



Dharmsinh Desai University, Nadiad

Faculty of Technology

Department of Computer Engineering

B. Tech. CE Semester – IV

SUBJECT: Software Engineering

**PROJECT TITLE:**  
**TOUR AND TRAVEL MANAGEMENT SYSTEM**

Guided By: Prof. Pinkal Chavda  
Assistant Professor  
Dept. of Computer Engg.

Made By:

- 1) NIKHIL BAMBHROLIYA, ROLL NO.: CE009,  
ID:19CEUEG099
- 2) DHAIRYA BILANDANI, ROLL NO.: CE013,  
ID:19CEUON065



**Faculty of Technology  
Department of Computer Engineering  
Dharmsinh Desai University  
CERTIFICATE**

This is to certify that the practical / term work carried out in the subject of  
Software Engineering and recorded in this journal is the  
bona fide work of

NIKHIL BAMBHROLIYA (CE-009) (19CEUEG099)  
DHAIRYA BILANDANI (CE-013) (19CEUON065)

of B. Tech semester IV in the branch of Computer Engineering  
during the academic year 2020-2021.

Prof. Pinkal C. Chavda  
Assistant Professor,  
Dept. of Computer Engg.  
Faculty of Technology  
Dharmsinh Desai University, Nadiad

Dr. C. K. Bhensdadia,  
Head,  
Dept. of Computer Engg.,  
Faculty of Technology  
Dharmsinh Desai University, Nadiad

1. Abstract.....	4
2. Introduction.....	5
3. Software Requirement Specifications.....	7
4.Design.....	9
5. Implementation Detail.....	16
6.Testing.....	21
7. Screenshots.....	23
8.Conclusion.....	26
9. Limitation and Future Extension.....	27
10.Bibliography.....	28

## **1 ABSTRACT**

In modern day society no one has time to stand queue and wait for hours to book a single ticket or bunch of them. Here time has become one of the important aspect of time.

To make travel more comfortable and better their is need for system to booking tickets before hand and easily accessible at any point of time or in emergencies. It also helps to get better offers and best seats to book at low prices.

## **2. INTRODUCTION**

## **2.1 INTRODUCTION TO TOURS AND TRAVEL MANAGEMENT SYSTEM**

Now a days no one has time to stand in queues and spend their precious time to buy a tickets. Even after standing in line for a long time they are not able to get ticket of their favourable seat. Because of that their journey becomes difficult and very tiresome which spoils the trip.

But their is a system that helps to book ticket in no time and help you to get favourable seats with best offers and cheaper prices.

### **FEATURES OF TOURS AND TRAVEL MANAGEMENT SYSTEM**

- **USER CAN CREATE HIS ACCOUNT ON THE SYSTEM.**
- **AFTER CREATING ACCOUNT HE CAN LOGIN THE SYSTEM.**
- **USER CAN SEARCH FOR FLIGHTS, TRAINS AND BUSES.**
- **AFTER THAT HE CAN BOOK TICKET OF BUS, TRAIN OR PLANE.**
- **PAYMENTS CAN BE DONE AFTER BOOKING OF TICKETS.**
- **IF USER WANT TO CANCEL THE BOOKING THAN HE CAN.**

## **2.2 Technology/Platform/Tools Used**

### **VISUAL STUDIO:**

Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.

**PYTHON:**

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation.

**DJANGO:**

DJANGO is a Python-based free and open-source web framework that follows the model-template-views (MTV) architectural pattern. Django's primary goal is to ease the creation of complex, database-driven websites.

**MYSQL:**

MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

### **3. Software Requirement Specifications**

**TOURS AND TRAVEL MANAGEMENT SYSTEM:-**

Managing user profile:-

Register user profile

Description: User can register themselves.  
Input: Basic personal details.  
Output: Accounted has been created.

#### Login user profile

Description: User can enter the system.  
Input: Enter username and password.  
Output: Enter into the system.

#### Logout user profile

Description: User can logout from the system  
Input: Click on logout.  
Output: Out of the system.

### Booking Tickets-:

#### Selection of mode of transport

Description: Choose the mode of transport.  
Input: Select the mode of transport.  
Output: Move to page Search page.

#### Searching for particular mode of transport

Description: search for flight, train, bus for particular date and time.  
Input: Enter date and time.  
Output: Details along with time is displayed.

#### Booking for particular ticket

Description: Enter details and get the ticket booked.

Input: Enter personal details

Output: Ticket is booked.

Payment of ticket which is booked

Description: Payment of ticket which is booked.

Input: Enter payment Details.

Output: Receipt of booking with booking details.

Cancellation of ticket

Description: Cancel the ticket which is booked.

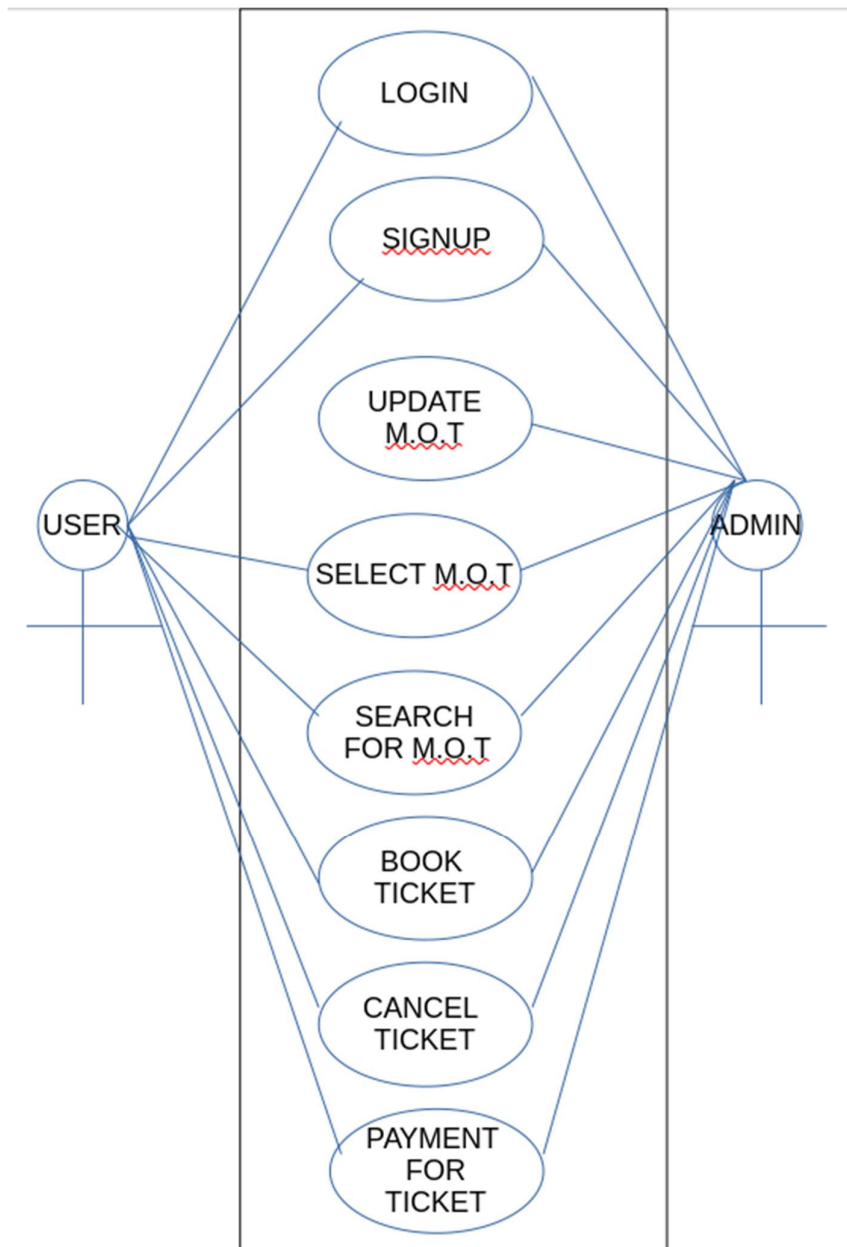
Input: Cancel the ticket.

Output: Cancellation confirmation.

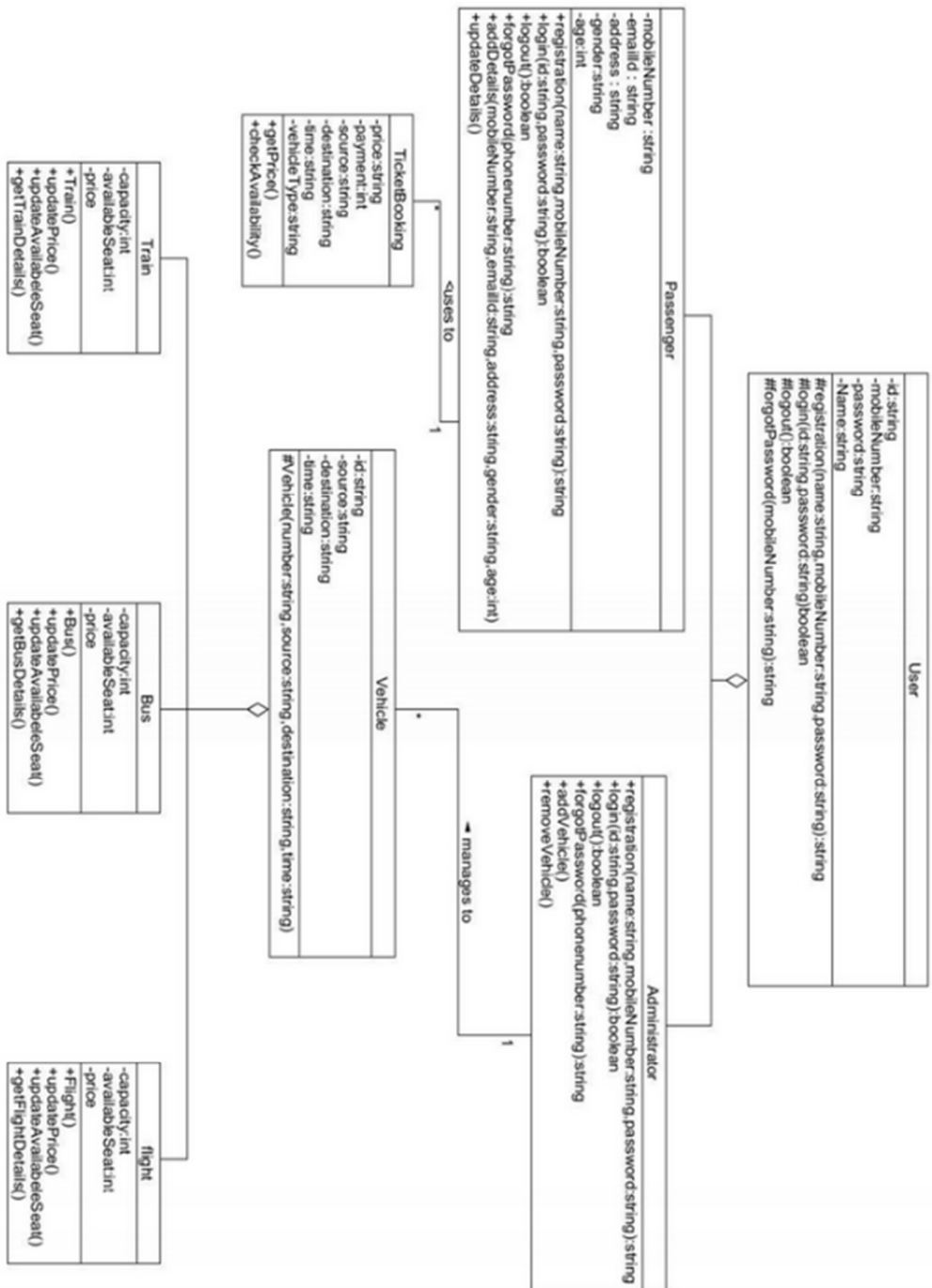
## **4 DESIGNS**

### **4.1 Usecase Diagrams:-**



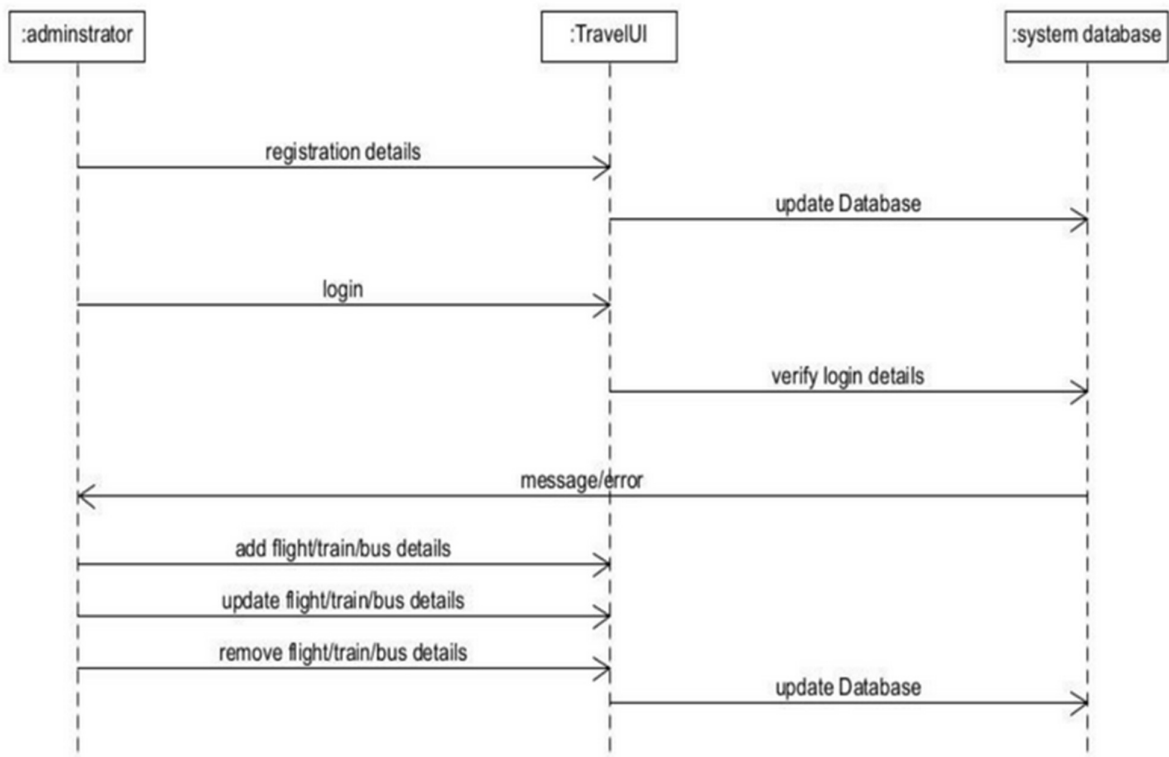


## 4.2 Class Diagrams:-

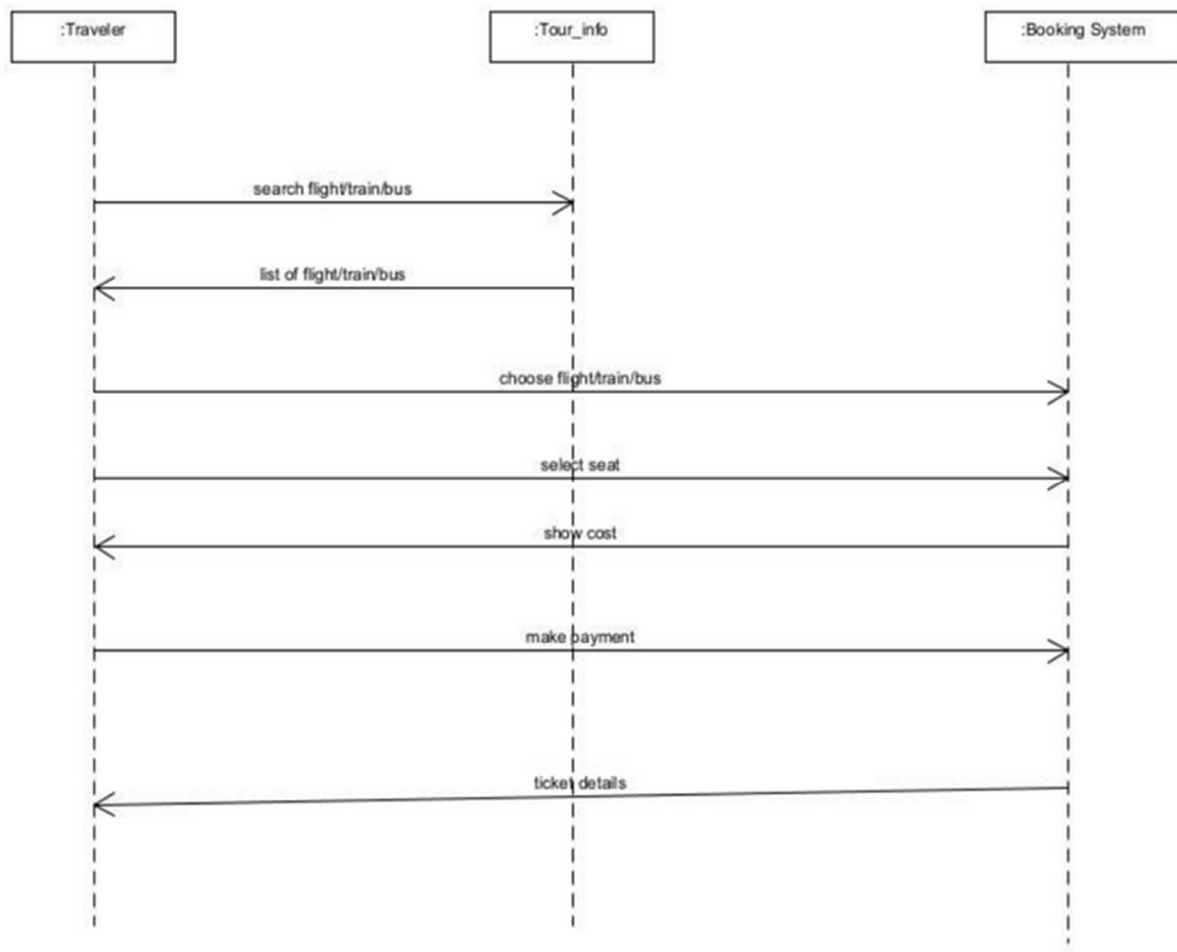


## 4.3 SEQUENCE DIAGRAM

### Sequence diagram of login/registration

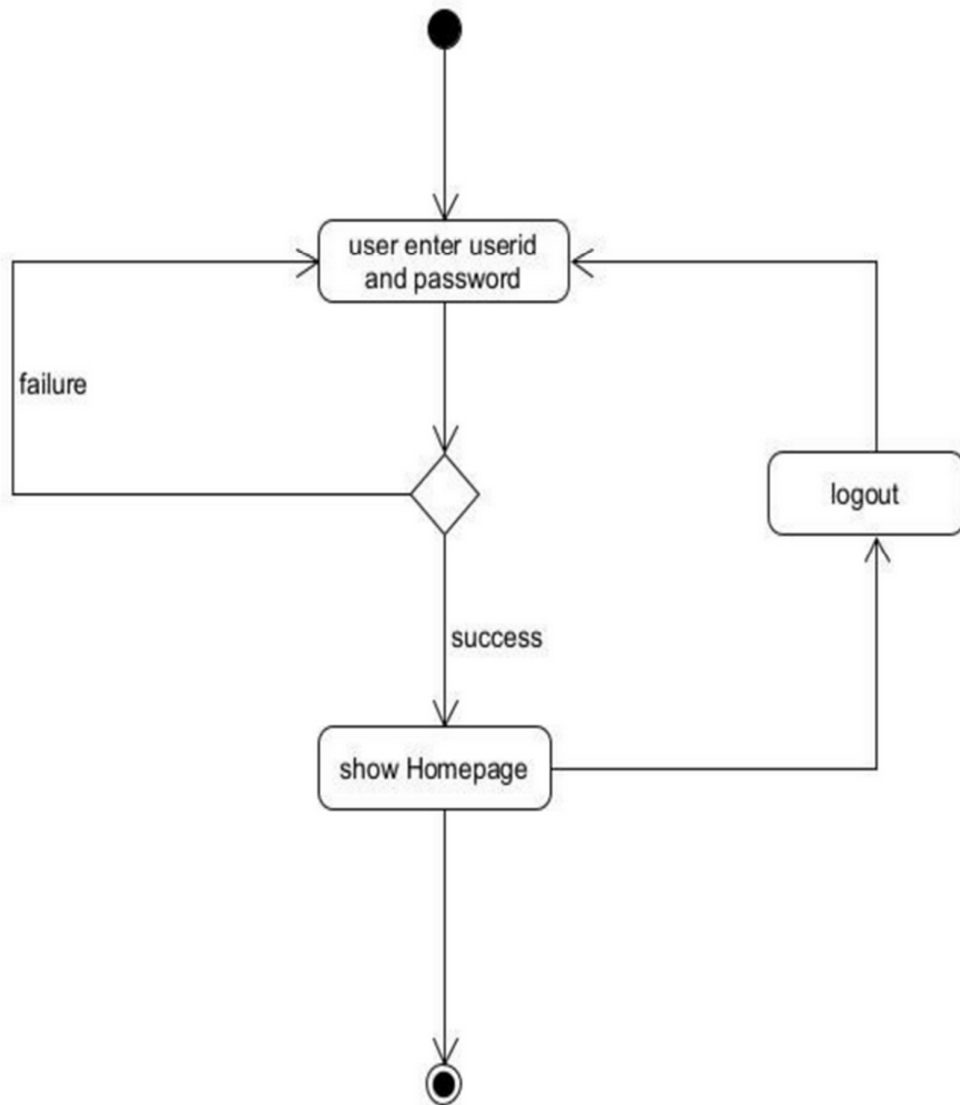


## Sequence diagram of Ticket Reservation

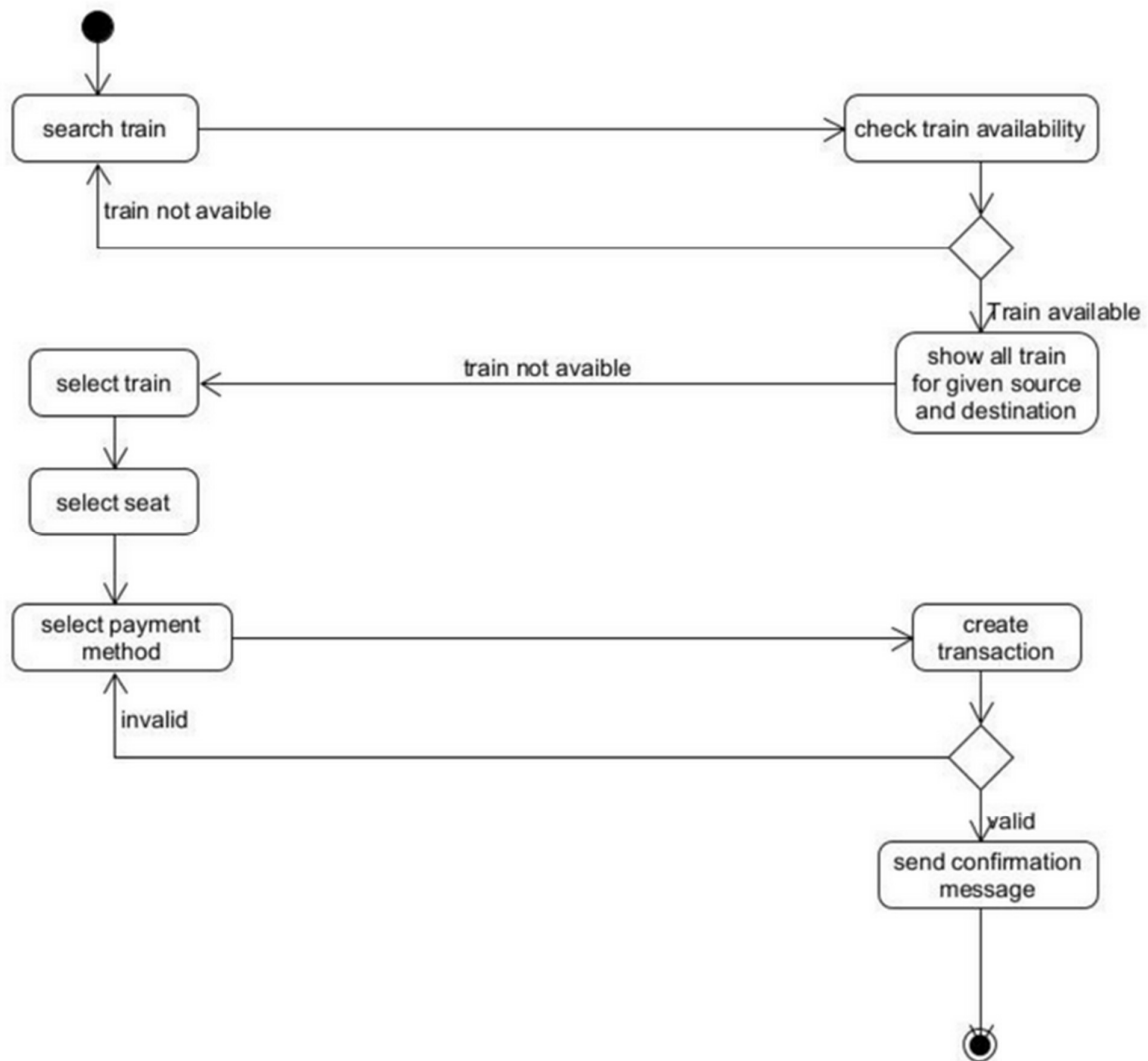


#### 4.4 ACTIVITY DIAGRAM:-

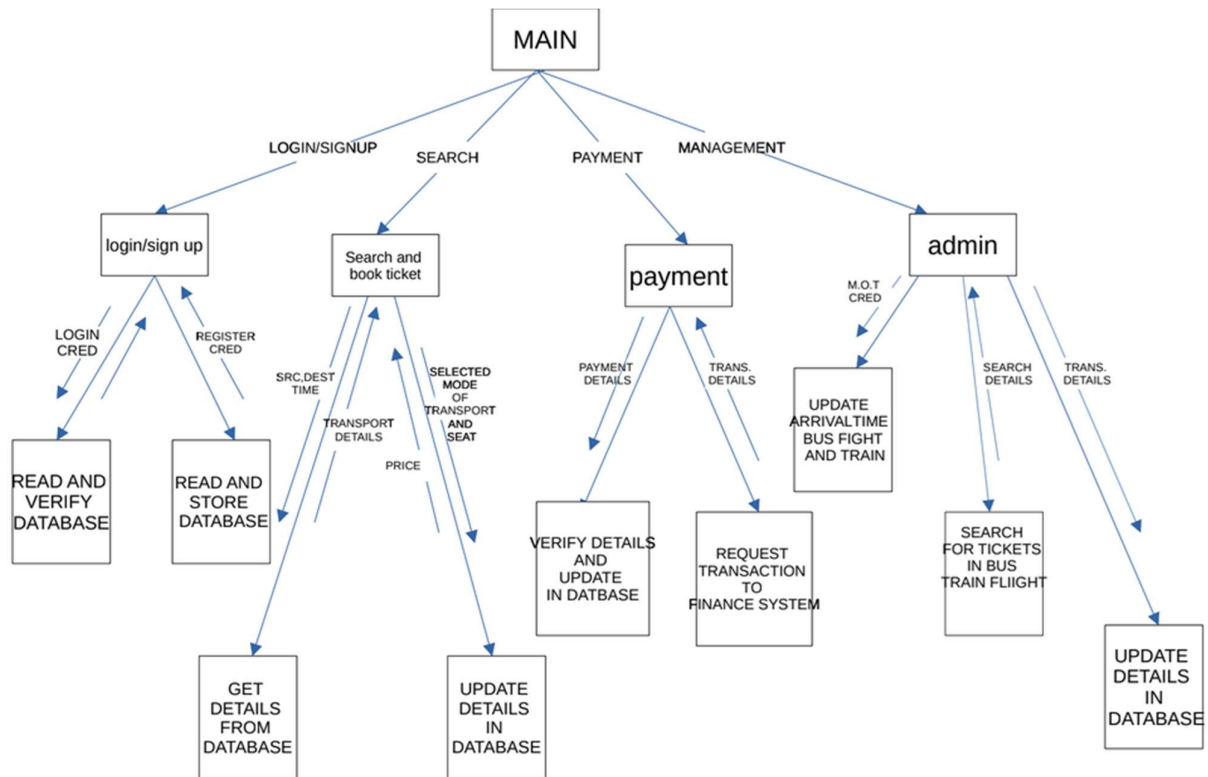
##### Activity Diagram of Login



## Activity Diagram of Ticket Reservation



## 4.5 STRUCTURE CHART:-



## 5. Implementation Details

## 5.1 Description of Modules :-

### 1. Register User Module

Basic information of user is taken by system and stored in database.

### 2. Login User Module

Users are able to login themselves. System logs user in, then and only according to the user can use other functionalities of system.

### 3. Sign out User Module

Users can log out themselves from the system.

### 4. Selection Module

User can select mode of transport of his own convenience

### 5. Search Module

Users can search for certain mode of transport according to his convenient time and date.

### 6. Reservation Module

Users can book the tickets for the place they want to visit. They can book instantly by giving some personal details.

Signup Module



```

144
145 def signup(request):
146     if (request.method == 'GET'):
147         form = UserCreationForm()
148         context = {
149             "form" : form
150         }
151
152         return render(request,'signup.html',context=context)
153     else :
154         form = UserCreationForm(request.POST)
155         context = {
156             "form" : form
157         }
158
159         if form.is_valid() :
160             form.save()
161             return redirect('home')
162
163         else :
164             return render(request,'signup.html',context=context)
165

```

## Login Module

```

123 def login(request):
124     if(request.method == 'GET'):
125         form = AuthenticationForm()
126         context = {
127             'form' : form
128         }
129         return render(request,'login.html',context=context)
130     else:
131         form = AuthenticationForm(data=request.POST)
132         context = {
133             'form' : form
134         }
135
136         if form.is_valid():
137             username = form.cleaned_data.get('username')
138             password = form.cleaned_data.get('password')
139             user = authenticate(username=username,password=password)
140             loginUser(request,user)
141             return redirect('home')
142         else :
143             return render(request,'login.html',context=context)

```

## Signout Module

```

@login_required(login_url='login')
def signout(request):
    logout(request)
    return redirect('home')

def index(request):
    return render(request, 'index.html')

def contact(request):
    return render(request, 'contact.html')

def about(request):
    return render(request, 'about.html')

```

## Selection Module

```

9  def buss(request):
10     bus = BUS.get_all_buss()
11     request.session['mot']="Bus"
12     return render(request, 'showBuss.html', {'buss':bus})
13
14  def flights(request):
15     flight = FLIGHT.get_all_flights()
16     request.session['mot']="Flight"
17     return render(request, 'showFlights.html', {'flights':flight})
18
19  def trains (request):
20     train = TRAIN.get_all_trains()
21     request.session['mot']="Train"
22     return render(request, 'showtrains.html', {'trains':train})
23

```

## Search Module

```
def search(request):
    mode_of_transport=request.session['mot']
    source = request.GET['source']
    destination = request.GET['destination']
    if mode_of_transport == 'Bus':
        buss = BUS.objects.filter(source__icontains=source,destination__icontains=destination)
        return render(request,'showbuss.html',{'buss':buss})

    if mode_of_transport == 'Flight':
        flights = FLIGHT.objects.filter(source__icontains=source,destination__icontains=destination)
        return render(request,'showflights.html',{'flights':flights})

    if mode_of_transport == 'Train':
        trains = TRAIN.objects.filter(source__icontains=source,destination__icontains=destination)
        return render(request,'showtrains.html',{'trains':trains})
```

## Reservation Module

1)

```
@login_required(login_url='login')
def reservation(request,**kwargs):
    if (request.method == 'GET'):
        form = BOOKINGForm()
        context = {'form':form , 'id' : kwargs['id']}
        return render(request,'booking.html',context=context)
    else:
        user=request.user
        id=kwargs['id']
        form = BOOKINGForm(request.POST)

        if form.is_valid:
            book=form
            book.user=user
            if request.session['mot']=="Flight":
                flight = FLIGHT.objects.get(pk=id)
                source = flight.source
                destination=flight.destination
                price=flight.price

            if request.session['mot']=="Train":
                train = TRAIN.objects.get(pk=id)
                source = train.source
                destination=train.destination
                price=train.price

            if request.session['mot']=="Bus":
                bus = BUS.objects.get(pk=id)
                source = bus.source
                destination=bus.destination
                price=bus.price
```

2)

```
name = request.POST.get('name')
age = request.POST.get('age')
phone = request.POST.get('phone')
email = request.POST.get('email')
seat = request.POST.get('seat')
total_cost = price * seat

b = BOOKING(
    user=user,
    name=name,
    age=age,
    phone=phone,
    email=email,
    seat=seat,
    total_cost=total_cost,
    source=source,
    destination=destination,
    price=price,
    mode_of_transport=request.session['mot']
)
global fun
def fun():
    return b

return redirect('payment')

else:
    context = {'form':form , 'id':id}
    return render(request,'booking.html',context=context)
```

## 6 TESTING

Manual testing was performed in order to find and fix the bugs in development process.

### Testing Method: Manual Testing

<b>Sr. No.</b>	<b>Test Scenario</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status</b>
1	Sign-up with invalid credentials	Not able to create an account	Not able to create an account	Success
2	Sign-up with Correct credentials	Account Should be Created	Account was created	Success
3	Login with incorrect credentials	Invalid Credentials, please try again	Invalid Credentials, please try again	Success
4	Login with correct credentials	You are Successfully logged in	You are Successfully logged in	Success
5	Choosing 2 Mode of Transport at once	Error	Error	Success
6	Searching invalid Flights, Bus, Train	It will not give results	It will not give results	Success

7	Searching valid Train, Flight and bus	It will give result along with its time	It will give result along with its time	Success
8	Filling incorrect details	It will not book the ticket	It will not book the ticket	Success
9	Filling correct details	Ticket will be booked	Ticket will be booked	Success

## Code :

```

app > tests.py
1  from django.test import TestCase
2  from selenium import webdriver
3  from django.contrib.staticfiles.testing import StaticLiveServerTestCase
4  from django.urls import reverse
5  import time
6
7  # Create your tests here.
8
9  class TestProjectListPage(StaticLiveServerTestCase):
10
11     def setUp(self):
12         self.browser=webdriver.Chrome('chromedriver.exe')
13
14     def tearDown(self):
15         self.browser.close()
16
17     def Test(self):
18         self.browser.get('http://127.0.0.1:8000/app/login/')
19         self.browser.maximize_window()
20         self.browser.find_element_by_name("username").send_keys("nikhil")
21         time.sleep(1)
22         self.browser.find_element_by_name("password").send_keys("nikhil")
23         time.sleep(1)
24         self.browser.find_element_by_class_name("btn").click()
25         time.sleep(5)
26
27         self.browser.find_element_by_class_name("bus").click()
28
29         self.browser.find_element_by_name("source").send_keys("nadiad")
30         time.sleep(1)
31         self.browser.find_element_by_name("destination").send_keys("rajkot")
32         time.sleep(3)
33         self.browser.find_element_by_class_name("btn").click()
34         time.sleep(3)
35         link = self.browser.find_element_by_link_text('Reserve')
36         link.click()

```

```

28
29     self.browser.find_element_by_name("source").send_keys("nadiad")
30     time.sleep(1)
31     self.browser.find_element_by_name("destination").send_keys("rajkot")
32     time.sleep(3)
33     self.browser.find_element_by_class_name("btn").click()
34     time.sleep(3)
35     link = self.browser.find_element_by_link_text('Reserve')
36     link.click()
37
38     time.sleep(2)
39     self.browser.find_element_by_name("name").send_keys("nikhil")
40     time.sleep(1)
41     self.browser.find_element_by_name("age").send_keys("2")
42     time.sleep(1)
43     self.browser.find_element_by_name("phone").send_keys("1212121212")
44     time.sleep(1)
45     self.browser.find_element_by_name("email").send_keys("nnn@gmail.com")
46     time.sleep(1)
47     self.browser.find_element_by_name("seat").send_keys("1")
48     time.sleep(5)
49     self.browser.find_element_by_class_name("btn").click()
50
51     self.browser.find_element_by_name("card_num").send_keys("451212121245")
52     time.sleep(1)
53     self.browser.find_element_by_name("month").send_keys("january")
54     time.sleep(1)
55     self.browser.find_element_by_name("year").send_keys("2023")
56     time.sleep(1)
57     self.browser.find_element_by_name("cvv").send_keys("123")
58     time.sleep(5)
59
60     self.browser.find_element_by_name("submit").click()
61     time.sleep(3)

```

```

60     self.browser.find_element_by_name("submit").click()
61     time.sleep(3)
62
63     self.browser.find_element_by_link_text("Home").click()
64     time.sleep(3)
65
66     self.browser.find_element_by_link_text("My Bookings").click()
67     time.sleep(3)
68
69     self.browser.find_element_by_link_text("Cancel").click()
70     time.sleep(3)
71

```

## 7 Screenshots

### Create Account

Username:  Required. 100 characters or fewer. Letters, digits and @/./+/-/\_ only.

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:  Enter the same password as before, for verification.

Create Account

## SIGN UP

### Login Here

Username:

Password:

Login

## LOGIN



search here

Search

Flights

#	source	destination	time	price	available seat
1	ahmedabad	nadiad	6:17 a.m.	10000	15

Reserve

## SEARCH FOR FLIGHT

Reservation

Name:

Age:

0

Phone:

Email:

Seat:

0

Submit

## BOOKING TICKET

Payment	
Card Number	12 digit number
Expiry Date	Month <input type="text"/> Year <input type="text"/>
CVV	***
<input type="button" value="Make Payment"/>	

## PAYMENT

FIRST NAME
<input type="text" value="Your name.."/>
LAST NAME
<input type="text" value="Your last name.."/>
CONTACT NUMBER
<input type="text" value="CONTACT NUMBER.."/>
Country
<input type="text" value="BEST"/>
EXPERIENCE
<input type="text" value="Write something.."/>
<input type="button" value="Submit"/>

## FEEDBACK FORM

## 6 Conclusion

The functionalities are implemented in system after understanding all the system modules according to the requirements.

Functionalities that are successfully implemented in the system are:

- HOMEPAGE
- SIGN UP
- LOGIN
- SELECTING MODE OF TRANSPORT
- SEARCHING FOR PARTICULAR MODE OF TRANSPORT
- RESERVATION PAGE
- PAYMENT OF TICKET BOOKED
- PRINT THE TICKET WHICH IS BOOKED
- FEEDBACK AND CONTACT PAGE
- SIGN OUT

After the implementation and coding of system, comprehensive testing was performed on the system to determine the errors and possible flaws in the system.

## 7 Limitations and Future Enhancements

We were able to make initial but effective software . It was not up to the mark because of many functionality we have added do not have effective verification and better features.

We also want to see our software performing better by adding some features like routes through which bus and train travel. We want to make our payment system more secure.

About how much time Train, Bus or Flight is running late and addition features like adding lunch or dinner to your ticket.

We also are planning to expand our businesses by collaborating with hotel and restaurants for stay. Also looking forward to add feature where our software shows the best places to visit in well known cities.

We want to also apply AI to software so that our customers can ask for initial help or doubts directly to our BOT.

## 8 Bibliography

Following links and websites were referred during the development of this project:

- [stackoverflow.com](https://stackoverflow.com)
- [github.com](https://github.com)
- [docs.djangoproject.com](https://docs.djangoproject.com)
- [youtube.com](https://youtube.com)
- [w3school.com](https://w3school.com)
- [bootstrap.com](https://bootstrap.com)
- [google.com](https://google.com)

Project Git Repository link:

<https://github.com/nikhibpatel/online-tourism-and-travels>