

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



Lab Record

Object Oriented Analysis and Design

Submitted in partial fulfillment for the 6th Semester Laboratory

Bachelor of Technology
in
Computer Science and Engineering

Submitted by:

SHEHYAAZ KHAN NAYAZI

1BM17CS094

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
Jan-May 2019

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



CERTIFICATE

This is to certify that the Object-Oriented Analysis and Design(16CS6DCOOM) laboratory has been carried out by **SHEHYAAZ KHAN NAYAZI(1BM17CS094)** during the 6th Semester Jan-May-2019.

Signature of the Faculty Incharge:

NAME OF THE FACULTY: Rekha G S

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

Table of Contents

1.College Information System
2.Hostel Management System
3.Stock Maintenance System
4.Coffee Vending Machine
5.Online Shopping System
6.Railway reservation system
7.Graphics Editor

1. College Information System

Problem Statement

Design a computerized College Information System which automates the admission process in the college, makes report generation easier and simplifies the storage and retrieval of information.

Software Requirement Specification

Purpose

The purpose of the College Information System is to automate the existing manual process of admission, make report generation easier and simplify the storage and retrieval of information.

Scope

This application will be a web-based software that will automate the various activities that take place in managing a college. This system promises to make the process as convenient and easy as possible. The system provides much more efficiency than the existing process.

Requirements

1. Functional Requirements

- a) Entry of new student to the department.
- b) Entry of new staff to the department.
- c) Entry of attendance information.
- d) Entry of examination details.
- e) Provide individual and class-wise report

2. Non-functional Requirements

- a) The database storing the details must be backed-up and secure.
- b) The system should be easy to handle.
- c) The response time should be very small.
- d) The system should be developed within the specified budget.

3. User Requirements

- a) The student details should be displayed,.
- b) The attendance information must be displayed.
- c) Course, Department and Exam information must be displayed.

Class Diagram

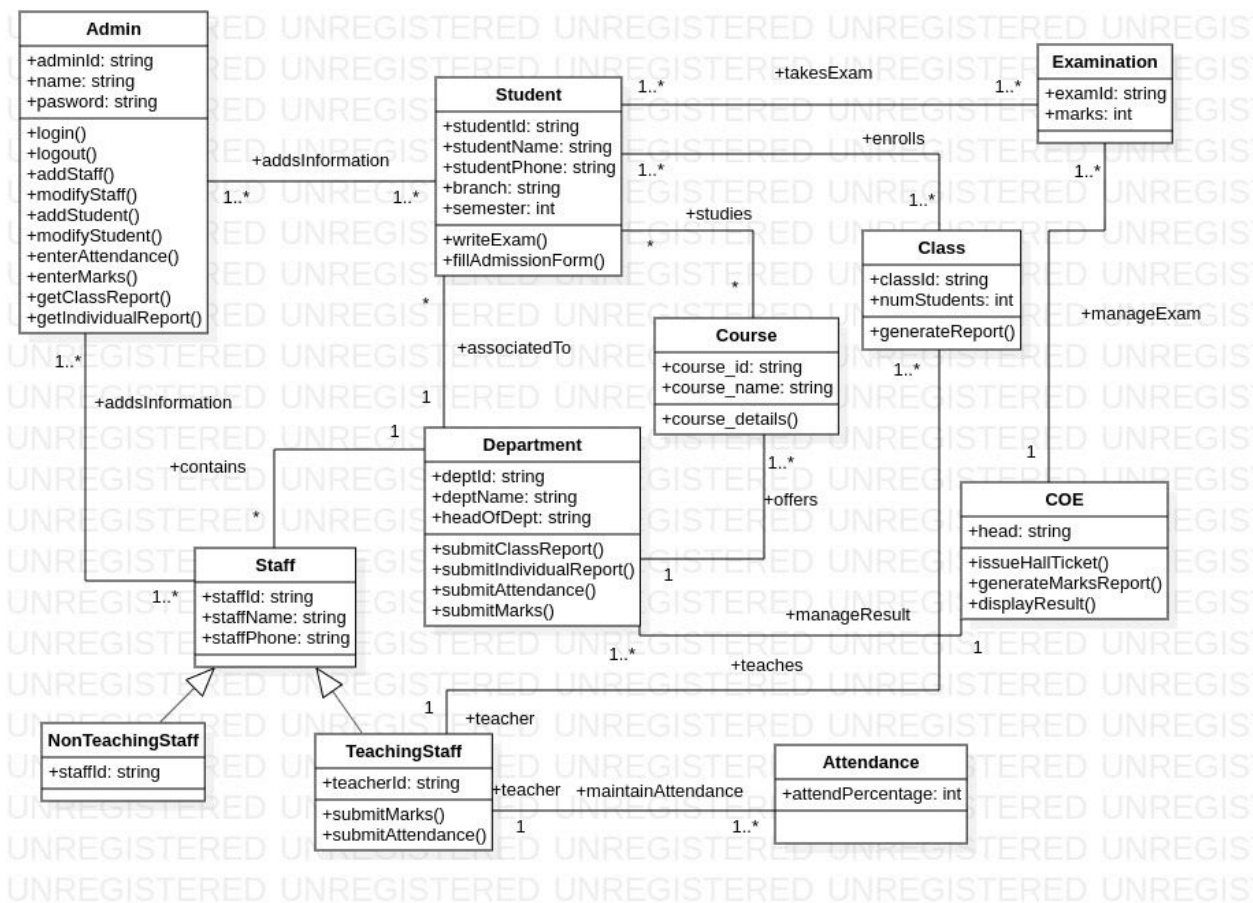


Figure 1: Class Model

Admin class: This class models the admin. It provides various functionalities to be performed by the admin such as adding student and staff information, etc.

Department class: This class models a department in the college.

COE class: This handles all the exam related information from declaring results to calculation marks, GPA.

Staff class: This is a generalized class which contains the most common details of staff members. This class is inherited by two more classes namely teaching and non-teaching staff class which are more specific in their functionalities.

Student class: This class models a student. It has operations to write exam, etc.

Attendance class: This class models the attendance information.

Course class: This class models a course and provides details such as course name and course id, along with some operations.

State Diagram

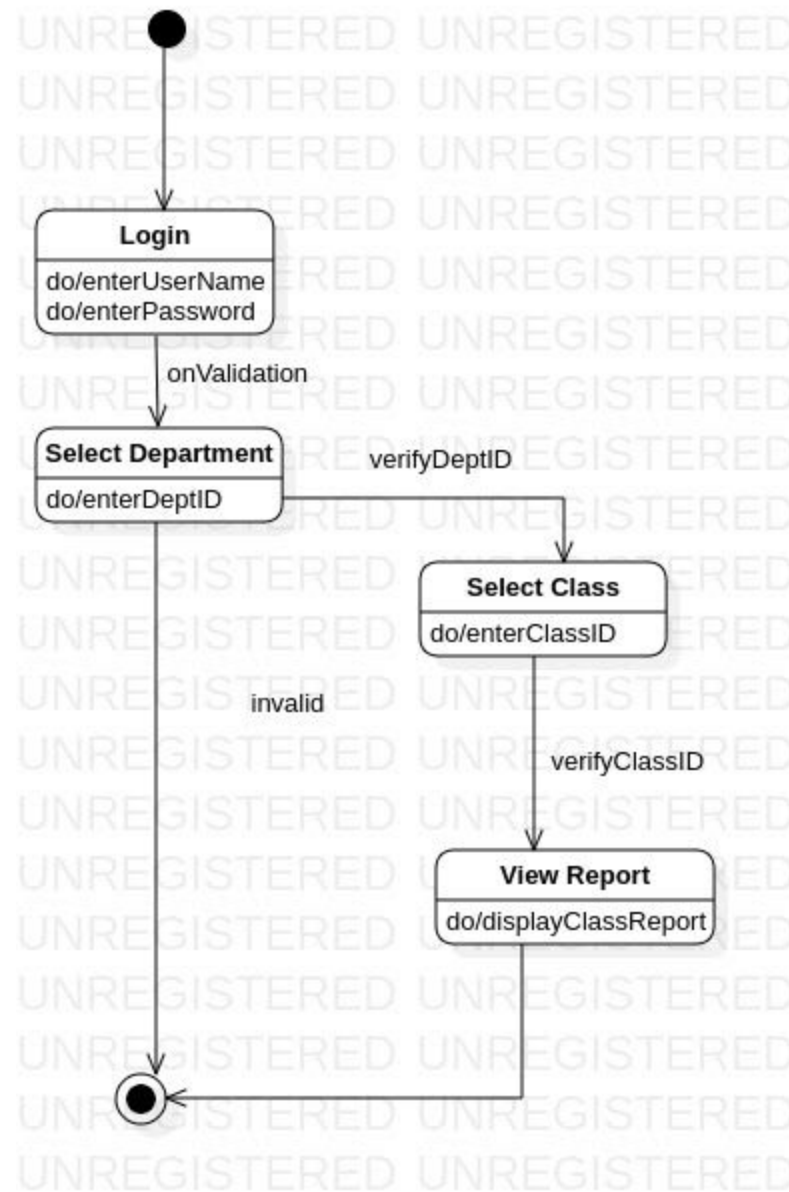


Figure 2 : State Model

This state diagram shows how a user can obtain a class report. The user first logs into the system, enters the department id of the particular class, followed by entering the class id and finally views the class report.

Advanced State Diagram

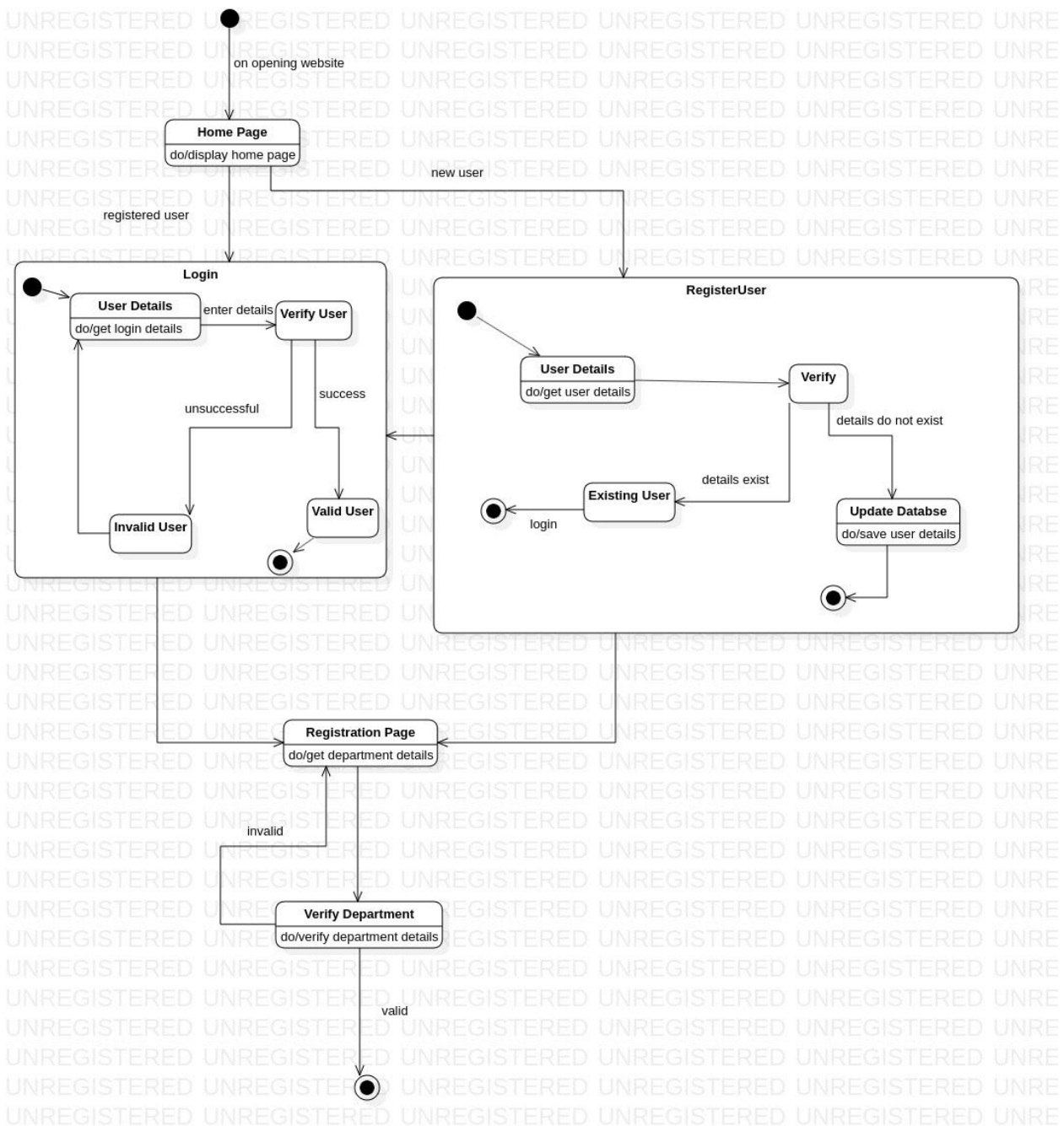


Figure 3 : Advanced State Model

This diagram illustrates the steps involved in registering a department. It also shows the steps taken by a new as well as existing user to login, Once successfully logged in, the department details are entered and verified.

2. Hostel Management System

Problem Statement

To design an online Hostel Management System which will provide an interface to help the students with the process of being allocated rooms, a hostel being assigned to a warden and also to maintain all the related information and details regarding the hostel. This system also aims to reduce error and data redundancy.

Software Requirement Specification

Purpose

The purpose of this document is to describe the requirements to develop a Hostel Management System. This is a description of the features required to address the process of hostel management and add the necessary framework to explain the functionality. It is also a description of requirements and other factors needed to provide a complete and convenient system.

Scope

This system will be a web-based system that will cater to the needs of the hostel management process. This system promises to make the process as convenient as possible. The benefit of this system is that it looks forward to providing better efficiency than the existing systems.

Requirements

1. Functional Requirements

- a) It should contain an interface for hostel details.
- b) It should contain an interface for student details.
- c) It should contain an interface for warden and other staff details.
- d) The student must be able to choose the facilities needed.

2. Non-functional Requirements

- a) The database storing the details must be backed-up and secure.
- b) The system should be easy to handle.
- c) The system should be able to handle multiple users.

3. User Requirements

- a) The hostel details must be displayed.
- b) The various facilities provided must be displayed.

Class Diagram

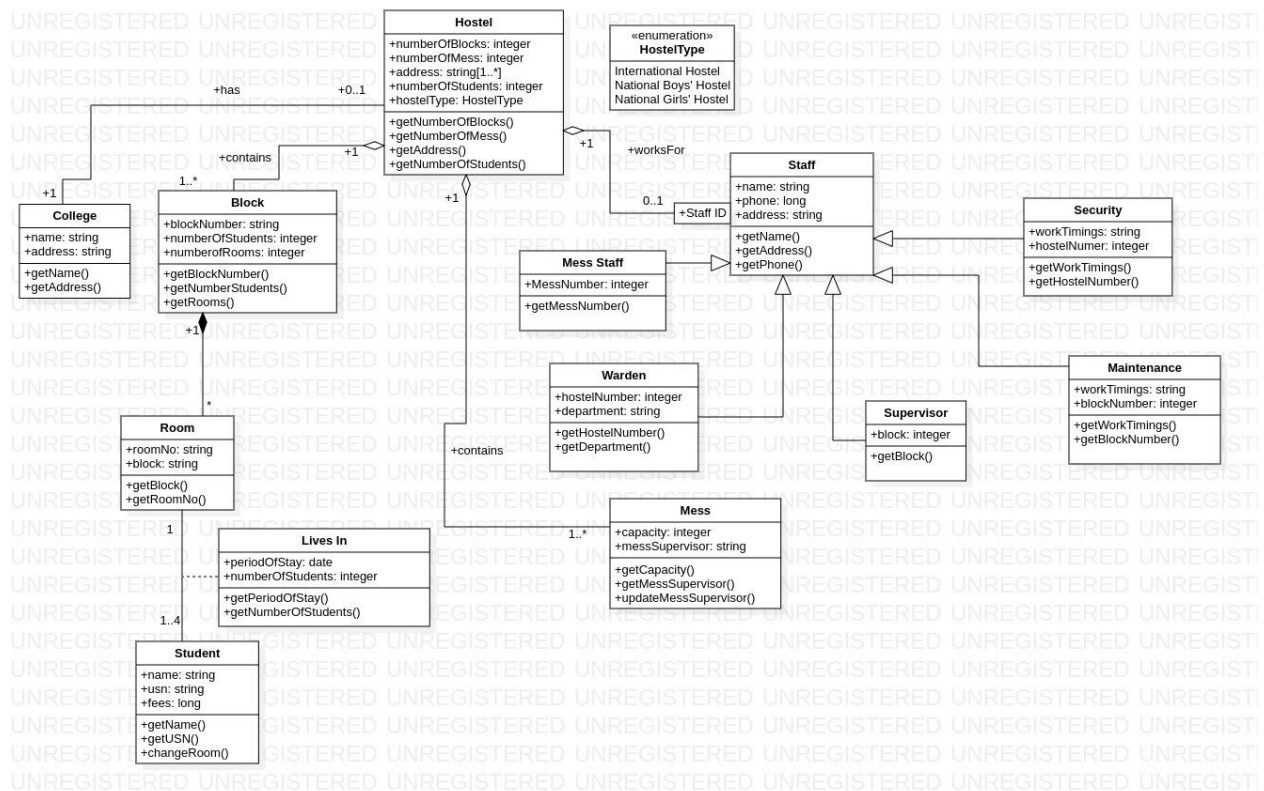


Figure 1 : Class Model

Hostel Class : This class models a hostel and contains hostel details.

College Class : This class models a college.

Block Class : This class models a particular hostel block.

Room Class : This class models a room in a hostel block.

Student Class : This class models a student.

Lives in Class : An association class between Room and Student.

Staff Class : This is a general class which contains details common to all staff.

Warden Class : This is inherited from Staff and models a hostel warden.

Mess Class : This class models the hostel mess.

State Diagram

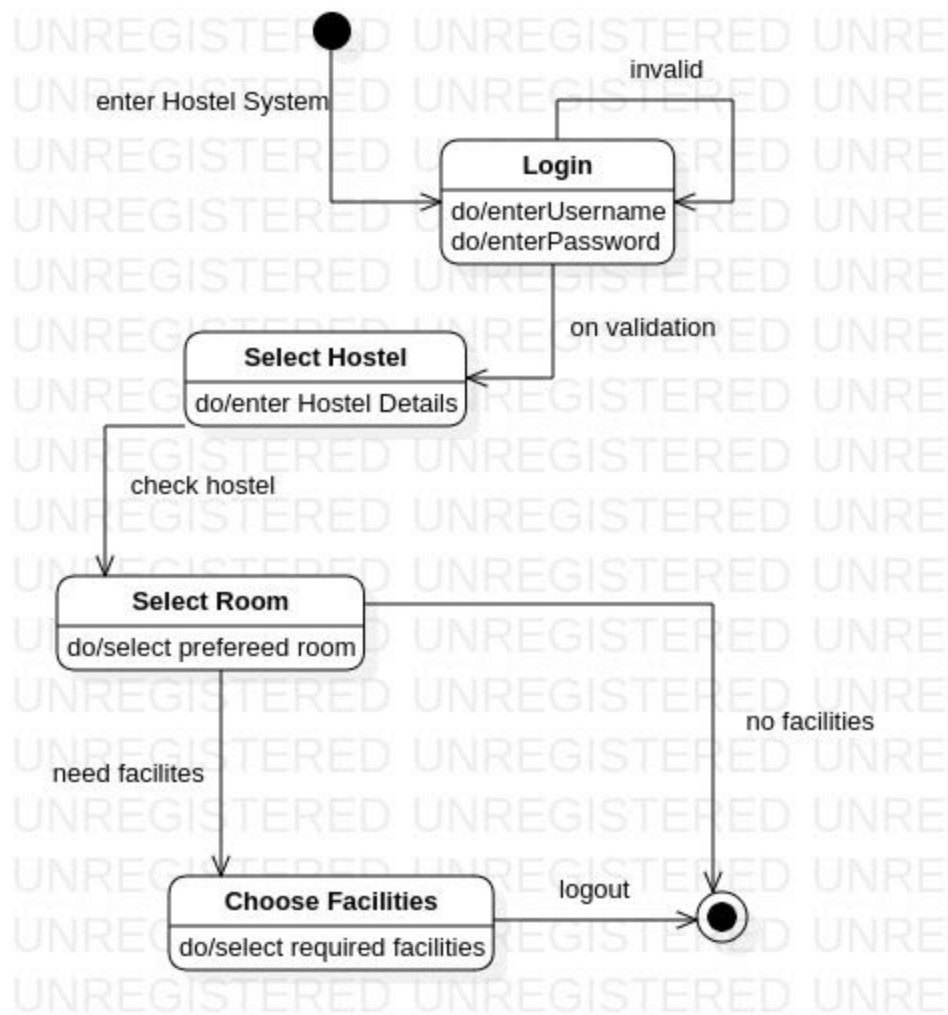


Figure 2 : State Model

This model shows the steps to be taken for a student to select a room in a hostel block. The student logs into the system, selects a particular hostel, selects a preferred room, specifies any facilities required, and then logs out of the system.

Advanced State Diagram

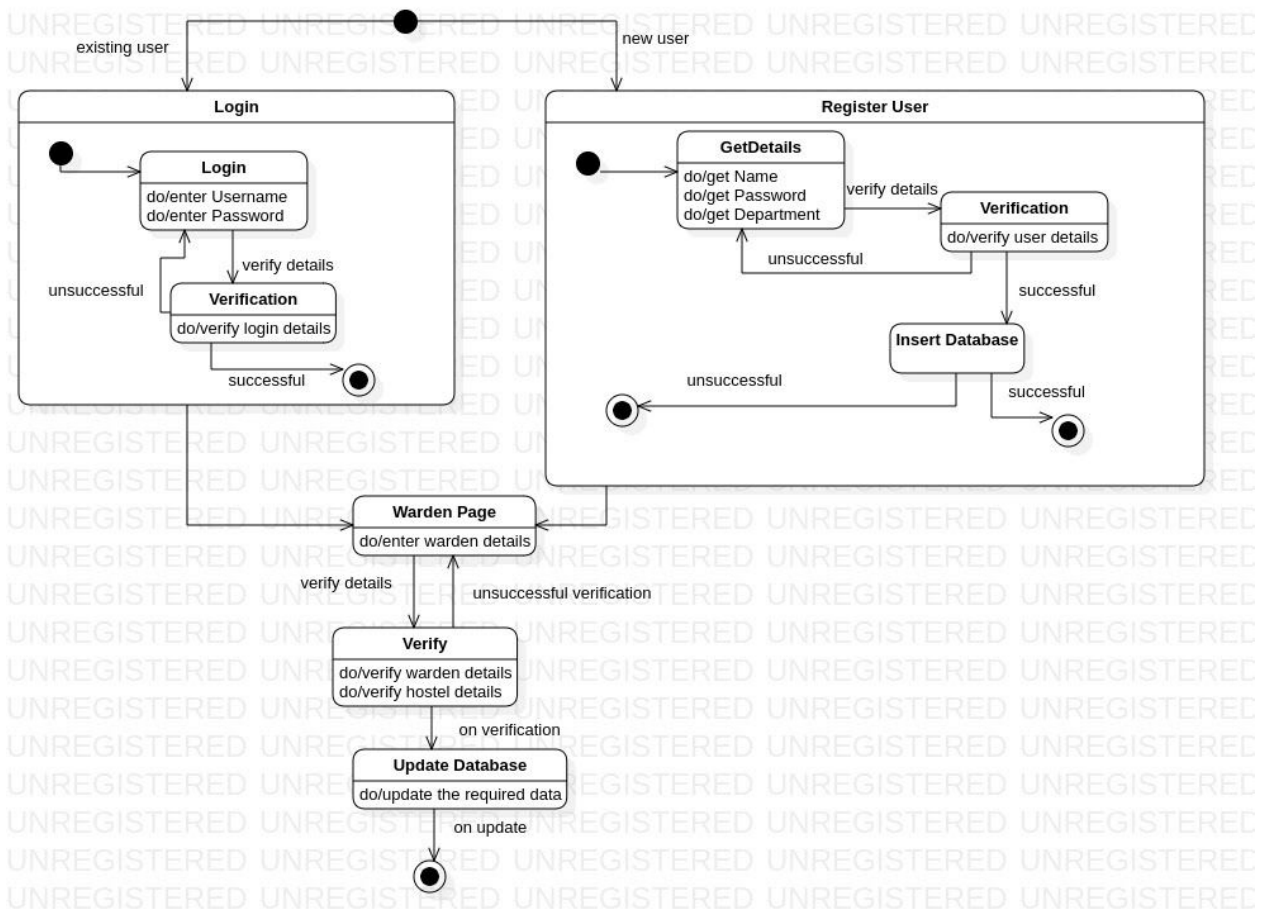


Figure 3 : Advanced State Model

This model shows the steps involved in updating a hostel warden. It shows the steps for existing and new users to login to the system. On successful login, the warden details are entered, verified and the database is updated.

3. Stock Maintenance System

Problem Statement

To design a simple stock maintenance system that minimizes the interaction between the user and a physical ledger to maintain stocks. This system aims at moving the entire system online for more robust and easier use of the system.

Software Requirement Specification

Purpose

The purpose of this document is to describe the requirements to develop the Stock Maintenance System. It contains description of the features required to address the process of stock maintenance. It is also a description of other requirements and factors needed to provide a convenient system.

Scope

This system will be a web-based system that will cater to the needs of a stock maintenance process. This system promises to make the process as convenient as possible. The benefit of this system is that it looks forward to providing better efficiency than most other similar systems.

Requirements

1. Functional Requirements

- a) It should be able to track products.
- b) It should be able to update stock details.
- c) It should be able to generate stock reports.

2. Non-functional Requirements

- a) The database storing the details must be backed-up and secure.
- b) The system should be easy to handle.
- c) The system should be very fast.

3. User Requirements

- a) Stock details must be displayed.
- b) Product details must be displayed.
- c) Vendor details must be displayed.

Class Diagram

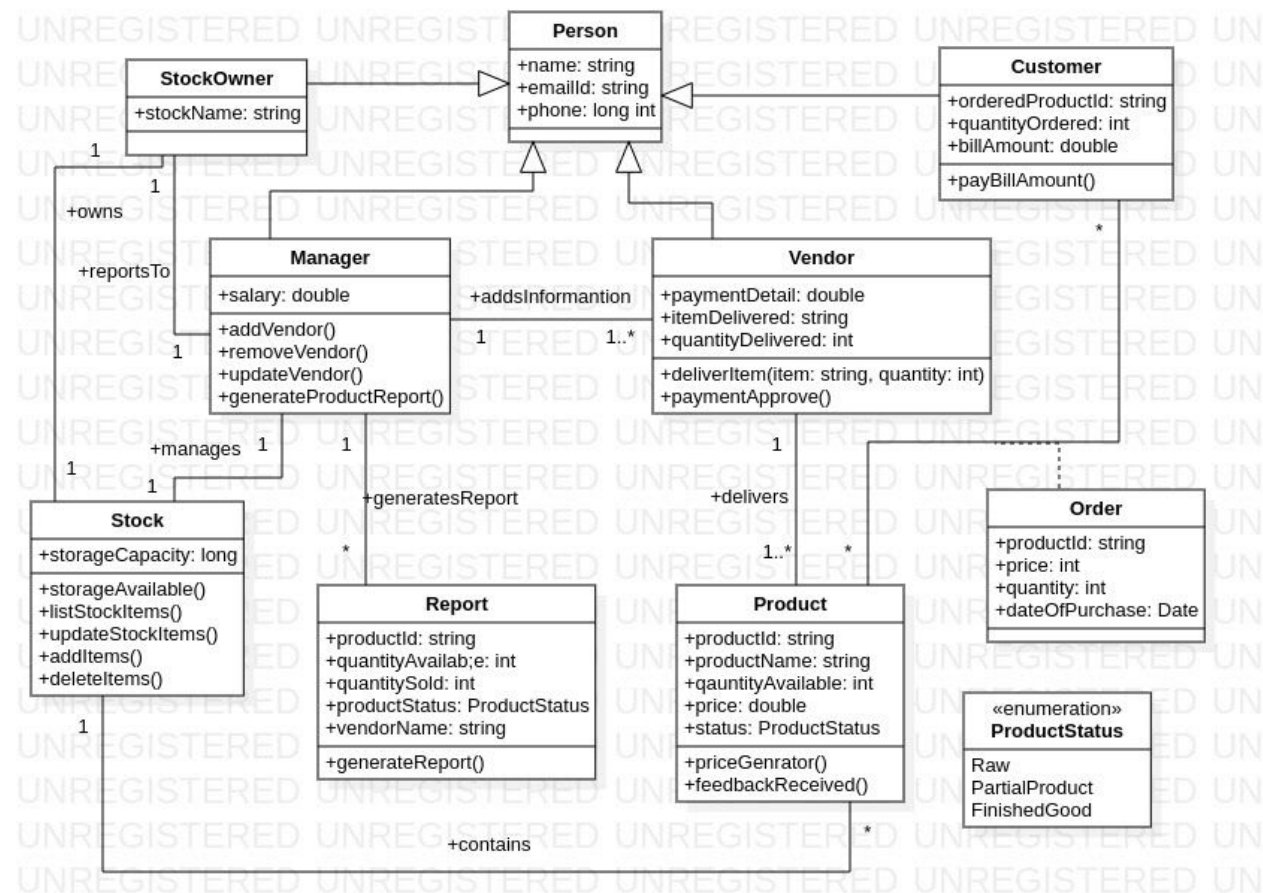


Figure 1 : Class Model

Person Class: This generalized class contains the details of the person.

Manager Class: This class models a manager and has operations such as adding, updating and removing vendors.

Vendor Class: This class models a vendor.

Product Class: This class models a product and contains product details.

Report Class: This class contains the report details such as the number of items available, vendor names, quantity sold, the status of the products which is generated by the manager. It has operations to generate a report.

Customer Class: This class contains the customer details like the order placed, quantity ordered, etc.

Stock Class: This class models a stock.

Order Class: This is an association class between Customer and Product classes.

State Diagram

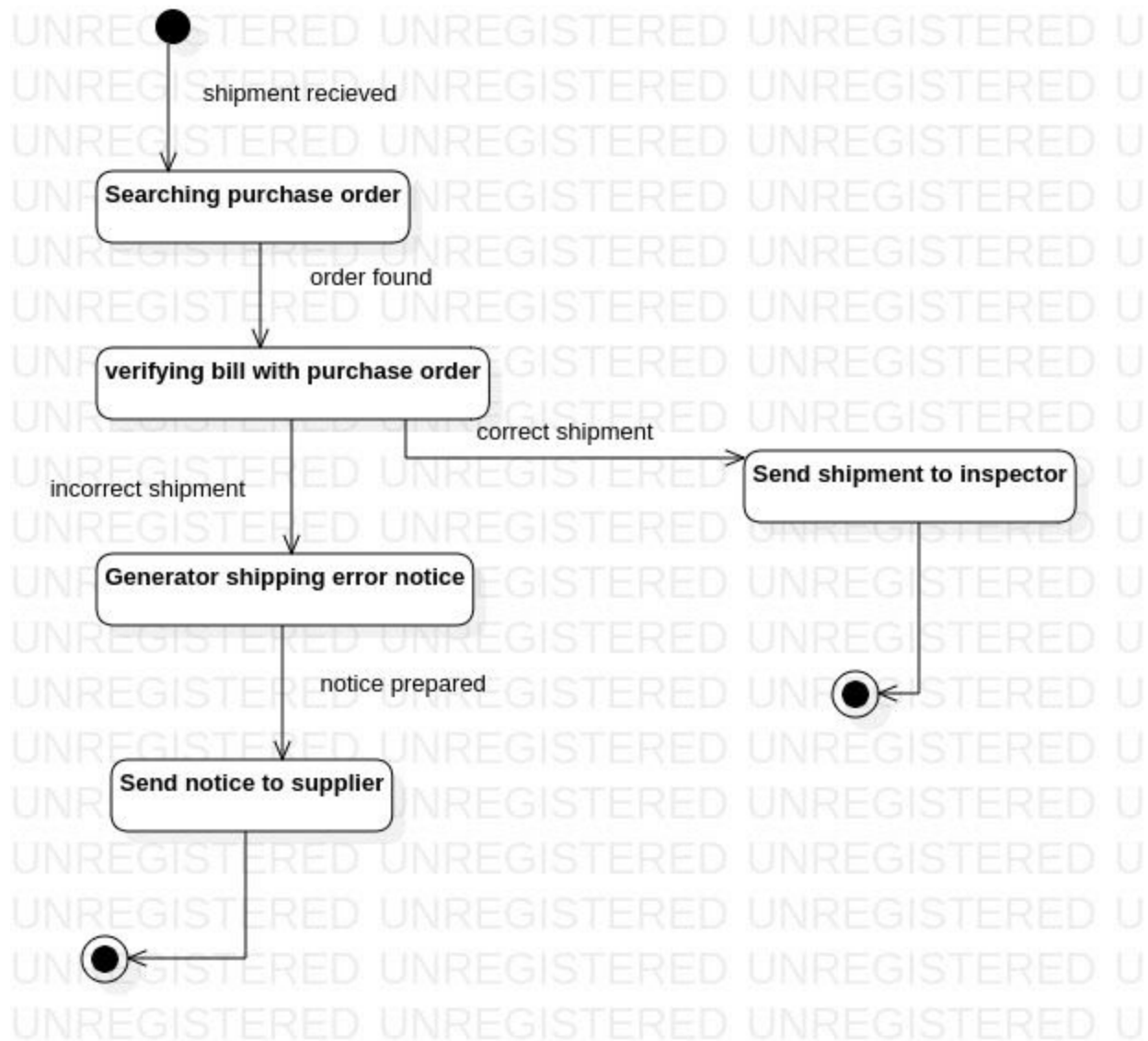


Figure 2 : State Model

This state diagram denotes the steps when a shipment is received and is delivered. When a shipment comes into stock, its associated order is searched. The order details are then verified and confirmed. On successful verification, the order is sent to the inspector. If the verification is unsuccessful, an error notice is generated and is sent to the supplier.

Advanced State Diagram

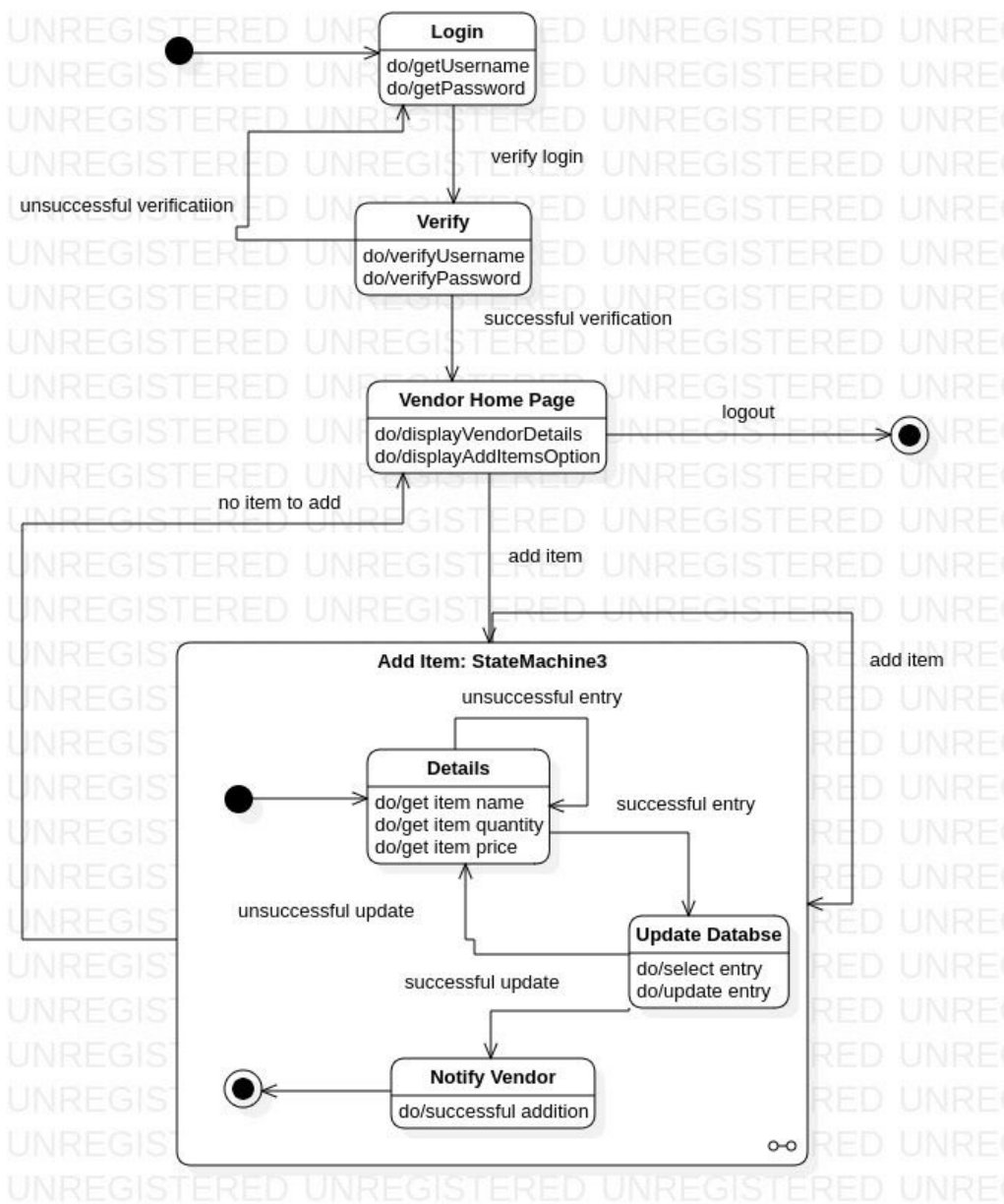


Figure 3 : Advanced State Model

This state diagram depicts the registration of a vendor into the system. The vendor is added to the system by the manager. On logging into the system, the manager is redirected to the vendor homepage where vendor details are displayed. These can be updated. New items can be added which is shown by the add item state machine. The database is updated each time an item is added.

4. Coffee Vending Machine

Problem Statement

To design a simple coffee vending machine with simple and straightforward features that minimize the interaction between the person and the machine and simplifies the coffee vending process to consume less time of the user.

Software Requirement Specification

Purpose

The purpose of this document is to describe the requirements to develop a coffee vending machine. It contains a description of the features required in the coffee vending machine and also the description of other requirements and factors needed to provide a convenient system.

Scope

This system will be a cloud-based system that will cater to the needs of a coffee vending machine. This system promises to make the process as convenient as possible. The benefit of this system is that it looks forward to providing better efficiency than most other similar systems.

Requirements

1. Functional Requirements

- a) Touch screen feature.
- b) Display a list of available items.
- c) Amount check.
- d) Ability to select the desired item.

2. Non-functional Requirements

- a) The database storing the details must be backed-up and secure.
- b) The system should always be powered.

3. User Requirements

- a) Item details should be displayed.
- b) Transaction cancellation and updation should be possible.

Class Diagram

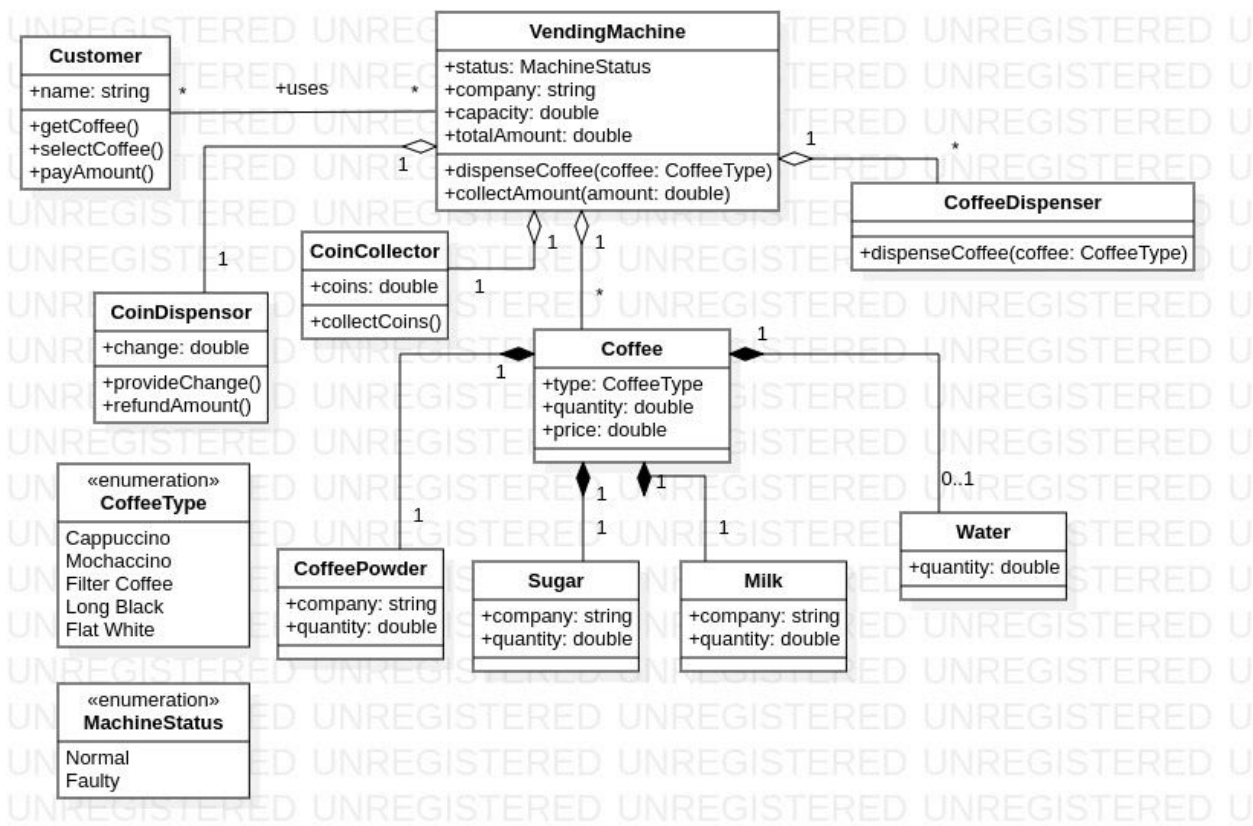


Figure 1 : Class Model

Customer Class : This class contains customer details.

VendingMachine Class : This class models the coffee vending machine. It is an aggregation of CoinDispenser, CoinCollector, Coffee and Coffee Dispenser class.

CoinDispenser Class : This class models the coin dispenser module of the vending machine.

CoinCollector Class : This class models the coin collector module of the vending machine.

Coffee Class : This class contains coffee details such as type and price. It is a composition of CoffeePowder, Sugar, Milk and Water class.

Coffee Type : An enumeration which lists the various types of coffee available in the vending machine.

State Diagram

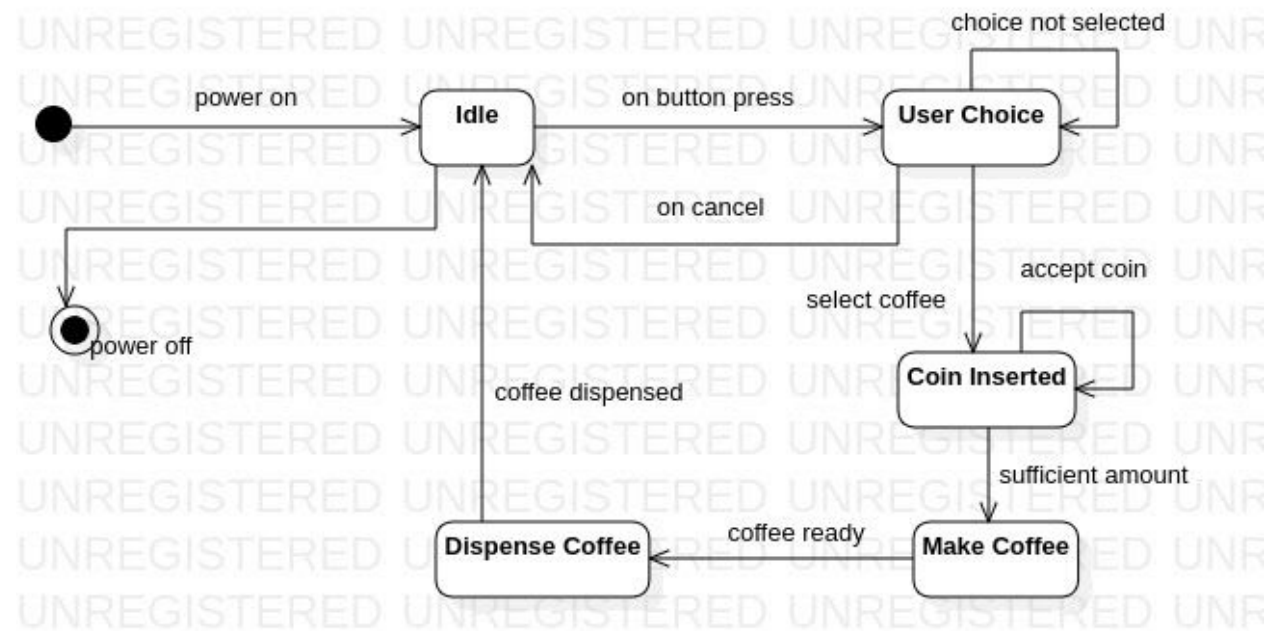


Figure 2 : State Model

The diagram describes the process of vending coffee. The user presses the Select button on the machine and chooses a coffee type. Then, the amount is inserted. If sufficient amount is inserted, the coffee is made and dispensed.

Advanced State Diagram

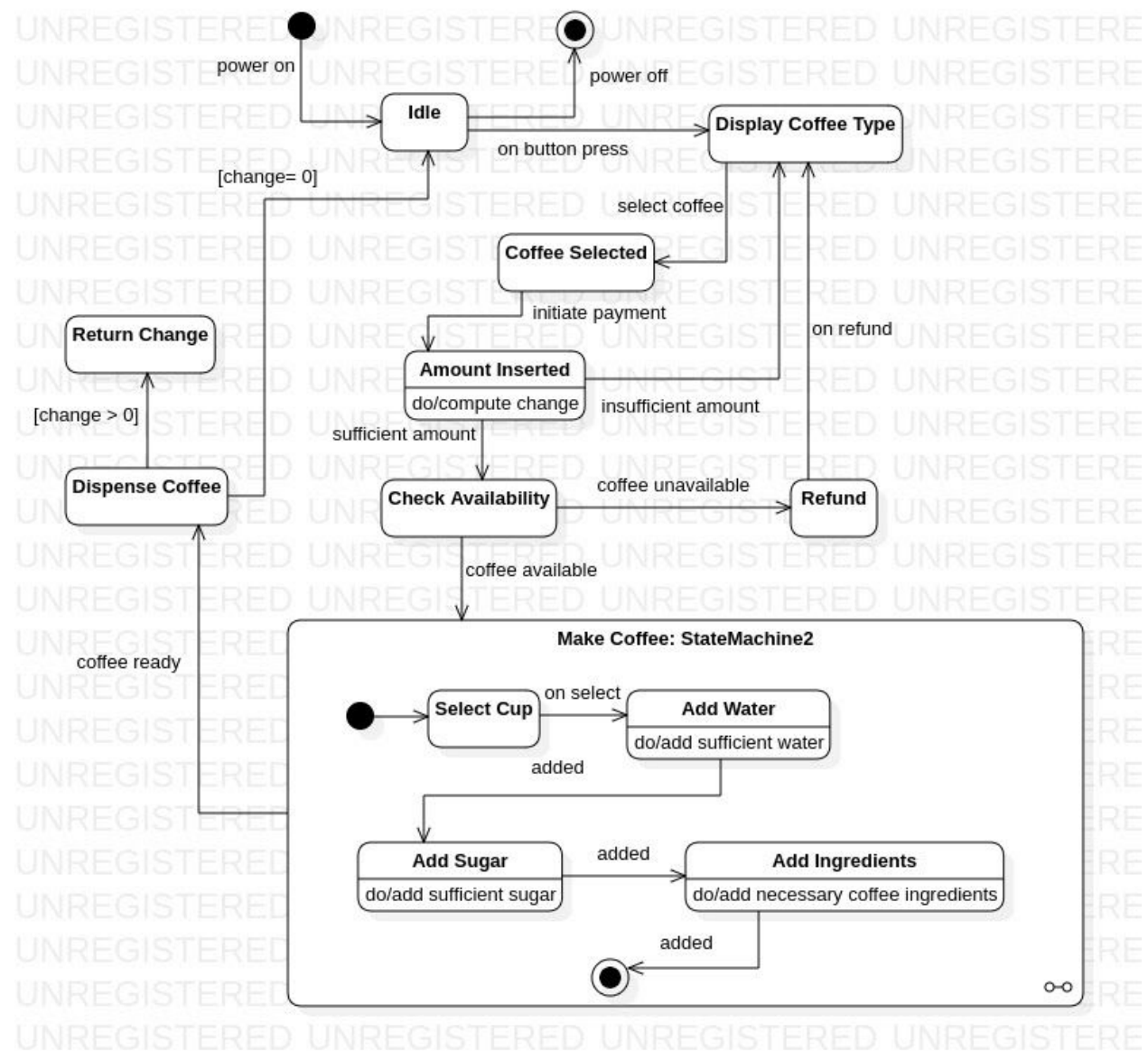


Figure 3 : Advanced State Model

The diagram describes the process of vending a coffee in detail. The user presses the select button and views the coffee types available. Then, a particular coffee type is selected and the amount is inserted. If the coffee is unavailable, the amount is refunded. Else, the selected coffee is made. To make the coffee, a cup is selected and all the ingredients are added. Once the coffee is ready, it is dispensed. If there is any change, it is returned to the user.

5. Online Shopping System

Problem Statement

To design an Online Shopping System that helps in buying products on a website application and provides a simple and interactive interface for the whole process of consumers buying the products.

Software Requirement Specification

Purpose

The purpose of this document is to describe the requirements to develop an Online Shopping System. It contains a description of the features required in the web application and also the description of other requirements and factors needed to provide a convenient system.

Scope

This system will be a web-based system that will cater to the needs of an Online Shopping System. This system promises to make the process as convenient as possible. The benefit of this system is that it looks forward to providing better efficiency than most other similar systems.

Requirements

1. Functional Requirements

- a) Provide a product search facility.
- b) Provide a personalized profile to the user.
- c) Detailed invoice generation.
- d) Provide a shopping cart facility.
- e) Allow multiple payment methods.

2. Non-functional Requirements

- a) The database storing the details must be backed-up and secure.
- b) To provide secure data transfer between server and customer system.
- c) To provide secure and encrypted storage of customer details.
- d) The system must provide very fast response time.

3. User Requirements

- a) Homepage containing products according to user interest.
- b) The screen displaying information about products and user searches.
- c) A user copy of the invoice generated.

Class Diagram

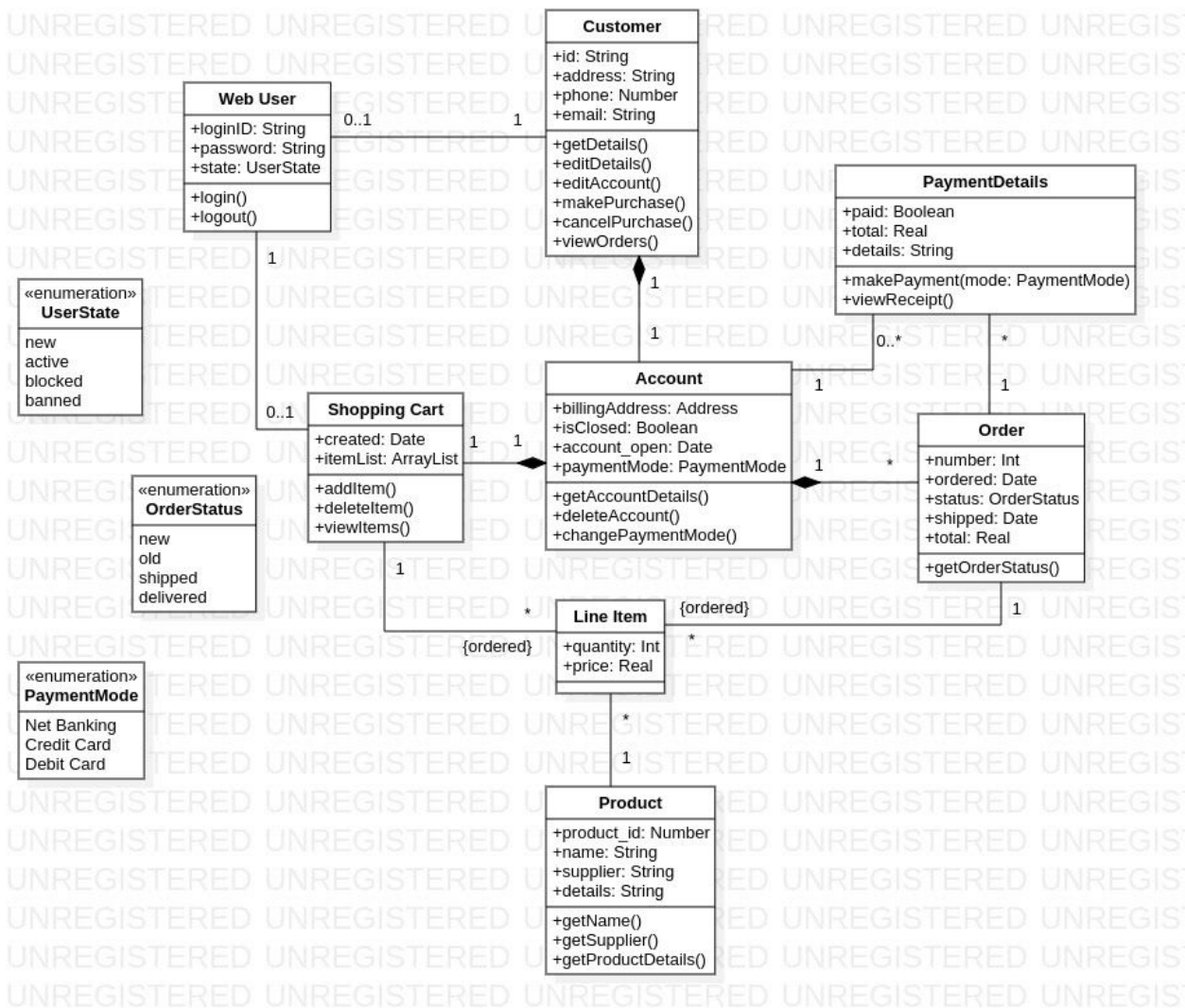


Figure 1 : Class Model

Web User Class: This class contains information of the web user.

Customer Class: This class contains all the details of the customers.

PaymentDetails Class: This class contains the details of the payment transactions.

Account Class: This class contains the details of the account such as the billing address, etc.

Shopping Cart Class: This class consists of details of the shopping cart.

Order Class: This class consists of details of each individual order.

Product Class: This class consists of the details of all the products and its supplier etc.

State Diagram

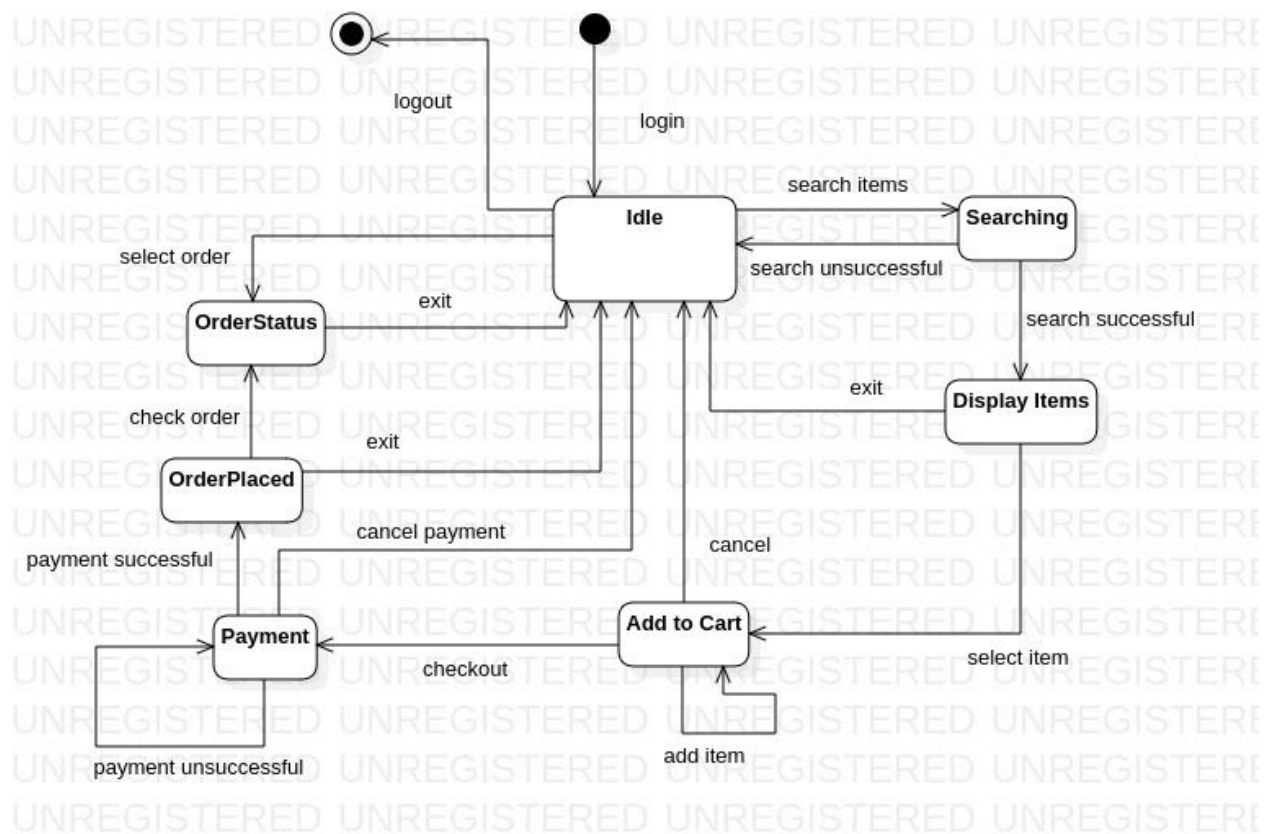


Figure 2 : State Model

The state diagram shows the steps involved in ordering of an item. On logging in, the system is initially in Idle state. The user can then either search for items or view the status of existing orders. On searching for items, the items are displayed and the user can add the item to his cart. The user can add more items to the cart. Following this, he checks out and pays for the items in the cart. On successful payment, the order is placed and the order status can be viewed.

Advanced State Diagram

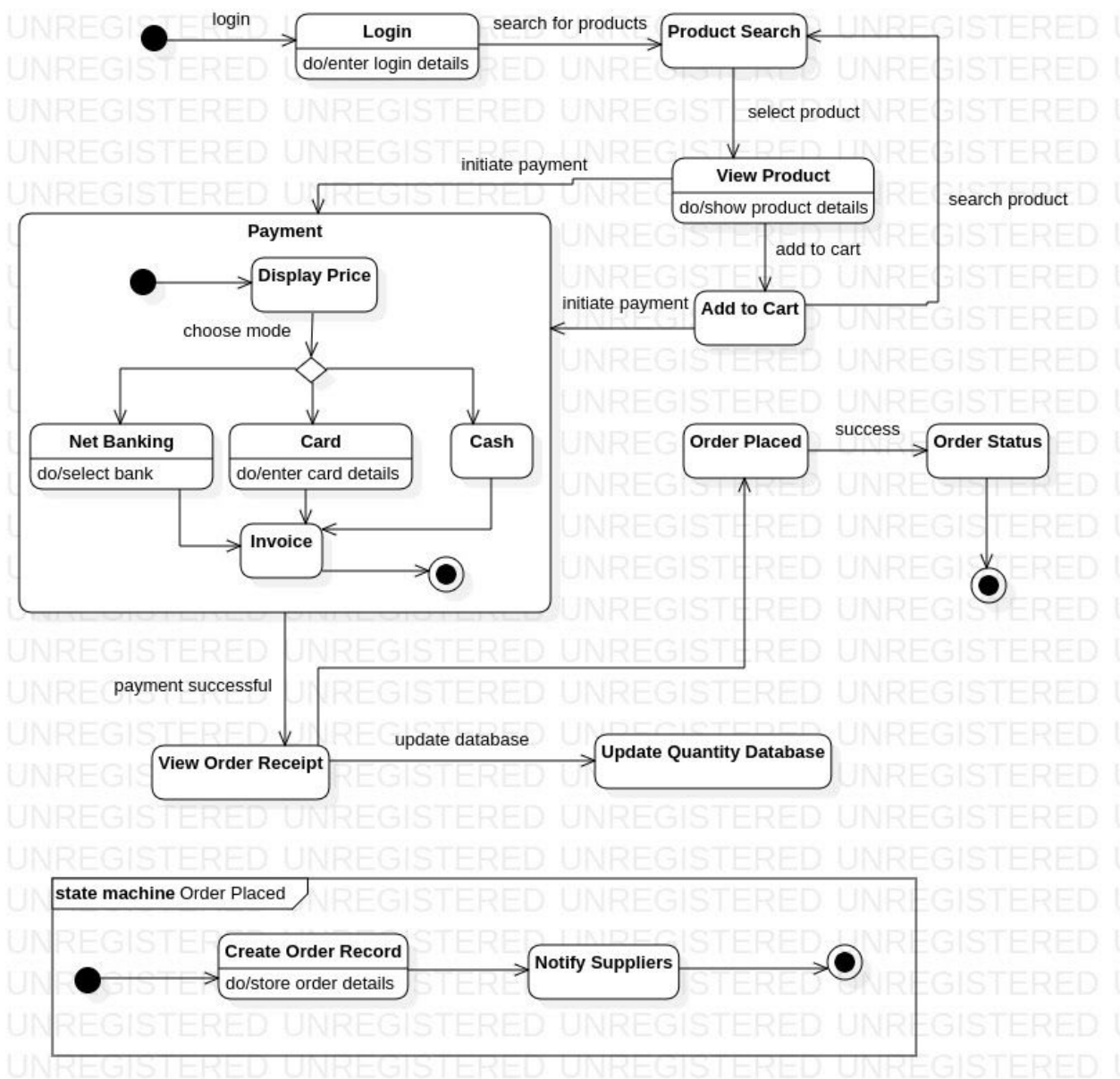


Figure 3 : Advanced State Model

The state diagram shows the steps involved in ordering of a product. Once logged in, the user can then search for products. On searching for products, the results are displayed and the user can add products to his cart. The user can search for more products and add them to the cart. Following this, he checks out and pays for the products in the cart. The user can choose net banking, card or cash as modes of payment. He then receives the invoice, indicating successful payment. The order is then placed and also the database is updated. The user can then see the order status.

6. Railway Reservation System

Problem Statement

To design a Railway Reservation System that helps in railway ticket reservation and cancellation process. It provides an interface for the user to book tickets or cancel them at the same time from anywhere.

Software Requirement Specification

Purpose

The purpose of this document is to describe the features of the Railway Reservation System which provides the train timing details, reservation, billing and cancellation facilities. The system reduces the intervention of the user with the ticket counter and makes the process very convenient.

Scope

The scope of this system is to specify a reservation software that facilitates making a reservation from anywhere. It provides the ability to create a reservation from different places for a particular train. It also describes the other facilities needed to be provided by the system to make it more convenient and user-friendly.

Requirements

1. Functional Requirements

- a) Provide a railway search facility.
- b) Provide a ticket reservation facility.
- c) Provide a ticket cancellation facility.

2. Non-functional Requirements

- a) The database storing the details must be backed-up and secure.
- b) The system must provide secure data transfer between server and user system.
- c) The system must be very secure.
- d) The system must provide very fast response time.

3. User Requirements

- a) Options to reserve or cancel a ticket.
- b) A proper user-friendly interface.
- c) A screen that displays all user transactions.
- d) Secure payment methods.

Class Diagram

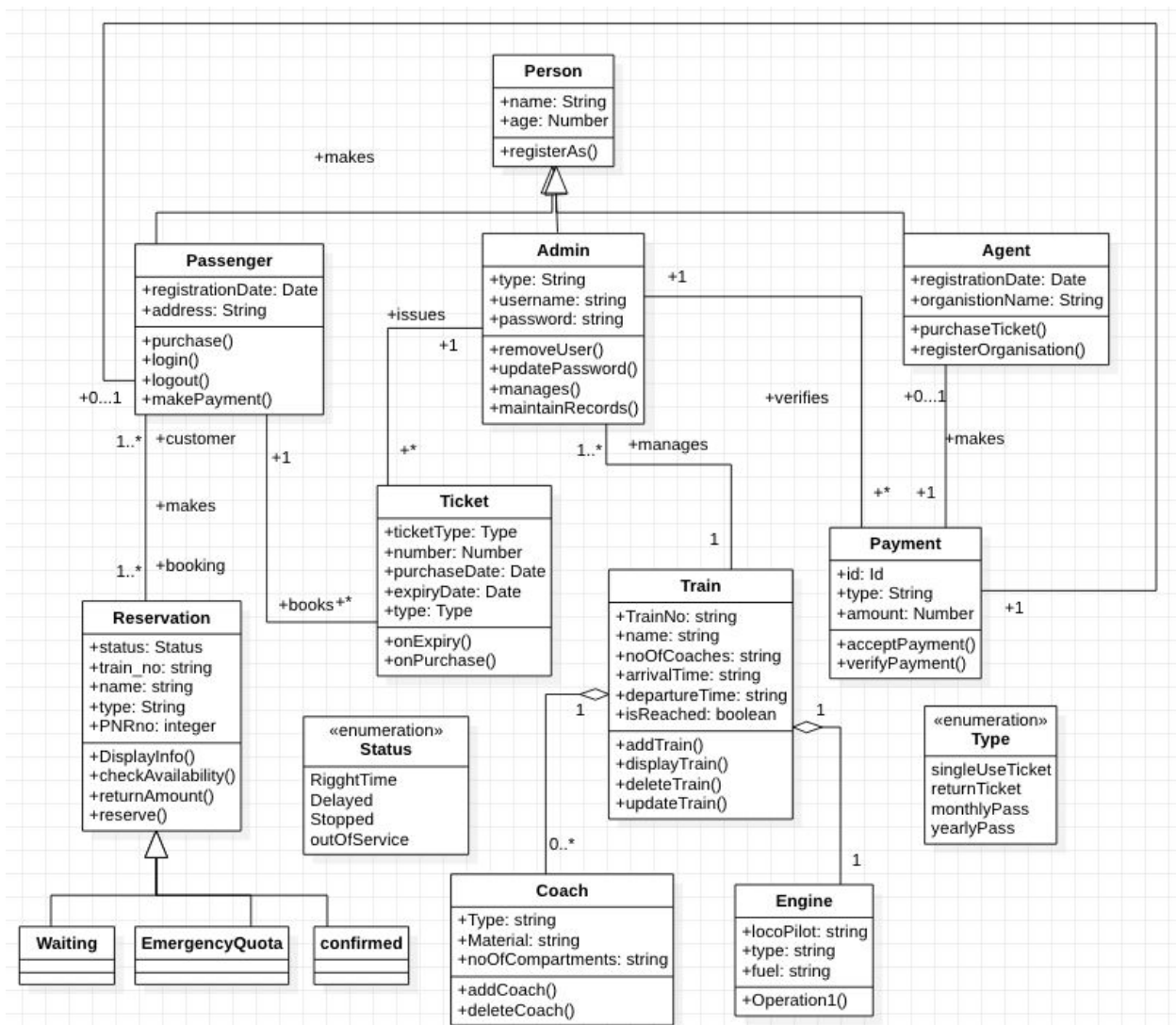


Figure 1 : Class Model

Admin Class: This class models an admin of the system.

Staff Class: This class contains the details of all the staff members.

Passenger Class: This class consists of passenger details.

Reservation Class: This class contains reservation details.

Train Class: This class contains train details.

Ticket Class: This class contains the details of the ticket booked.

Payment Class: This class contains payment details for each ticket.

Booking Class: This class contains booking details.

State Diagram

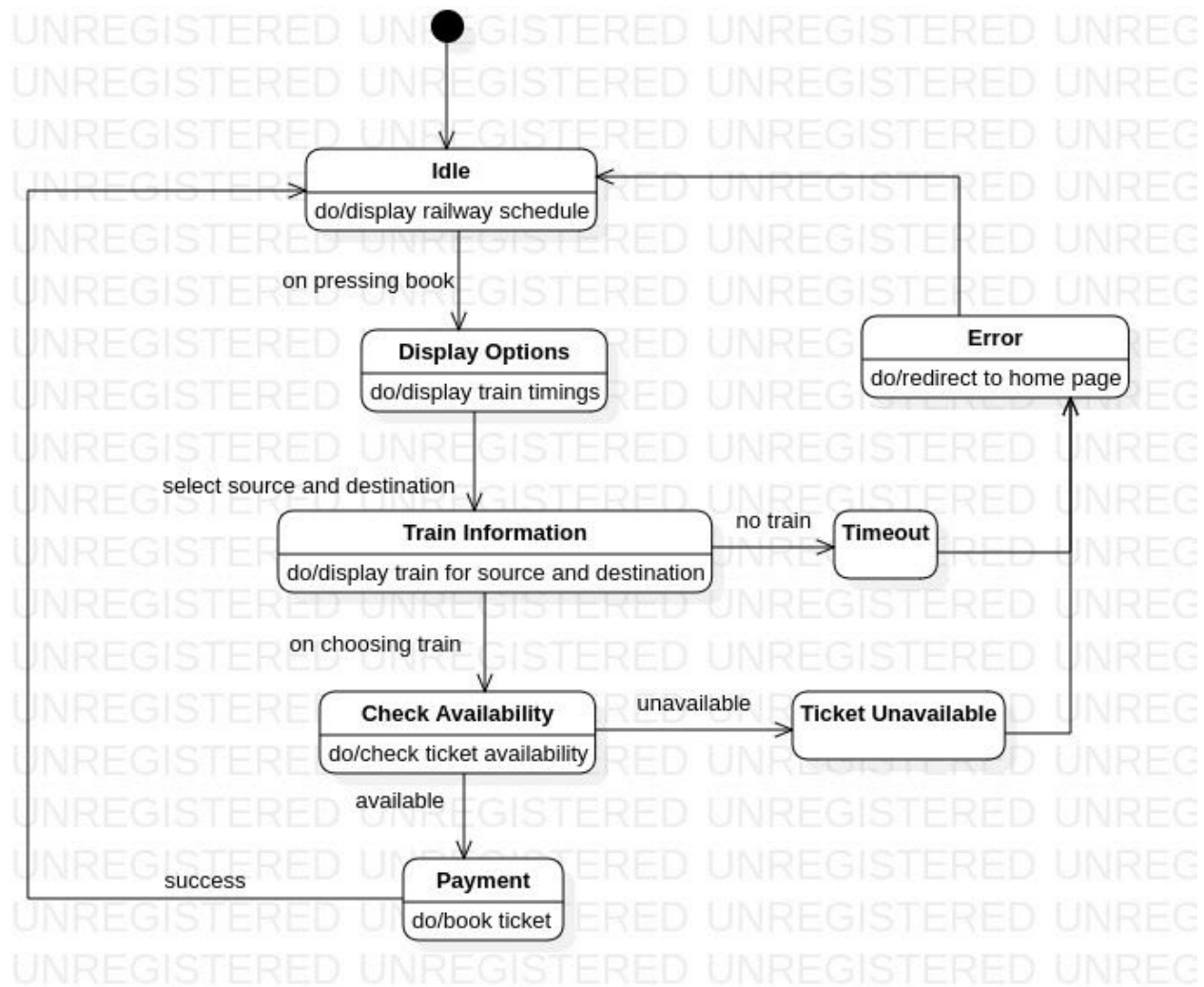


Figure 2 : State Model

This diagram describes the process of booking a ticket. On clicking the book button, you can choose to display train information and then choose a train. At last the payment is done and booking is confirmed. If no train is available, it goes to timeout state and then idle state.

Advanced State Diagram

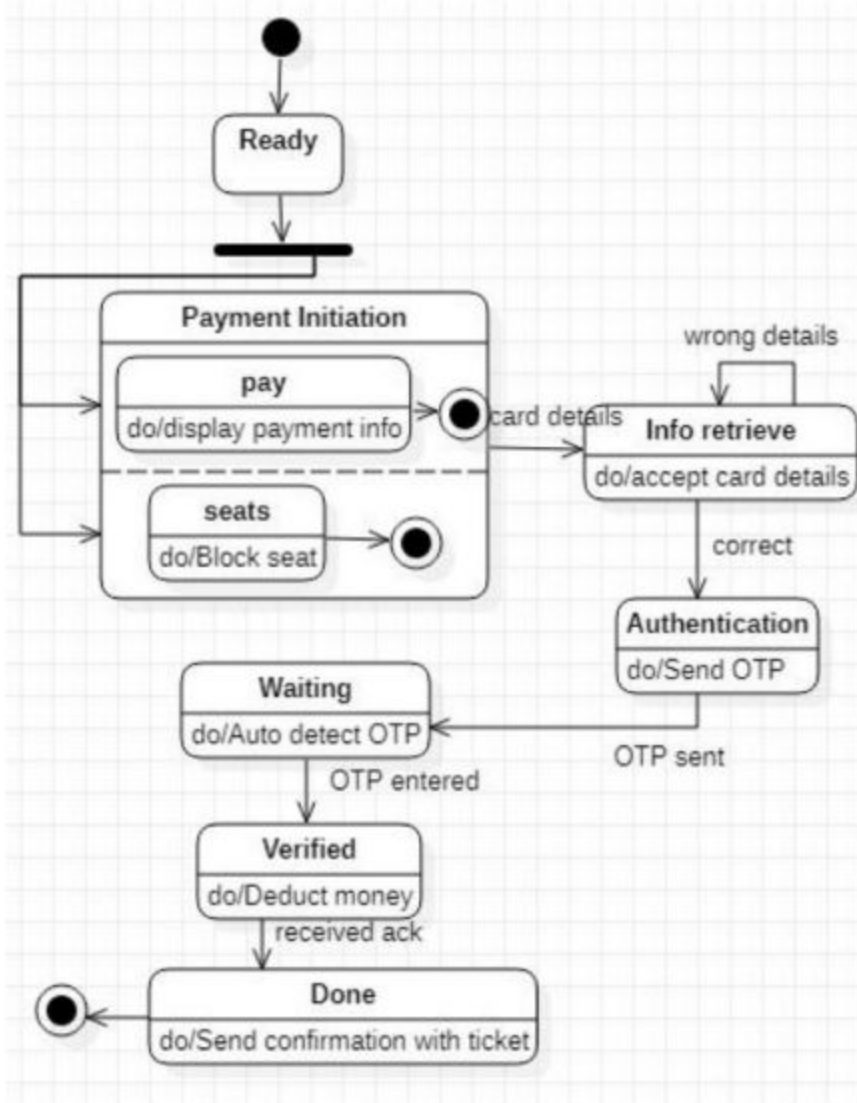


Figure 3 : Advanced State Model

The advanced state diagram has states for paying the ticket. From the ready state, the user goes to payment initiation after which the card details are accepted and an OTP is sent to the registered mobile number. On verification the money is deducted and ticket is sent to the user.

7. Graphics Editor

Problem Statement

To design a Graphics Editor System that helps to draw objects and build models with ease. The system should be flexible for professional model development as well as easy to use.

Software Requirement Specification

Purpose

The Graphics Editor System is an interactive application that allows the user to create, edit, layout, save, and print geometric models or graphs. The vision of the software is to provide an interactive yet simple way to design the models and export them as a soft copy or a hard one.

Scope

The system will be a desktop application to facilitate the user, who can use the system without the Internet. The software promises even the professional engineers an easy way to develop the graphical models without much effort.

Requirements

1. Functional Requirements

- a) Provide the option of editing, creating and saving the diagrams.
- b) Provide an option to print the diagrams created.
- c) Provide an option to convert the diagram into many formats such as jpeg, pdf, etc.

2. Non-functional Requirements

- a) Provide an efficient response time.
- b) Provide compatibility to various versions of the operating system.
- c) Continuous updates to improve the user interface and performance of the system.

3. User Requirements

- a) Provide a blank canvas where the user can start making diagrams.
- b) Provide an option to save diagrams.
- c) Provide an option to export the file into any format.
- d) Provide an option to import existing diagrams.

Class Diagram

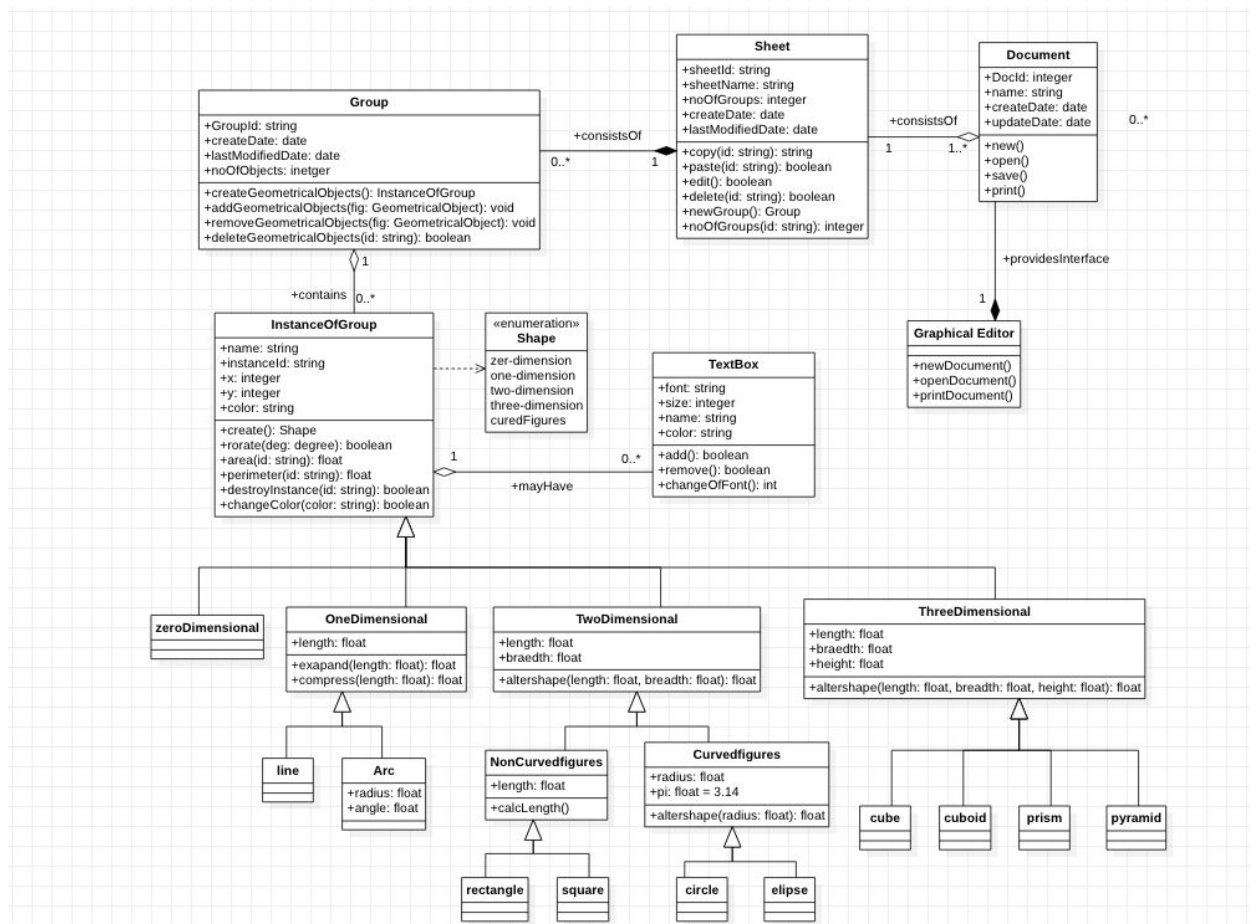


Figure 1 : Class Model

Graphical Editor Class: The main class of the system.

Document Class: The class which contains details about every new document created.

Sheet Class: This class contains the details of each sheet in a document.

TextBox Class: This class models a text box.

Group Class: Defines each group of objects uniquely.

OneDimensional Class : This class models 1D objects.

TwoDimensional Class : This class models 2D objects.

ThreeDimensional Class : This class models 3D objects.

State Diagram

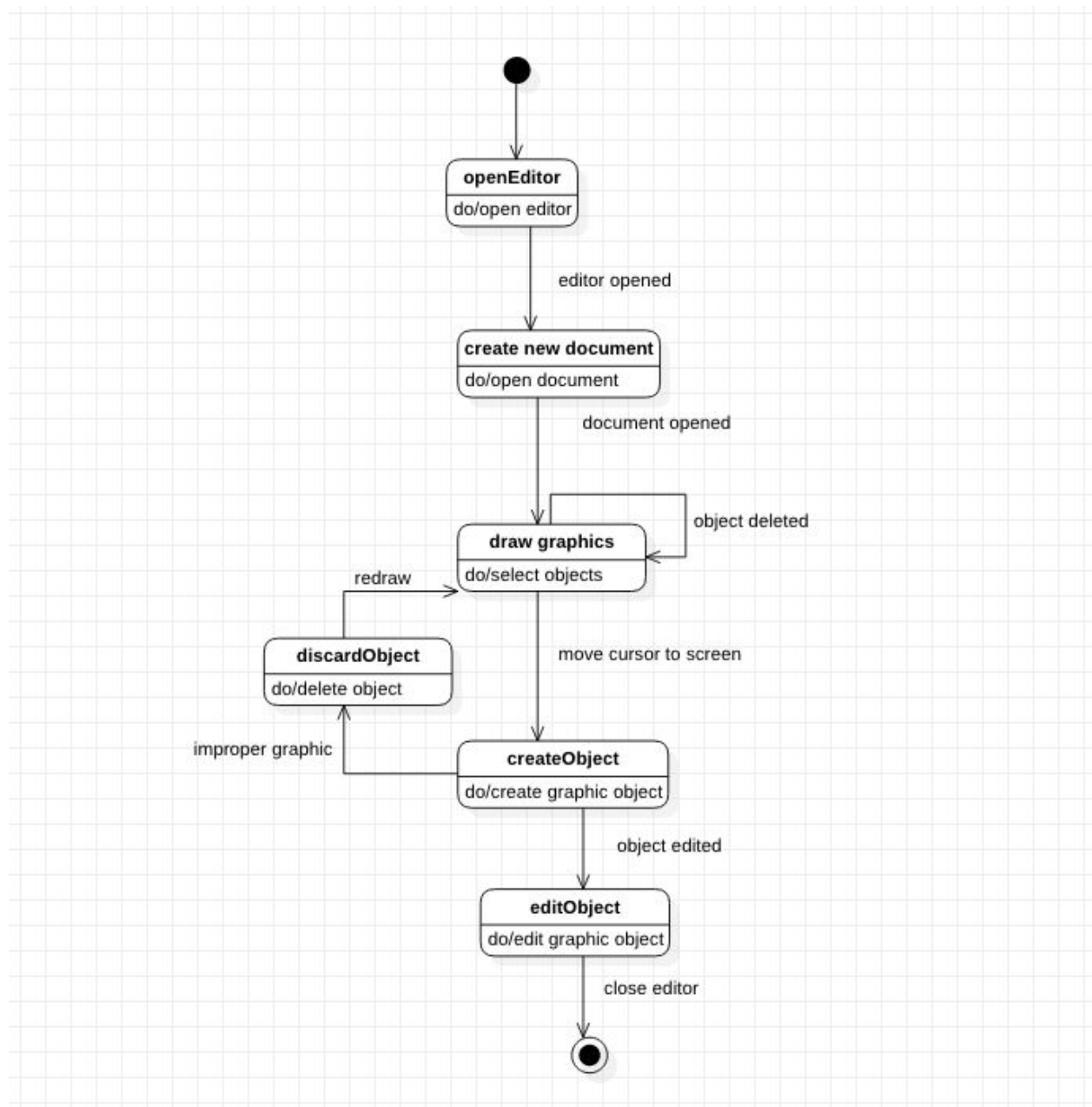


Figure 2 : State Model

This diagram describes the process of opening a document, drawing objects and other things and saving them. The first step is to open an editor and you start anew or open a pre-saved document. An object after creation can be discarded too. You select an object to draw. Fill the form details of the parameters of the object and click in the position to place the object. Enter the text label. Draw another object or save the current state and exit.

Advanced State Diagram

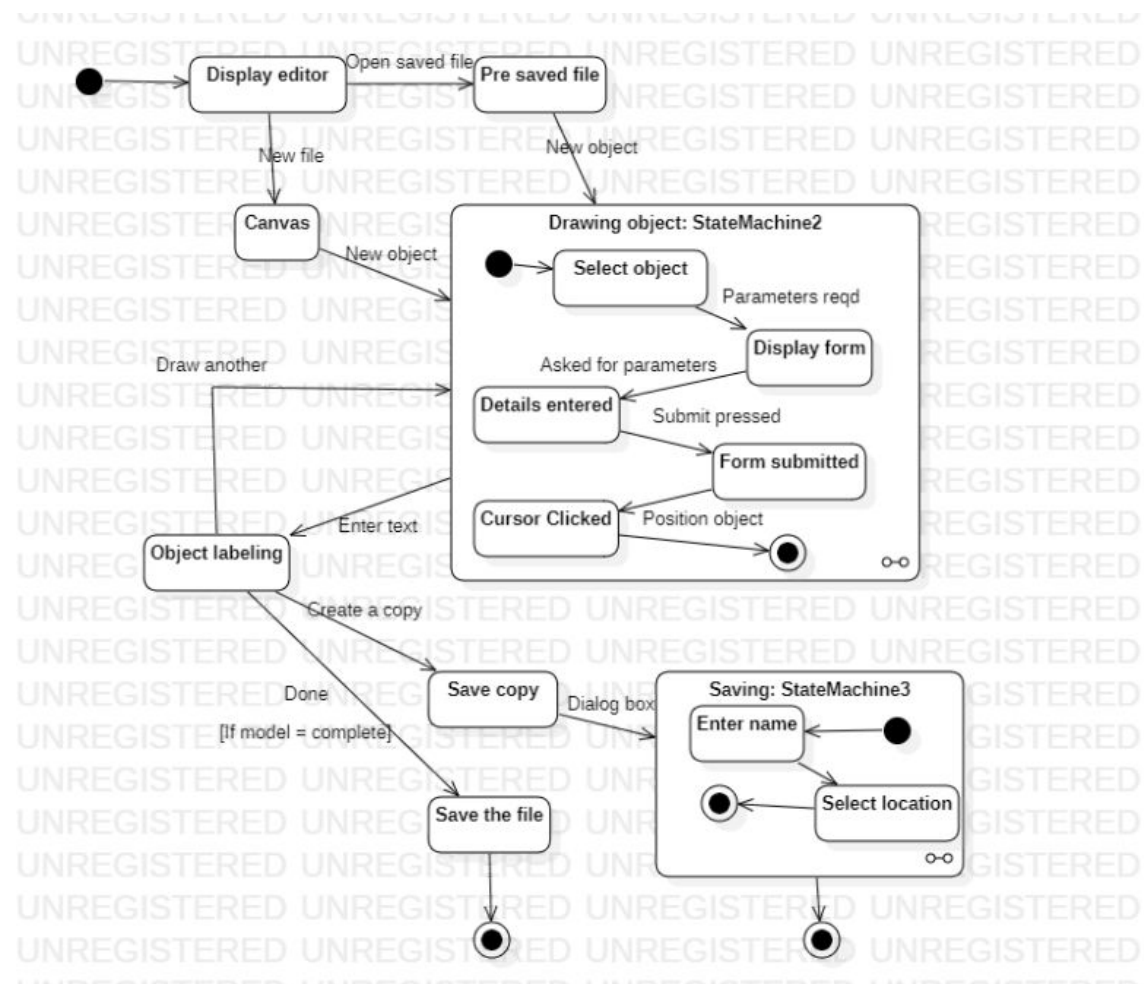


Figure 3 : Advanced State Model

This diagram also describes the same process but in a more detailed way. Open the editor to start a new or pre-saved file to resume. Drawing an object is taken as a complete process and the object selection and entering the details etc are taken as sub-states. The object is labeled. Then you can save the same file or make another copy. Saving is taken as a whole state and entering name and location are taken as sub-states.