Object Oriented Analysis and Design (OOAD)

Final Assignment

Train seat reservation system

Group number: 6

Name: MEHEDI HASAN RABBI ID: 20-44059-2

Name: PROSIT KUMAR DAS ID:20-44063-2

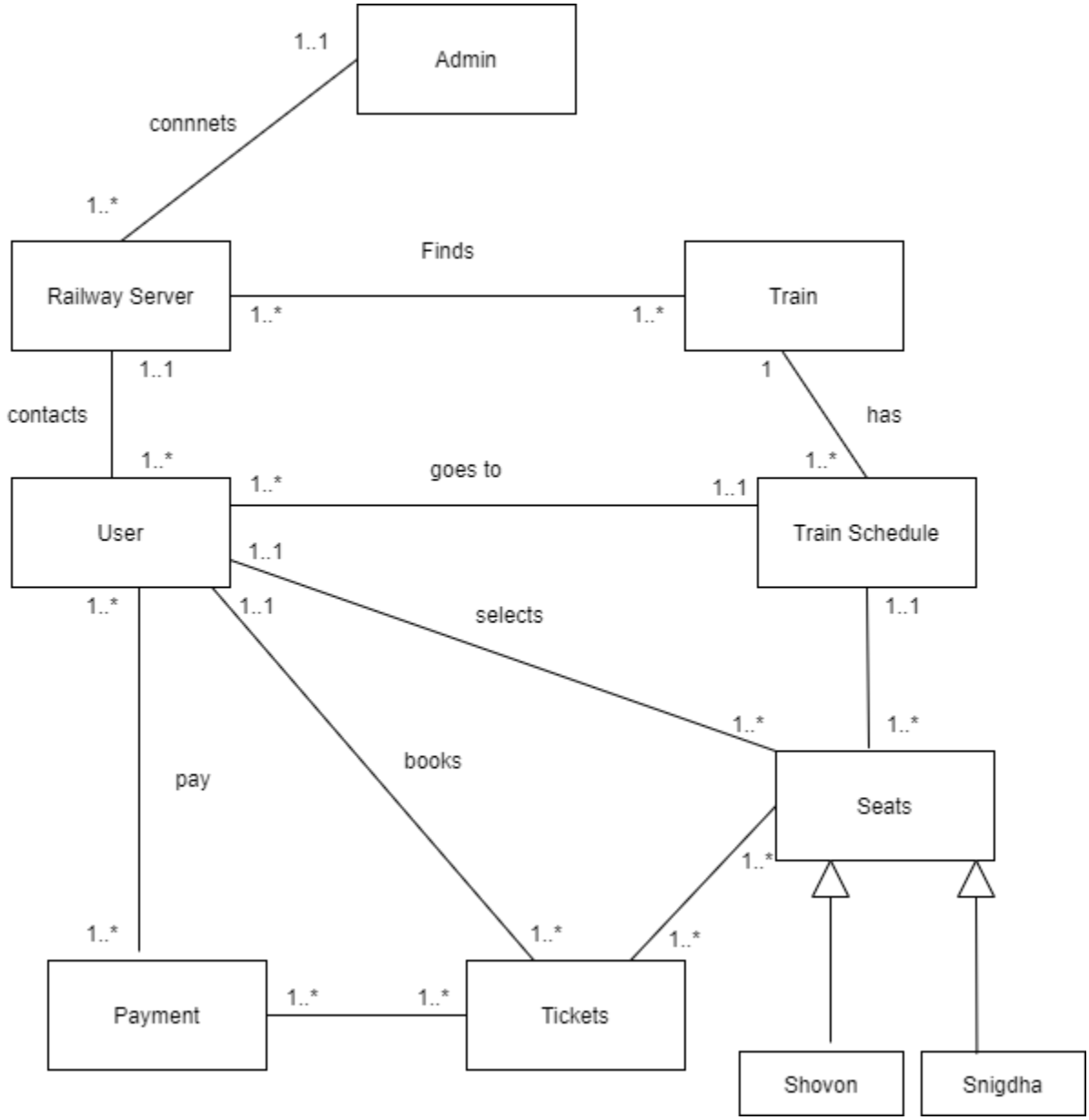
Section: D

Train seat reservation system

In a train seat reservation system, A user (customer/railway employee) will search for train in railway server. To know the availability the user, must select the details of the train so that he/she can be sure about confirming. First, he/she will see the destination and select one. Then he/she will be asked to select the preferable date and time. After selecting, the server will see if there is any train available of that time. If yes then server will ask for further details and if not, the server will deny. For the available train, user will select the train and check no. of seat which are available. Then server asks if the user is interested to go further for reservation. Then comes the booking process. To confirm the seat the user, have to login to the server as a passenger or railway employee. After verifying the information of the user by system, the user will select train details and then system will show the amount to pay. If the user wants to confirm the reservation, he/she must pay the bill otherwise the ticket will be cancelled. To pay the amount the user will enter the credit card info and verify it or cash. If the information is nor verified it will ask again to enter correct details. After verifying the user will pay the bill and confirm the reservation. After confirming if user want she/he can cancel reservation

USE CASE DIAGRAM

CLASS DIAGRAM



Detail design of 3 classes

User class
- user_id: integer{assigned by system} + user_name: String{Maximum 35 Characters} + user_address: String - user_mobile: Integer - user_email : String - user_password : String
+ setUser_id (Id: integer) + getUser_name ( ):String + getUser_address ( ): String + getUser_mobile: Integer + getUser_email : String - setUser_password (password: String)

Payment class
-Payment_id: integer{assigned by system} - payment_user_id: integer + payment_description: String + payment_amount: Integer + payment_date : Date
- set Payment_id (payment_Id: integer) - setPayment_user_id(Id: integer) + setPayment_description (description: String) + setPayment_amount(amount : integer) + setPayment_date (Date : date)

Train class
-train_id: integer{assigned by system} + train_name: String{Maximum 35 Characters} + train_type: String + train_description: String + train_number: Integer + train_ticket : String
-setTrain_id (train_Id: integer) + setTrain_name ( ):String + setTrain_type ( ): String + setTrain_description ( ) : String + setTrain_number ( ) : Integer +getTrain_ticket ( ) : String

Design 3 CRC Card for 3 classes

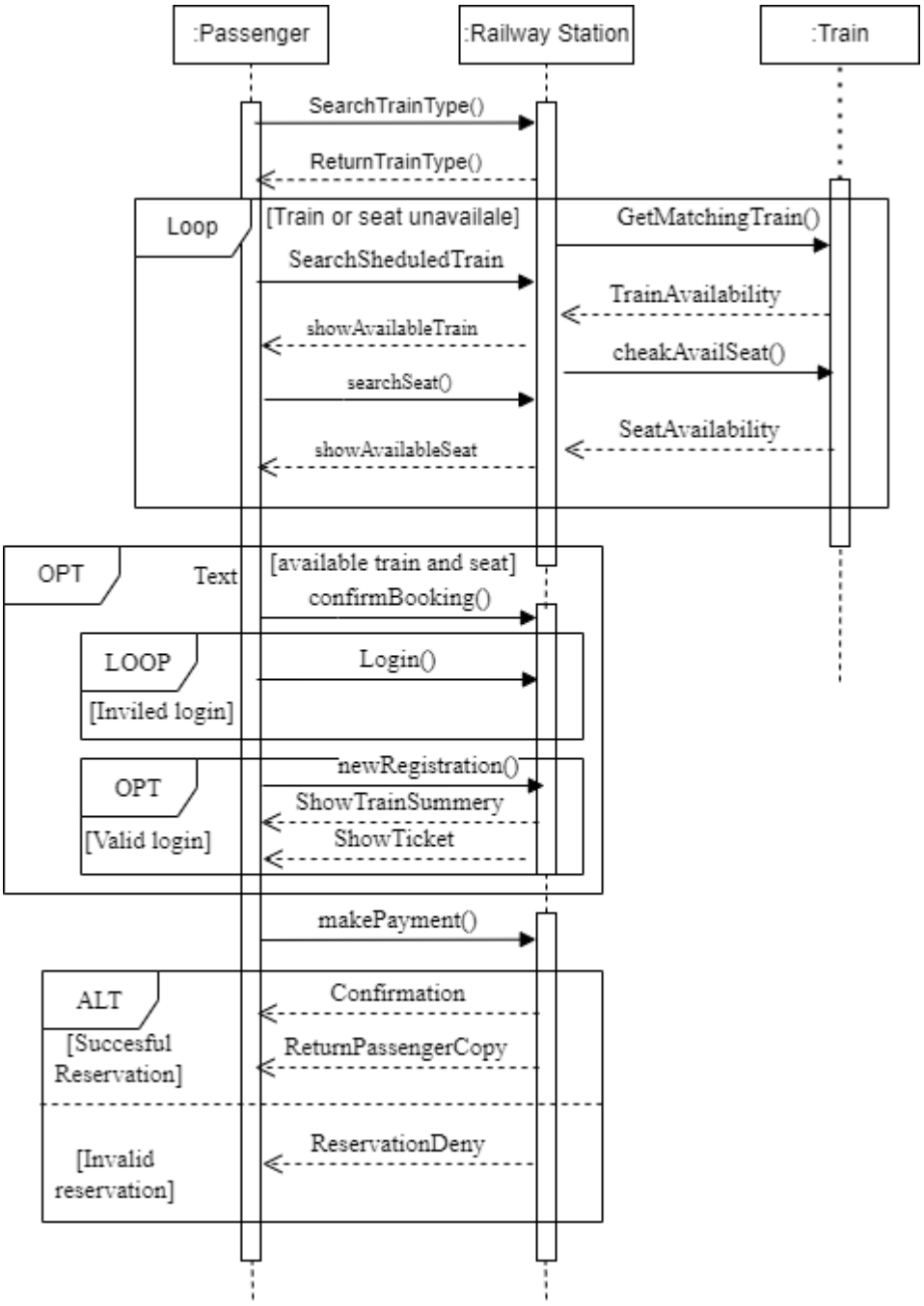
Class Name	Payment
Description	Keeps track of balance and transaction for each instance of user
Superclass	Railway server
Subclass	
Responsibilities	**Collaborators**
Keep Track of Tickets	Tickets
Keep track of user	user

Class Name	User
Description	User can be a customer or a railway employee. who log in to buy tickets
Superclass	Railway server
Subclass	
Responsibilities	**Collaborators**
can select seats	Seats
Can book tickets	Tickets
Can see train arriving and departure time	Train schedule

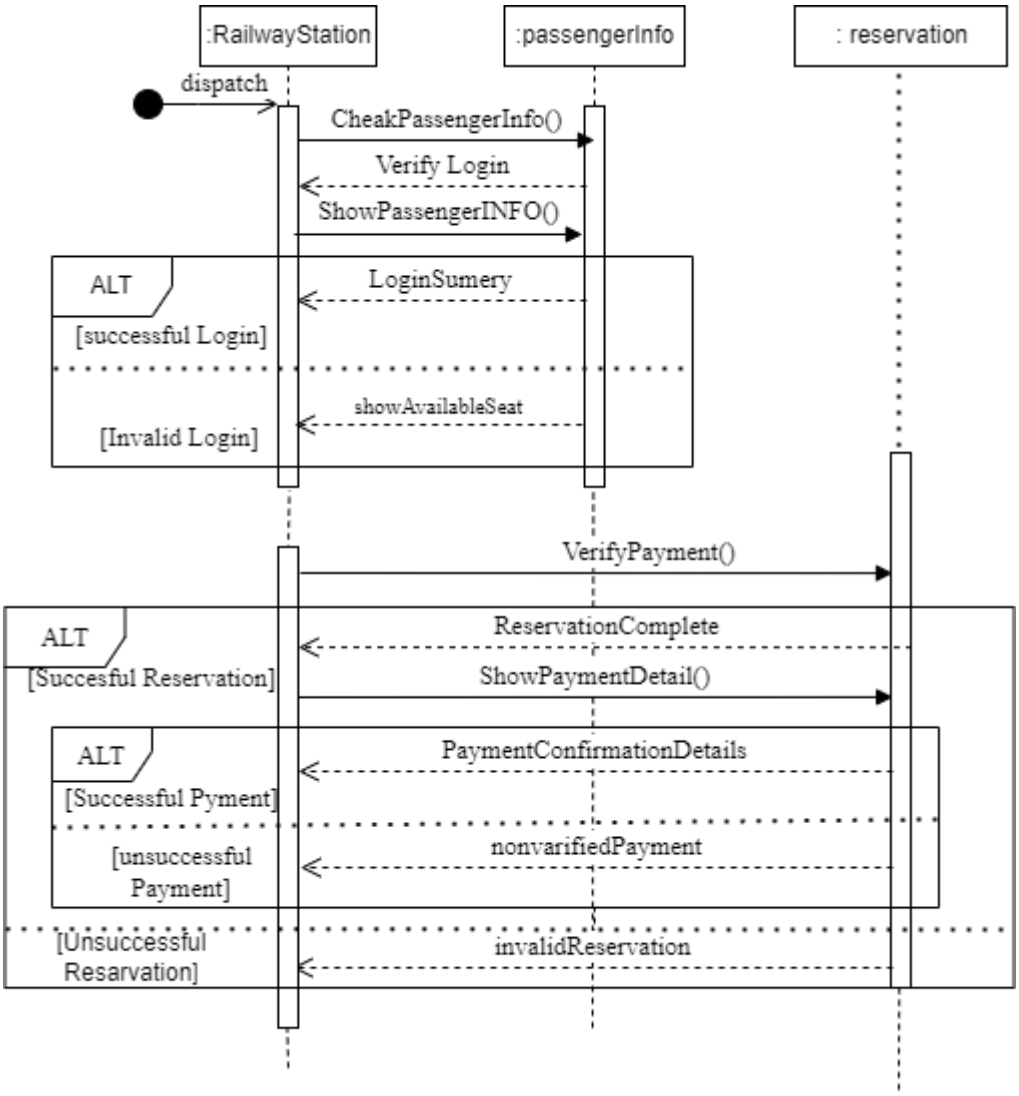
Class Name	Train
Description	A user (customer/railway employee) will search for train in railway server. Every train is unique by name and route and arriving and departure time.
Superclass	Railway server
Subclass	
Responsibilities	**Collaborators**
Has own capacity	Seats
Has own time schedule	Train schedule

Sequence Diagram of the project

Sequence diagram for select train :

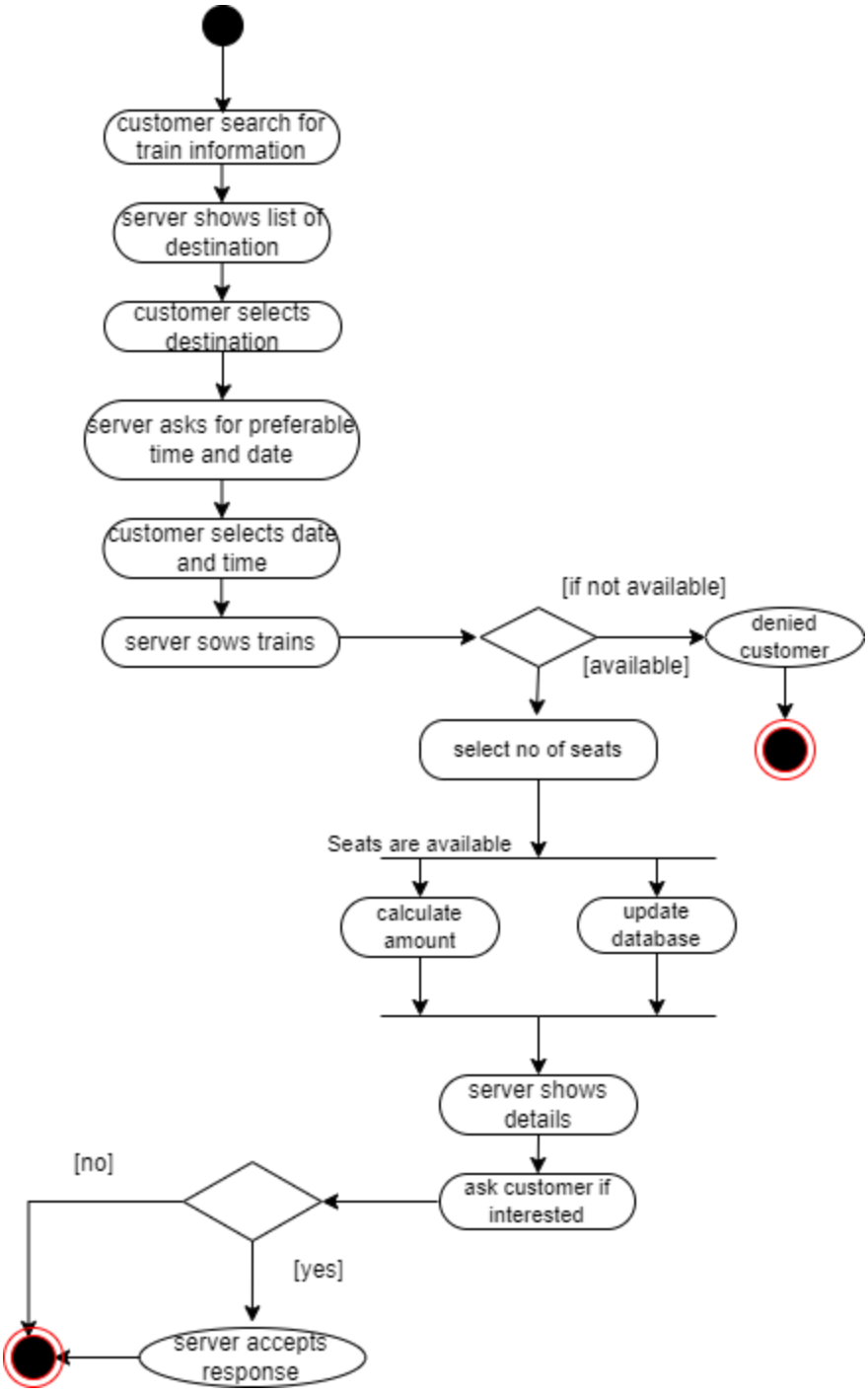


Sequence diagram for confirm payment:

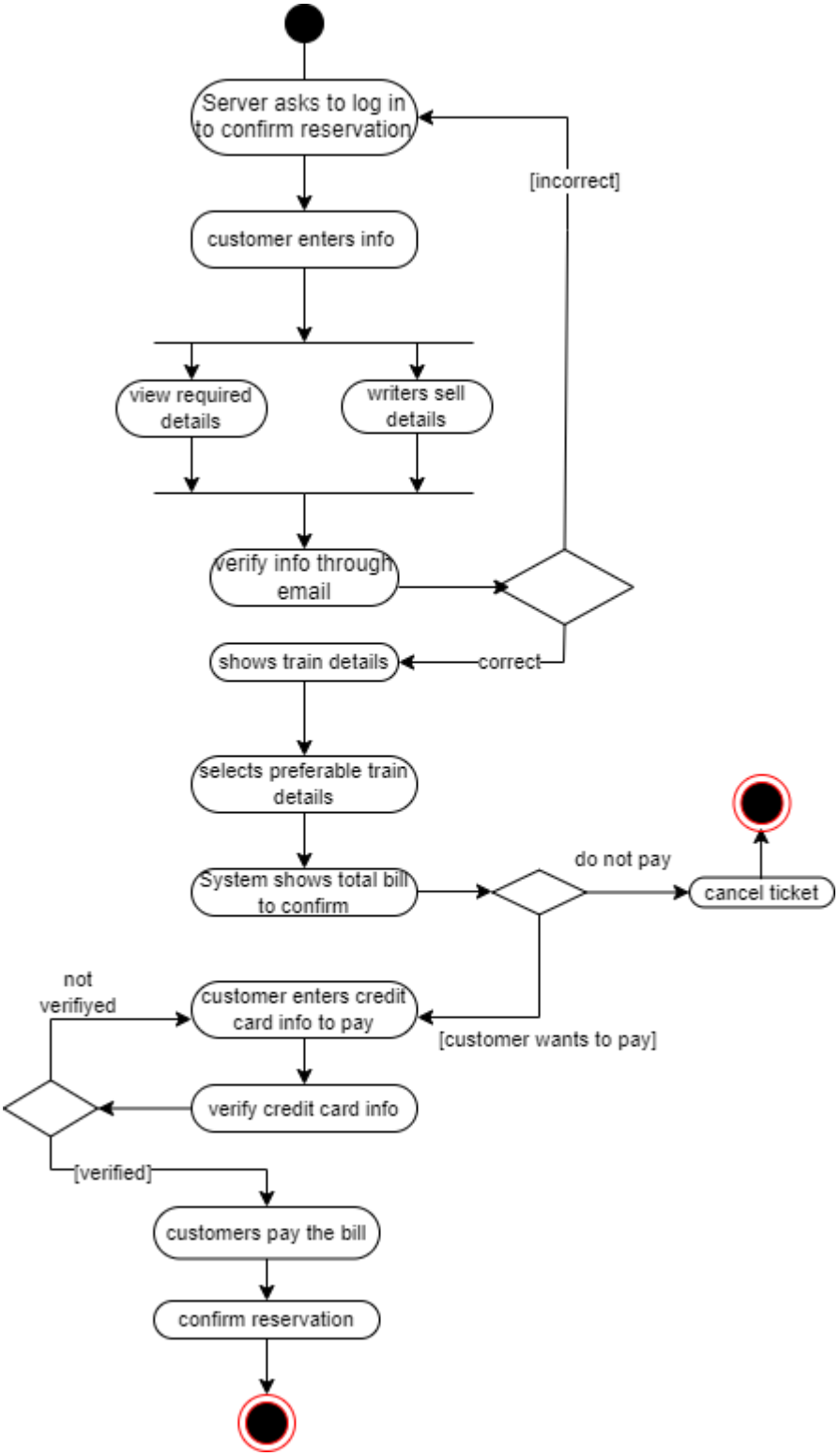


Activity Diagrams:

Activity diagram for checking Train info:

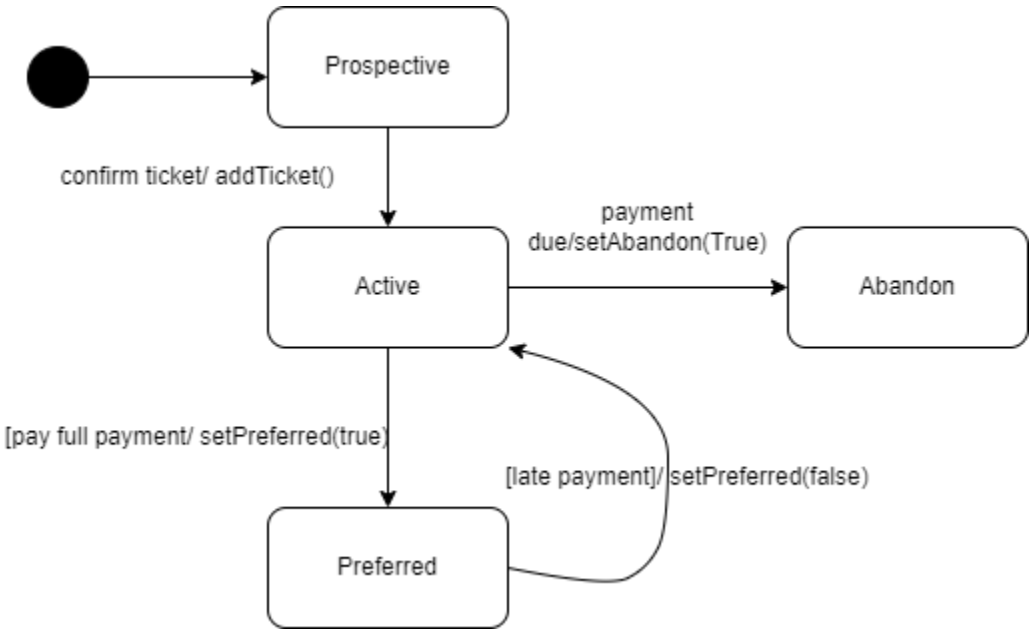


Activity diagram for confirm reservation:

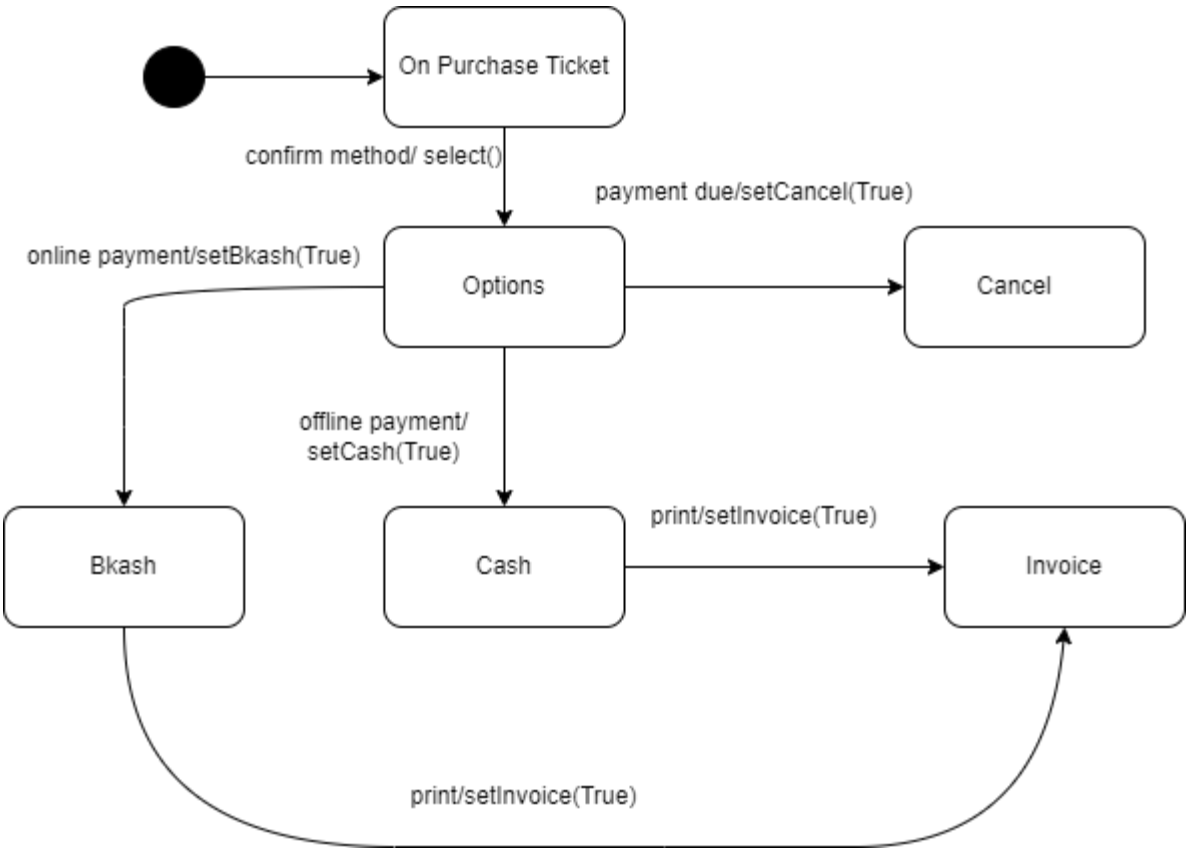


Statechart Diagrams:

Statechart diagram for customer object:



Statechart diagram for payment object:



LCOM(Lack of Cohesion among Methods) of two classes:

User class:

User class
- user_id: integer{assigned by system} + user_name: String{Maximum 35 Characters} + user_address: String - user_mobile: Integer - user_email : String - user_password : String
+ setUser_id (Id: integer) + getUser_name ( ):String + getUser_address ( ): String + getUser_mobile : Integer + getUser_email : String - setUser_password (password: String)

Pair:

setUser_id (Id: integer), getUser_name ( ):String = Non-cohesive pairs

setUser_id (Id: integer), getUser_address ( ): String = Non-cohesive pairs

setUser_id (Id: integer), getUser_mobile : Integer = Non-cohesive pairs

setUser_id (Id: integer) , getUser_email : String = Non-cohesive pairs

setUser_id (Id: integer), setUser_password (password: String) = Non-cohesive pairs

getUser_name ( ):String, getUser_mobile : Integer = Non-cohesive pairs

getUser_name ( ):String, getUser_email : String = Non-cohesive pairs

getUser_name ( ):String, getUser_address ( ): String = Non-cohesive pairs

getUser_name ( ):String, setUser_password (password: String) = Non-cohesive pairs

getUser_address ( ): String, getUser_mobile : Integer= Non-cohesive pairs

getUser_address ( ): String, getUser_email : String= Non-cohesive pairs

getUser_address ( ): String, setUser_password (password: String)= Non-cohesive pairs

getUser_mobile : Integer, getUser_email : String= Non-cohesive pairs

getUser_mobile : Integer, setUser_password (password: String)= Non-cohesive pairs

getUser_email : String, setUser_password (password: String)= Non-cohesive pairs

P (Non-cohesive pairs)=15

Q (Cohesive pair)=0

LCOM=|P|-|QI

= 15-0

=15

Comment : The LCOM value of the class indicates that the methods of the class are non-cohesive, and it is not a desirable design.

Train class :

Train class
-train_id: integer{assigned by system} + train_name: String{Maximum 35 Characters} + train_type: String + train_description: String + train_number: Integer + train_ticket : String
-setTrain_id (train_Id: integer) + setTrain_name ( ):String + setTrain_type ( ): String + setTrain_description ( ) : String + setTrain_number ( ) : Integer +getTrain_ticket ( ) : String

Pair:

setTrain_id (train_Id: integer), setTrain_name ( ):String = Non-cohesive

setTrain_id (train_Id: integer), setTrain_type ( ): String = Non-cohesive

setTrain_id (train_Id: integer), setTrain_description ( ) : String = Non-cohesive

setTrain_id (train_Id: integer), setTrain_number ( ) : Integer = Non-cohesive

setTrain_id (train_Id: integer), getTrain_ticket ( ) : String = Non-cohesive

setTrain_name ( ):String, setTrain_type ( ): String = Non-cohesive

setTrain_name ( ):String, setTrain_description ( ) : String = Non-cohesive

setTrain_name ( ):String, setTrain_number ( ) : Integer = Non-cohesive

setTrain_name ( ):String, getTrain_ticket ( ) : String = Non-cohesive

setTrain_type ( ): String, setTrain_description ( ) : String = Non-cohesive

setTrain_type ( ): String, setTrain_number ( ) : Integer = Non-cohesive

setTrain_type ( ): String, getTrain_ticket ( ) : String = Non-cohesive

setTrain_description ( ) : String, setTrain_number ( ) : Integer = Non-cohesive

setTrain_description ( ) : String, getTrain_ticket ( ) : String = Non-cohesive

setTrain_number ( ) : Integer, getTrain_ticket ( ) : String = Non-cohesive

P (Non-cohesive pairs)=15

Q (Cohesive pair)=0

LCOM=|P|-|QI

= 15-0

=15

Comment : The LCOM value of the class indicates that the methods of the class are non-cohesive, and it is not a desirable design.