



# Final Assignment

Name of the topic : **ONLINE TRANSPORT TICKET BOOKING SYSTEM**

Group No : 05

Course Name : Object Oriented Analysis & Design

Course Teacher : S. A. M. MANZUR HOSSAIN KHAN

Deadline : 30-04-2021

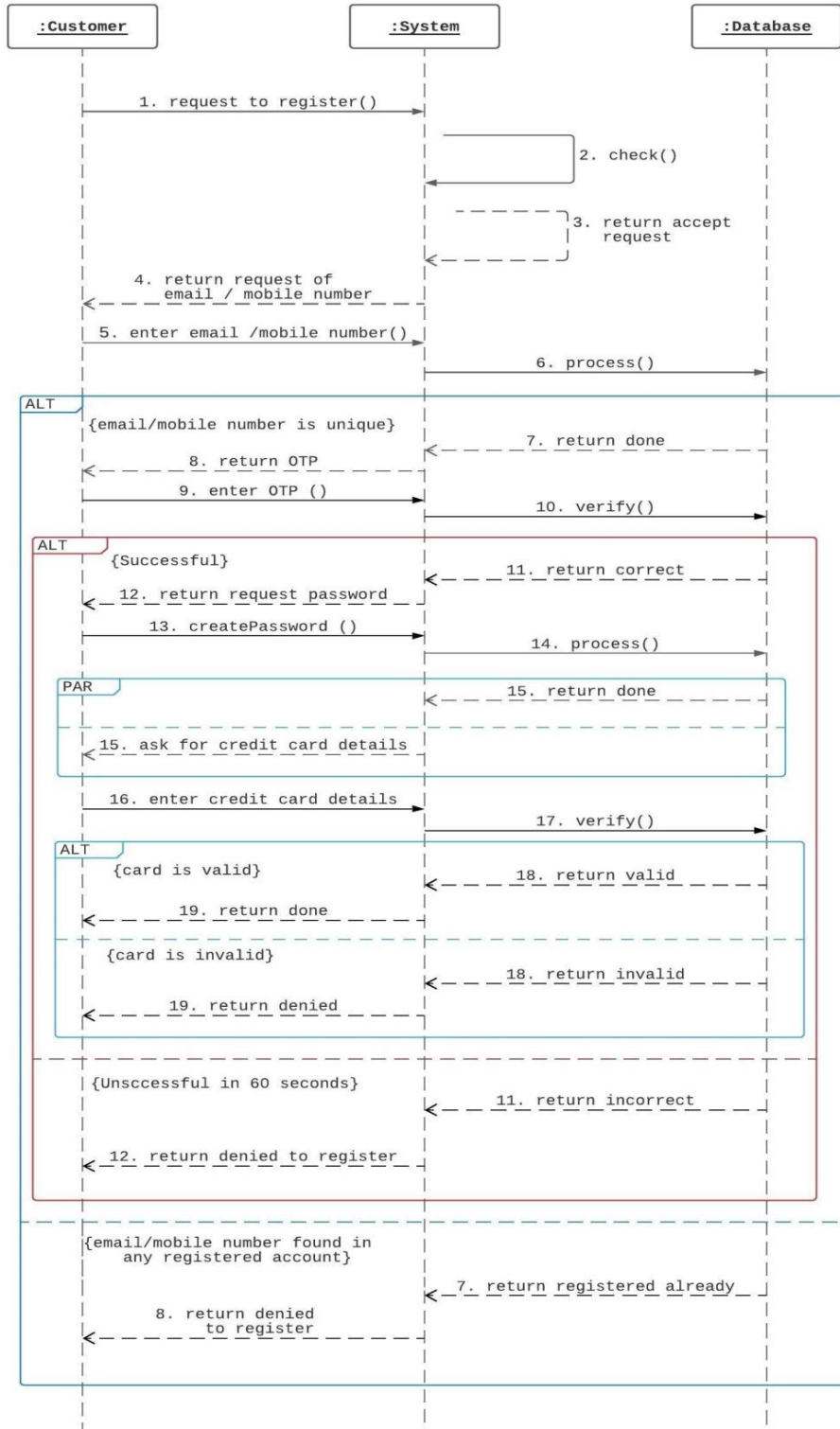
Name	ID	SECTION
<b>Sarzila Sahrin Jisha</b>	20-42526-1	E
<b>Mahmud Iftekhar Asef</b>	20-42547 -1	E
<b>Musfiqur Rahman</b>	20-42558-1	E

## ***ONLINE TRANSPORT TICKET BOOKING SYSTEM***

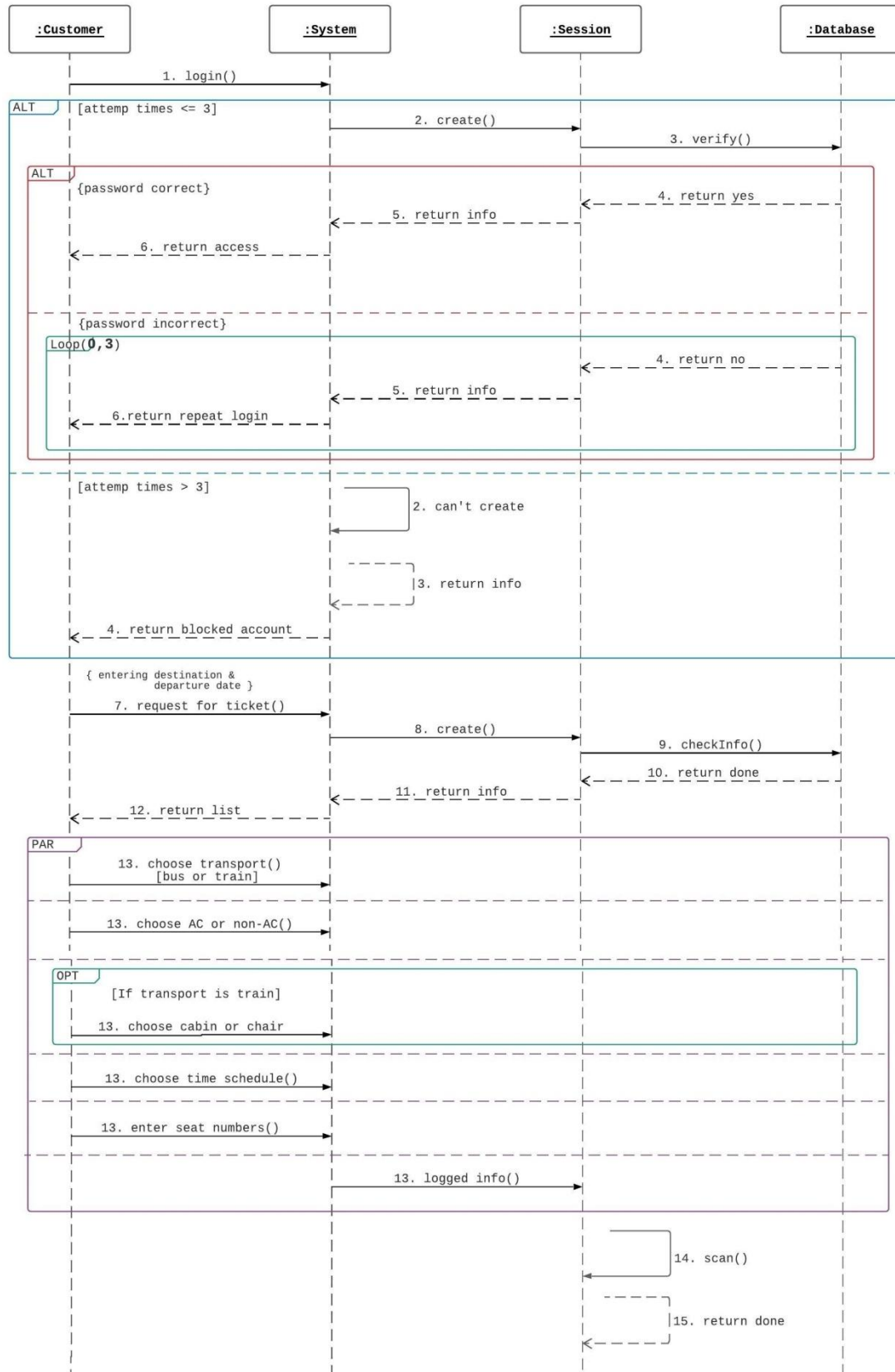
An online transport ticket booking system is a management system that acts as a pre-booking of ticket. Each customer needs an account. A new customer requests to register an account, he is asked to give his email or mobile number. If the email or mobile number isn't registered before, the system sends one OTP back to him. Then, if the customer doesn't make entry of the correct OTP in 60 seconds, his request will be failed. If the OPT is correct, customer has to enter a 6 digits password and a valid credit card details to complete the process. Then, customer needs to log in their account, which is verified by system. If the entered password is incorrect, system requests for the password and the customer enters password again. The verification repeated for 3 times for an incorrect password. If password is incorrect after three times attempt, the account will be blocked for 1 hour. If the entered password is correct, customer gets access. After login, a customer places a request to the system for booking a ticket, entering departure & arrival destinations and the date of departure. Checking these informations, system sends a list to customer. Customer chooses his transport preference such as bus or train, AC or non-AC, cabin or chair (for train only), time schedule as well as enters needed number of seats and logged these informations at the same time. System scans these informations and if customer preference is available, system noted them otherwise cancels the ticket request. Then, system checks credit card balance and if it is sufficient to buy the ticket, then system sends the ticket no and confirmed the booking simultaneously. But if the balance isn't sufficient, system cancels the ticket request. If any account is not used within 2 years, then it will be archived.

## a. Sequence Diagram

Sequence Diagram - 01 : Register Customer Account



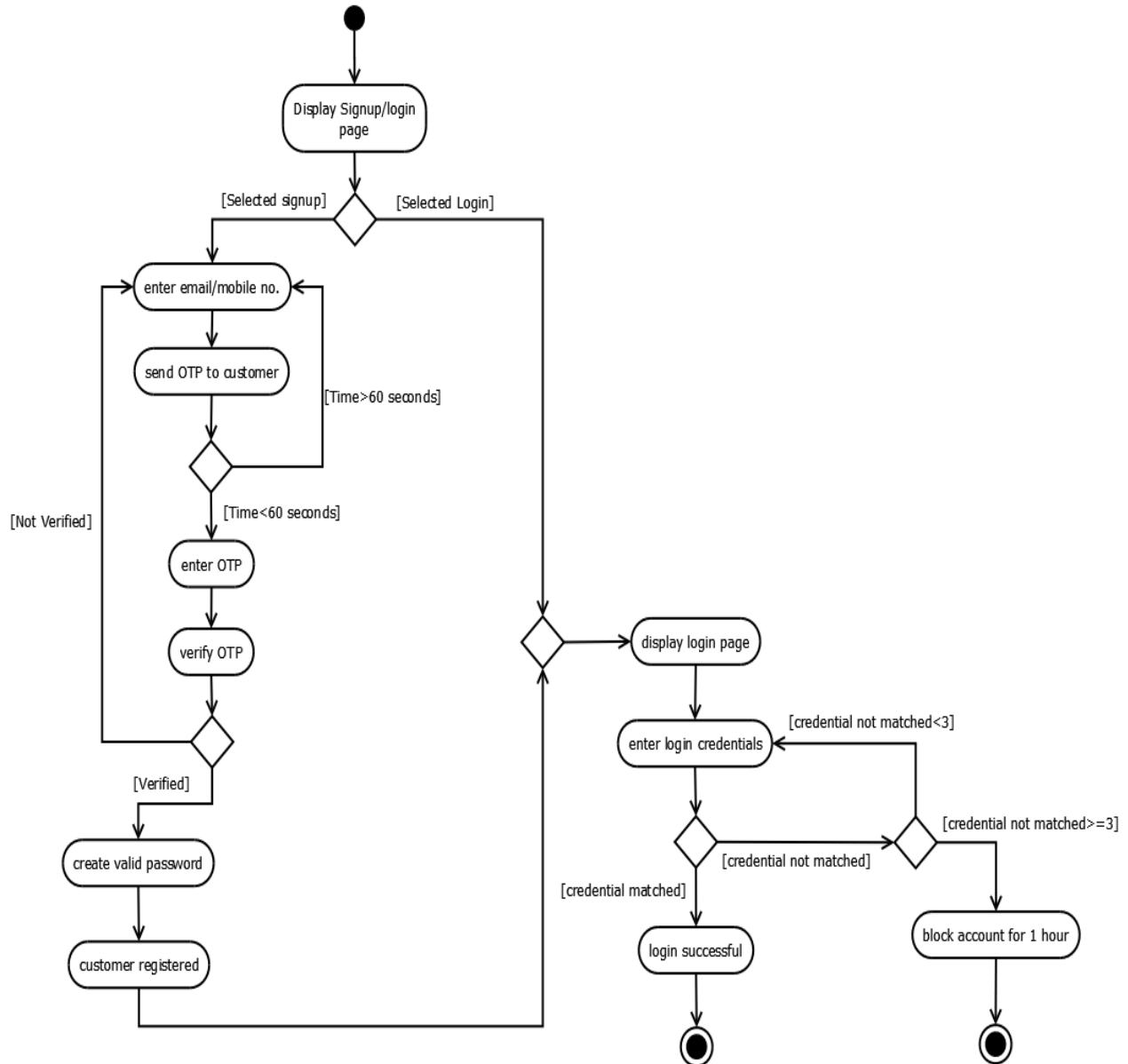
## Sequence Diagram - 02 : Ticket Booking



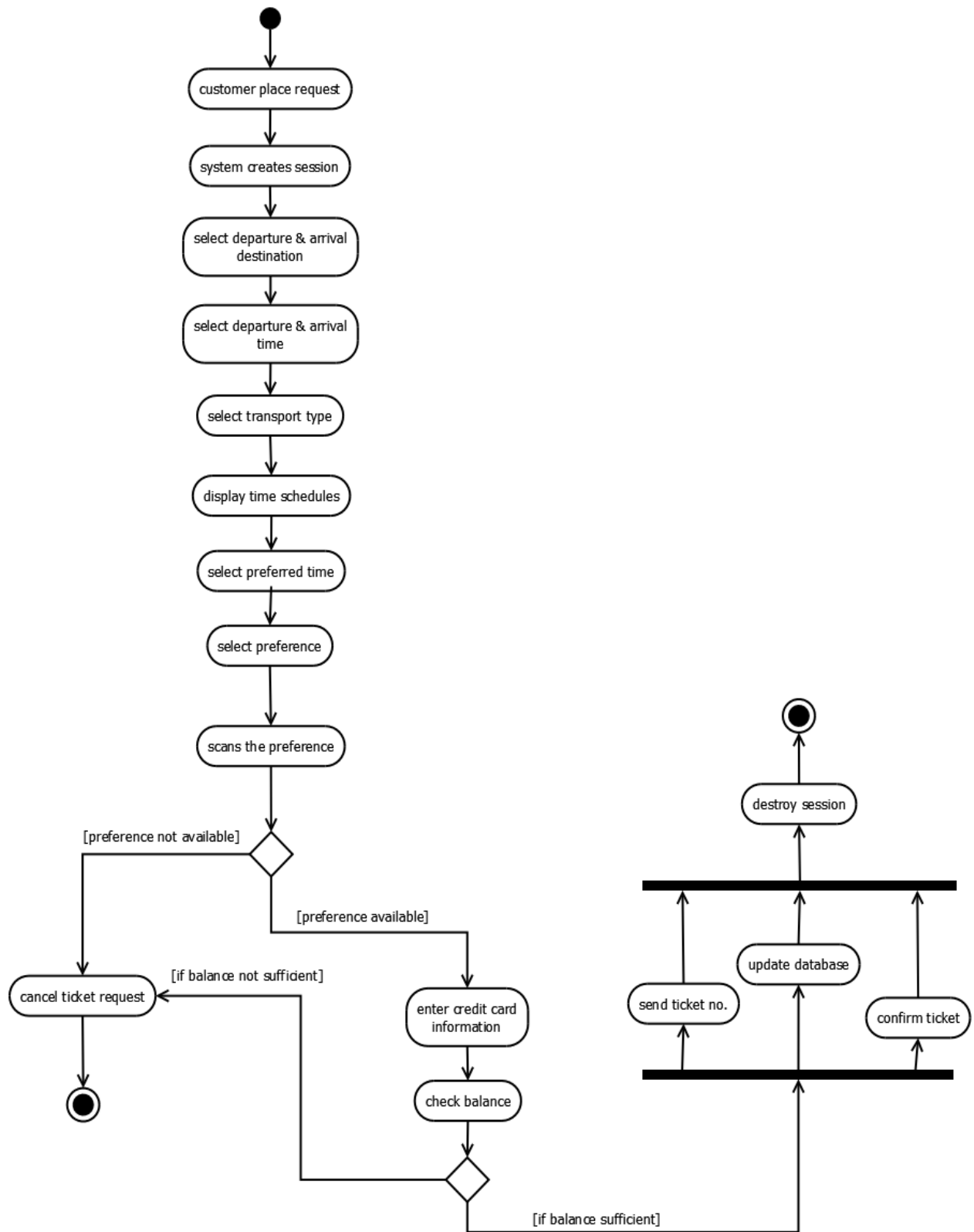


## b. Activity Diagram

Activity Diagram - 01 : Login Customer Account

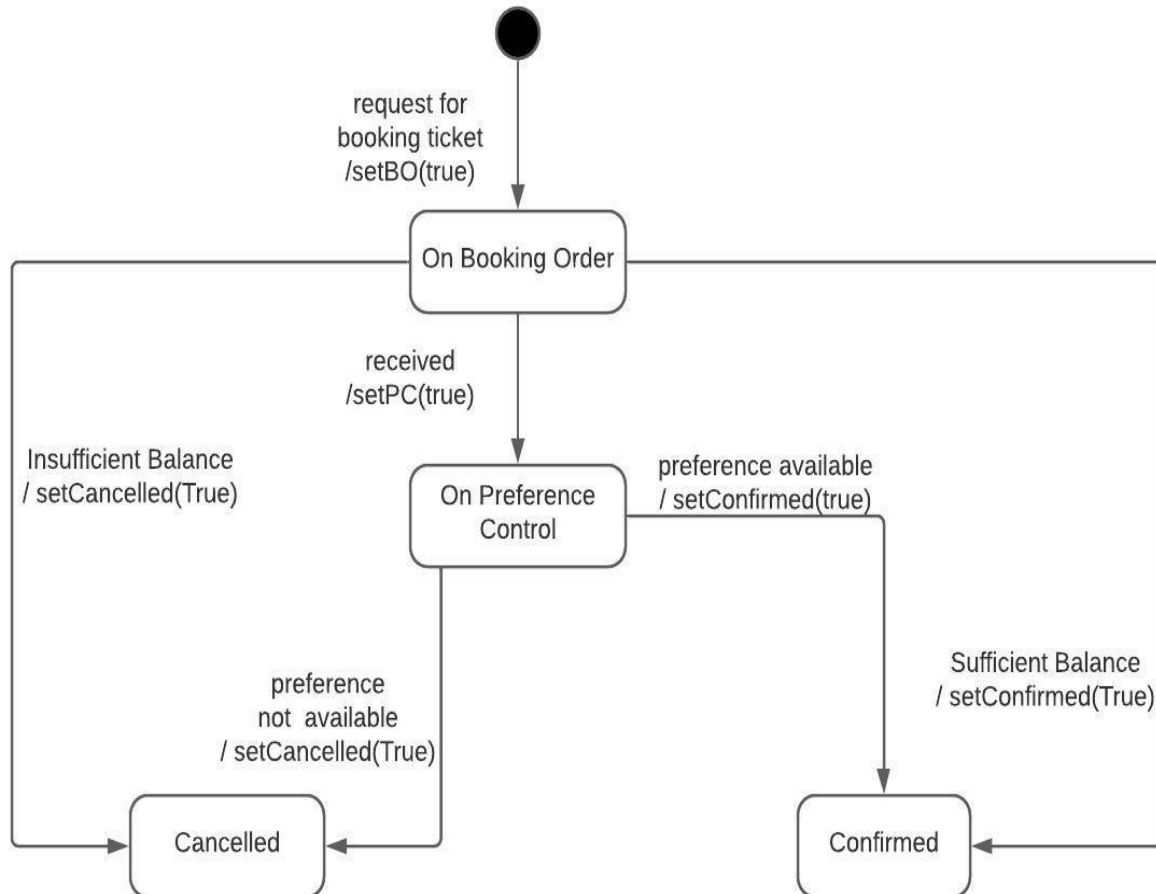


## Activity Diagram - 02 : Ticket Booking



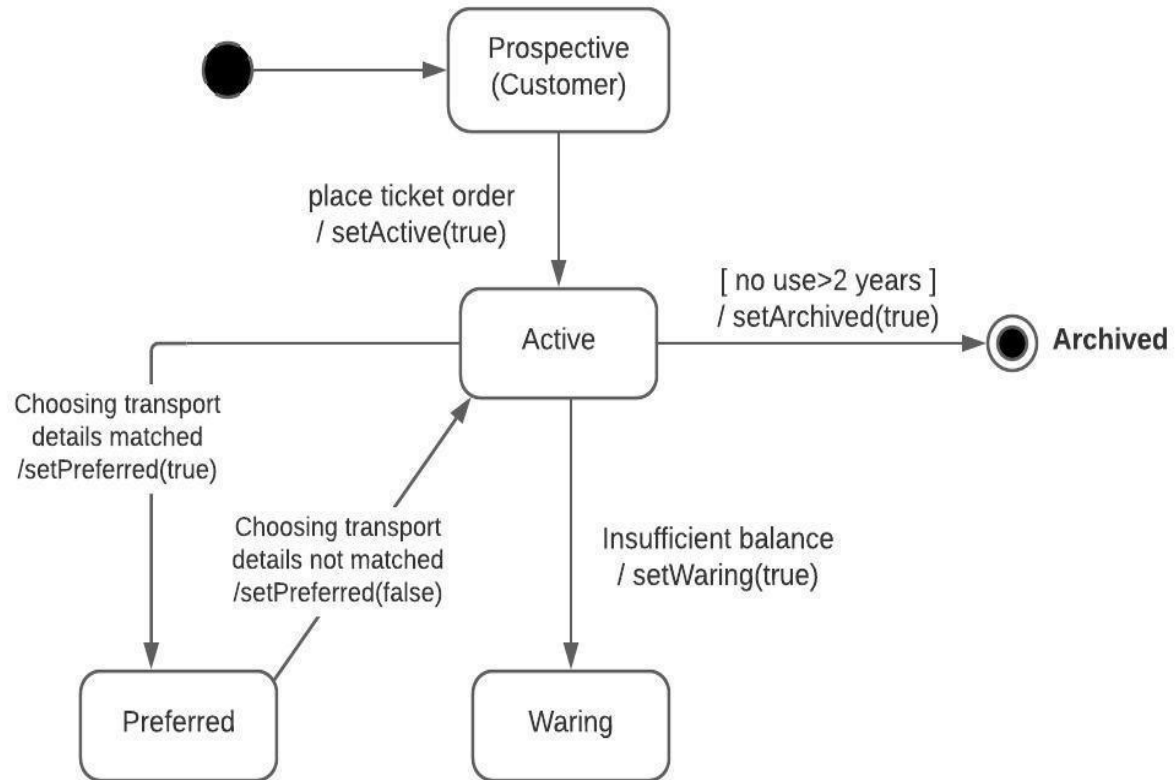
### c. Statechart Diagram

Statechart Diagram - 01 : Ticket





## Statechart Diagram - 02 : Customer



#### d. Calculation of the LCOM value with comment

Class - 01 : Customer

Customer
contact_method {email or mobile number} user_password credit_card_Details {valid} time date departure_place arrival_place transport_preference number_of_seats
CreateAccount(contact_method, user_password, credit_card_Details)  Login(contact_method, user_password)  CreateRequestForTicket(contact_method, time, date, departure_place, arrival_place, transport_preference, number_of_seats)  MakePayment(user_password, credit_card_Details)  CancelRequest(transport_preference, user_password)  Logout( )

Let,

CreateAccount(contact\_method, user\_password, credit\_cardDetails) = 01

Login(contact\_method, user\_password) = 02

CreateRequestForTicket(contact\_method, time, date, departure\_place, arrival\_place, transport\_preference, number\_of\_seats) = 03

MakePayment(user\_password, credit\_cardDetails) = 04

CancelRequest(transport\_preference, user\_password) =05

Logout( ) = 06

Pairs:

(O1, O2), (O1, O3), (O1, O4), (O1, O5), (O1, O6), (O2, O3), (O2, O4), (O2, O5), (O2, O6), (O3, O4), (O3, O5), (O3, O6), (O4, O5), (O4, O6), (O5, O6)

P = 6 (Non-Cohesive pairs)

Q = 9 (Cohesive pairs)

We know that,

$LCOM = |P| - |Q|$ , if  $|P| > |Q|$ , otherwise 0

Here,  $Q > P$

So,  $LCOM = 0$

*Comment:* The LCOM value of the class indicates that the methods of the class are cohesive, and it is a desirable design.

Class - 02 :      System

System
user_contact_method {email or mobile number} OTP user_credit_card_Details time date departure_place arrival_place transport_preference transport no number_of_seats
VerifyAccount(user_contact_method, OTP, user_credit_card_Details)  VerifyLogin(user_contact_method, user_password)  CheckRequestTicket(user_contact_method, time, date, departure_place, arrival_place, transport_preference, number_of_seats)  CheckAvailability(time, date, departure_place, arrival_place, transport_preference, number_of_seats, transport no)  CheckBalance(user_credit_card_Details)  CancelRequest(transport_preference, user_credit_card_Details, user_contact_method)  ArchivedAccount( )

Let,

VerifyAccount(user\_contact\_method, OTP, user\_credit\_card\_Details) = 01

VerifyLogin(user\_contact\_method, user\_password) = 02

CheckRequestTicket(user\_contact\_method, time, date, departure\_place, arrival\_place,  
transport\_preference, number\_of\_seats) = 03

CheckAvailability(time, date, departure\_place, arrival\_place, transport\_preference, number\_of\_seats, transport no) = 04

CheckBalance(user\_credit\_card\_Details) = 05

CancelRequest(transport\_preference, user\_credit\_card\_Details, user\_contact\_method) = 06

ArchivedAccount( ) = 07

Pairs:

(O1, O2), (O1, O3), (O1, O4), (O1, O5), (O1, O6), (O1, O7), (O2, O3), (O2, O4), (O2, O5),  
(O2, O6), (O2, O7), (O3, O4), (O3, O5), (O3, O6), (O3, O7), (O4, O5), (O4, O6), (O4, O7),  
(O5, O6), (O5, O7), (O6, O7)

P = 11 (Non-Cohesive pairs)

Q = 10 (Cohesive pairs)

We know that,

$LCOM = |P| - |Q|$ , if  $|P| > |Q|$ , otherwise 0

Here,  $P > Q$

So,  $LCOM = |11| - |10|$

= 1

**Comment:** The LCOM value of the class indicates that the methods of the class are non-cohesive. But as the LCOM value is while quite low, it might be considered as a desirable design.