

7. Graphics Editor

7.1 Problem Statement

The graphics editor provides an Application Programmer's Interface that enables a programmer to develop their own graphical model editor for a specific type of model. This API in turn, provides an environment in which the editor functions, and the programmer can create a graphical editor and palette of shapes in order to modify an underlying model. The graphical editor provides an interface with which the programmer implements said editor for a given underlying model. Such instance of the graphical editor allows a user to drag objects from a specified model into a working graphical diagram.

7.2 Class Diagram

7.3 State Diagram