**E-TICKET RESERVATION SYSTEM**

MINOR PROJECT REPORT

Submitted in partial fulfilment of the requirements for the award of the degree

Of

**BACHELOR OF COMPUTER APPLICATION**

By

**BANTI JHA**

**ENROLLMENT NO.: 02090202020**

Guided by

**Ms. Charanpreet Kaur**

**Asst. Professor**



**Sri Guru Tegh Bahadur Institute of Management and Information Technology, DELHI – 110033**

(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI)

January, 2023

**DECLARATION**

I hereby declare that the work, which is being presented in this project entitled “**E-TICKET RESERVATION SYSTEM”**, is an authentic record of my own work carried out during a period from SEPTEMBER 2022 To JANUARY 2023 under the supervision and guidance of Ms., Charanpreet Kaur (Asst. Professor). This project report was undertaken as a part of the Minor Project as per the curriculum of GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI for the partial fulfillment of BCA from Sri Guru Tegh Bahadur Institute of Management and Information Technology.

The matter embodied here in this project has not been submitted by me for the award of any other Degree/Diploma.

BANTI JHA

02090202020

**ACKNOWLEDGEMENT**

Firstly, I would like to thank Ms. Charanpreet Kaur our mentor with her patience and immense knowledge and experience, she helped us get out of the problems we faced during the project completion. It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of my fellow team mates and our mentor. I am highly indebted to my Mentor Ms. Charanpreet Kaur for her supervision and guidance and for her constructive criticism throughout our project span. I am extremely grateful to my department staff members, my teammates and friends who helped me in successful completion of this project.

BANTI JHA

02090202020

**ABSTRACT**

We built our **E-TICKET RESERVATION SYSTEM** with the motive of providing an easy, convenient, secure, user-friendly and hassle free solution to the customers for booking tickets from the ease of their homes.

This project presents a review on the software program “E-TICKET RESERVATION SYSTEM” as should be used in a bus transportation system, a facility which is used to reserve seats, cancellation of reservation and different types of route enquiries used on securing quick reservations. **E-TICKET RESERVATION SYSTEM** is built for managing and computerizing the traditional database, ticket booking and tracking bus and travel made. It maintains all customer details, bus details, reservation details.

An electronic ticket, more efficient method of ticket entry, processing and marketing for companies in the airline, railways and other transport and entertainment industries. Many sports, concert venues, and cinemas use electronic ticketing for their events. Electronic tickets, or "E-Tickets" as they are sometimes referred, are often delivered as PDFs or another downloadable format that can be received via email or through a mobile app. Electronic tickets allow spectators to download their tickets, as opposed to waiting for physical tickets to arrive in the mail. A printed copy of these tickets or a digital copy on a mobile phone should be presented on coming to the venue.

The application offers many advantages for both travelers and Bus Agencies, including security, flexibility, cost and convenience. At the same time, it also provides the standard assurances of the traditional paper ticket, such as seating choice, travel time options and other flexibilities.

Agencies can more easily track down passengers to inform them of itinerary adjustments, cancellations and other last-minute changes. Bus Agencies and travel agencies also are plugging into the growing numbers of devices travelers use to communicate and manage their lives. E-ticket passengers can receive everything from gate assignments to cancellation or delays through e-mail and [text messaging](https://computer.howstuffworks.com/e-mail-messaging/sms.htm) sent to their home or business computers or to their cell phones, [personal data assistants](https://electronics.howstuffworks.com/gadgets/travel/pda.htm) (such as a Drivers or Conductors).

We demonstrate the efficiency and effectiveness of our approach through an experimental evaluation using our implemented prototype.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| S.NO. | TITLE | PG.NO. |
|  | List of Tables |  |
|  | List of Figures |  |
|  | Introduction |  |
|  | Requirement and Analysis   * 1. SDLC   2. Software Requirement Specification   3. Use Case Diagram |  |
|  | Software Design  3.1. DFD-0  3.2. DFD-1  3.3. DFD-2 |  |
|  | Database Design  4.1 ER Diagram  4.2 Tables |  |
|  | Testing   * 1. What is Software Testing?   2. Types of Software Testing   3. Test Cases |  |
|  | Roles and Responsibility |  |
|  | Conclusion And Future Enhancement |  |
|  | Appendices (Coding Snippets) |  |
|  | References |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| S.NO. | TABLE NAME | PG.NO. |
|  | BUS TABLE |  |
|  | PASSENGER TABLE |  |
|  | ADMIN DETAILS |  |
|  | TEST CASES |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| FIG.NO. | FIG. NAME | PG.NO. |
|  | ITERATIVE MODEL |  |
|  | ATTRIBUTES OF SOFTWARE |  |
|  | LEVELS OF TESTING |  |
|  | TYPES OF SOFTWARE TESTING |  |

**CHAPTER-1**

**INTRODUCTION**

An electronic ticket, more efficient method of ticket entry, processing and marketing for companies in the airline, railways and other transport and entertainment industries. Many sports, concert venues, and cinemas use electronic ticketing for their events. Electronic tickets, or "E-Tickets" as they are sometimes referred, are often delivered as PDFs or another downloadable format that can be received via email or through a mobile app. Electronic tickets allow spectators to download their tickets, as opposed to waiting for physical tickets to arrive in the mail. A printed copy of these tickets or a digital copy on a mobile phone should be presented on coming to the venue.

This project presents a review on the software program “E-TICKET RESERVATION SYSTEM” as should be used in a bus transportation system, a facility which is used to reserve seats, cancellation of reservation and different types of route enquiries used on securing quick reservations. E-TICKET RESERVATION SYSTEM is built for managing and computerizing the traditional database, ticket booking and tracking bus and travel made. It maintains all customer details, bus details, reservation details.

Bus scheduling and booking system is a software application which is connected to a complete database. The database includes information about buses, no. of seats available, occupancy, availability, days and time of operation, no. of buses from point ‘A’ to point ‘B’, Price ranges, automated report and bill generation etc.

The database is updated time to time and a user can reserve his ticket from any part of the world and any time.

The application gives utmost importance to security and usability.

The application offers many advantages for both travelers and Bus Agencies, including security, flexibility, cost and convenience. At the same time, it also provides the standard assurances of the traditional paper ticket, such as seating choice, travel time options and other flexibilities.

Unlike the traveler who leaves his ticket at the office, e-tickets are impossible to "lose" because they reside in a computer database network. For this reason, they are hard to steal, as well. Passengers typically print out copies of their e-ticket, including confirmation [e-mails](https://computer.howstuffworks.com/e-mail-messaging/email.htm), itineraries and other documents. All those documents can be replaced by pulling them out of the computer again, and only a person with the proper identification can actually use the e-ticket. With the old paper tickets, passengers who lost or forgot them might be charged a fee for the airline to make new ones. In some cases, passengers were required to buy new tickets at full-price. E-tickets offer a distinct advantage in this area.

A passenger also may find it easier to make changes to their travel itinerary using an e-ticket, as the travel agency or airline need only update their database with the requested changes rather than incur the expense of physically issuing a new ticket. Dealing with e-tickets is much less costly to Bus Agencies -- the industry estimates a savings of 45 Lakhs annually if it used e-tickets only -- which makes it possible for Bus Agencies to offer more competitive fares to passengers. The tickets also are booked and processed in a more timely way, saving labor hours and cutting down on traveler frustration.

After the sale, Bus Agencies can more easily track down passengers to inform them of itinerary adjustments, cancellations and other last-minute changes. Bus Agencies and travel agencies also are plugging into the growing numbers of devices travelers use to communicate and manage their lives. E-ticket passengers can receive everything from gate assignments to cancellation or delays through e-mail and [text messaging](https://computer.howstuffworks.com/e-mail-messaging/sms.htm) sent to their home or business computers or to their cell phones, [personal data assistants](https://electronics.howstuffworks.com/gadgets/travel/pda.htm) (such as a conductors). For harried business travelers, this can be especially convenient, as it provides real-time updates on their travel arrangements without slowing them down. Bus Agencies and travel agencies also can use the system to alert travelers to special discounts or promotions, possibly saving them money. E-tickets, in short, allow Bus Agencies and travel agencies to use the strengths of today's e-culture, both for their own and their passengers' benefit.

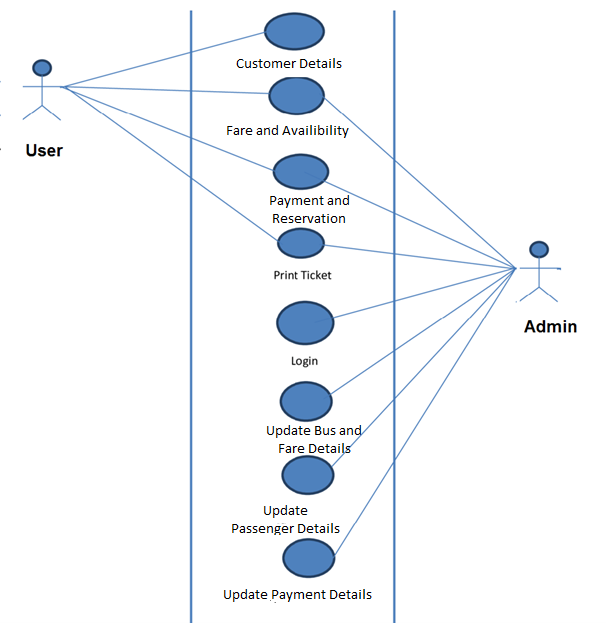
Travel agencies save on the cost of maintaining ticket printers and ticket inventory control. Using an e-ticket often gets the passenger through the gate and on the plane quicker and with less hassle than a paper ticket as Bus Agencies encourage passengers to use the e-tickets.

Finally, using e-tickets is a more environmentally friendly approach.

**CHAPTER-2**

**REQUIREMENT ANALYSIS**

* 1. **SOFTWARE REQUIREMENT SPECIFICATION**
* **SOFTWARE REQUIREMENTS**
* Operating System: Windows 10/11, Linux , macOS
* Programming Language: Python 3.9.10
* Database: MySQL
* **HARDWARE REQUIREMENTS**
* Device: PC/Laptop 32/64 Bit System
* Storage Size: 8GB
* Ram: 4GB
* Space On Disc: 2GB
* Processor: Intel core 3/4/5,RYZEN-3
  1. **USE CASE DIAGRAM**

****

**CHAPTER-3**

**SOFTWARE DESIGN**

Software design is a mechanism to transform user requirements into some suitable form, which helps the programmer in software coding and implementation. It deals with representing the client's requirement, as described in SRS (Software Requirement Specification) document, into a form, i.e., easily implementable using programming language.

**Data Flow Diagrams**

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.



The Data flow diagram can be explained as the separate levels indicating the individual’s complexity in the each level of the system and gives a detailed explanation in the further levels that are following them.

**LEVEL 0**

It is also known as a context diagram .It’s designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

**LEVEL 1**

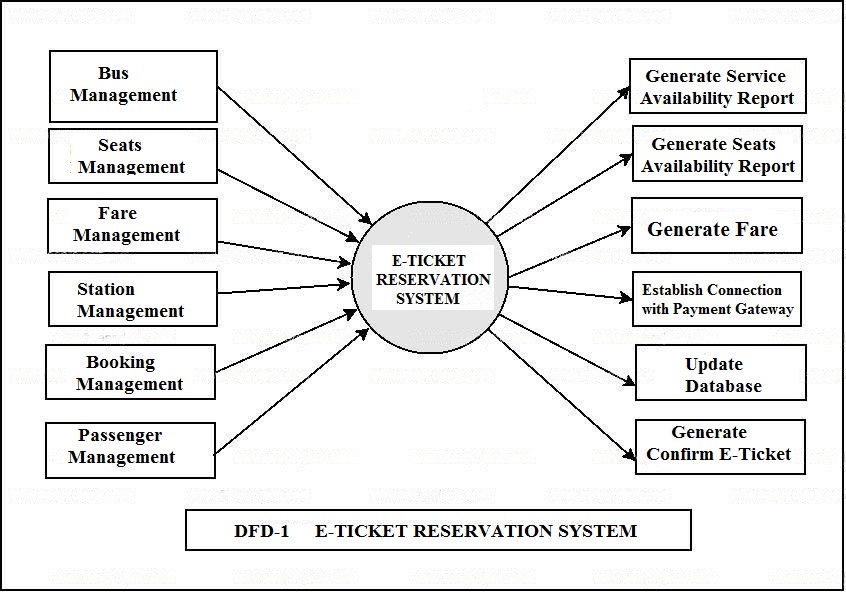
In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into sub-processes. The level 1 of the Data Flow Diagram gives explain in detail about packet watching system which was marked under as 0 in the previous level.

* 1. **CONTEXT DIAGRAM (0-LEVEL DFD)**

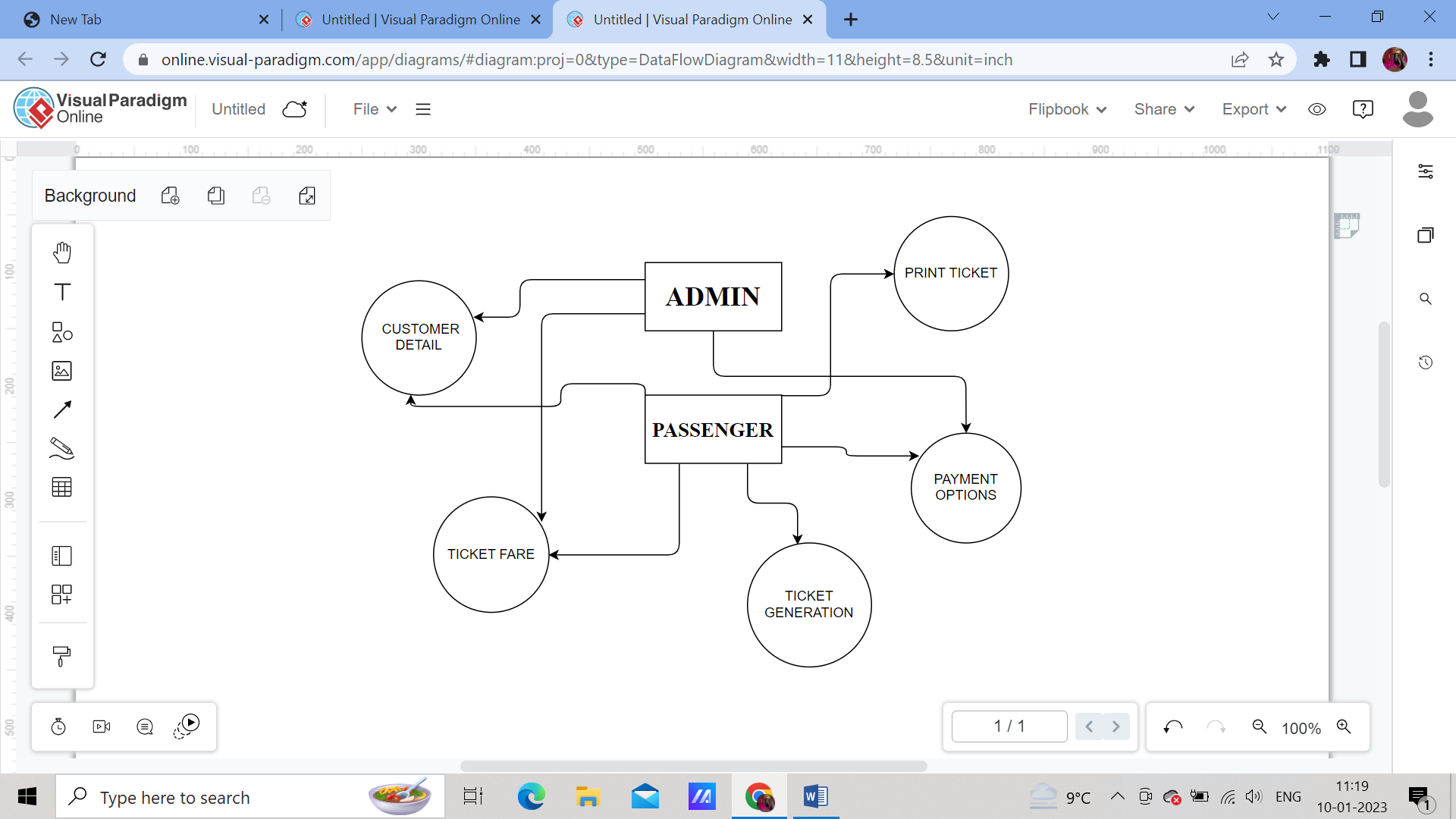
**Diagram

Description automatically generated**

* 1. **LEVEL - 1 DFD**

****

* 1. **LEVEL - 2 DFD**

****

**DFD-2 E-Ticketing Reservation System**

**CHAPTER-4**

**DATABASE DESIGN**

* 1. **ER DIAGRAM**

**ER Diagram** stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

##### Entities

Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.

Entities

Relationship

Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box

R

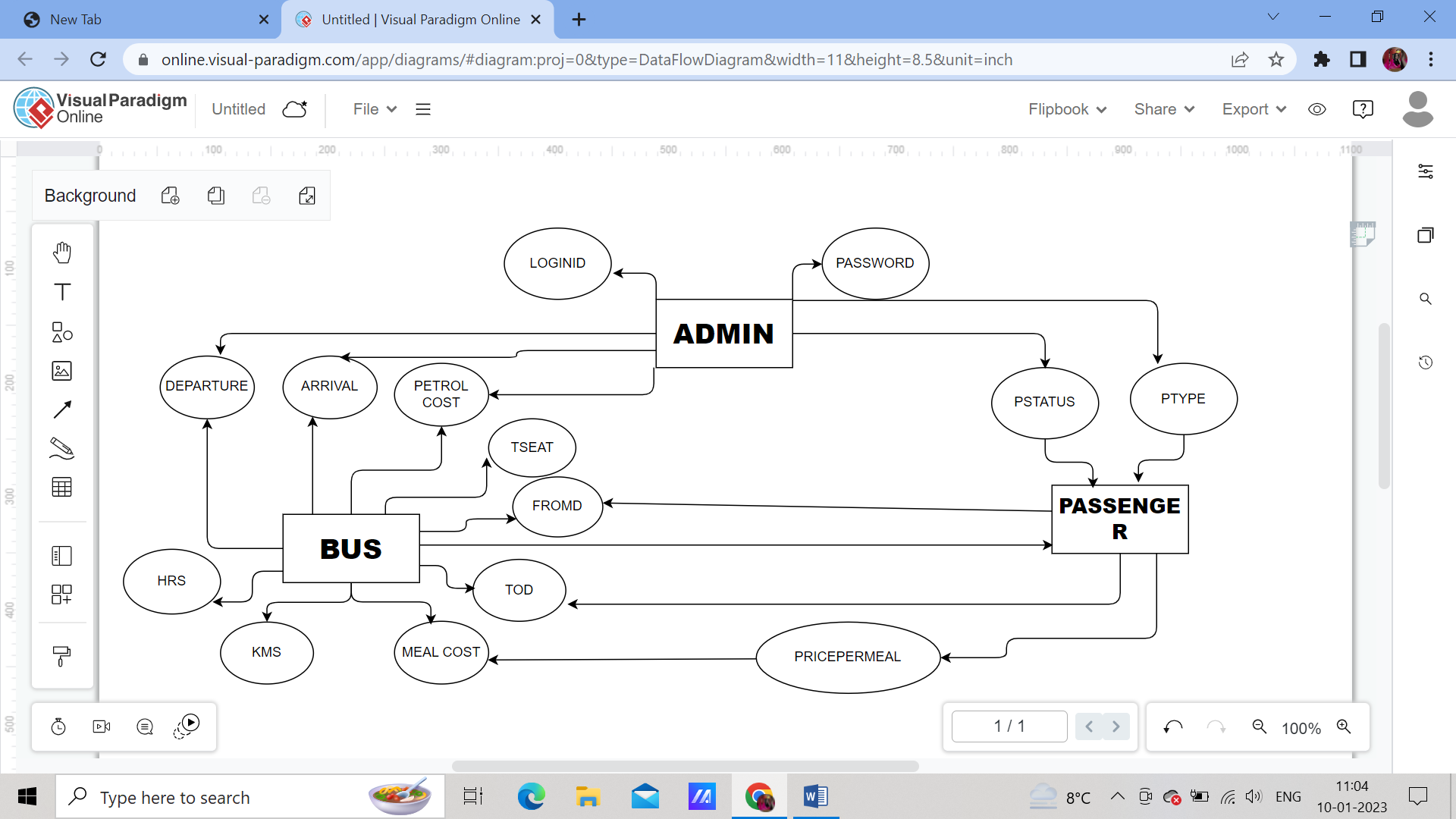
elationship

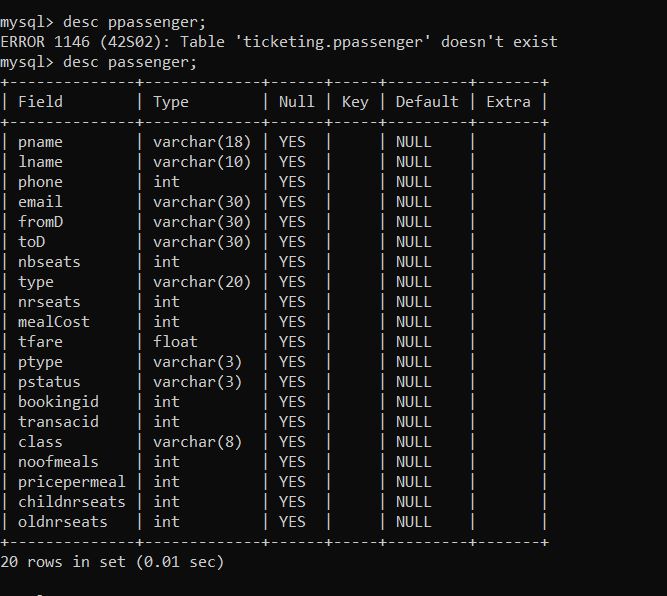
Attribute

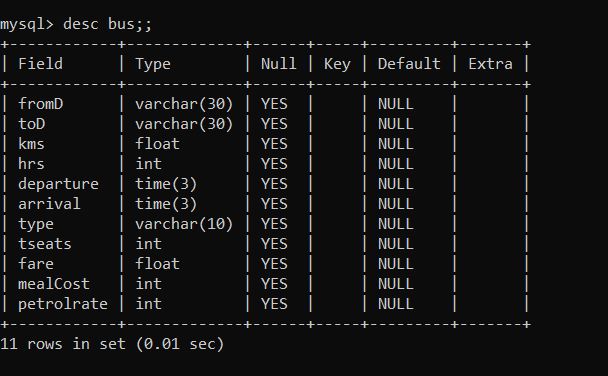
Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle).

A

ttribute

****

****

****

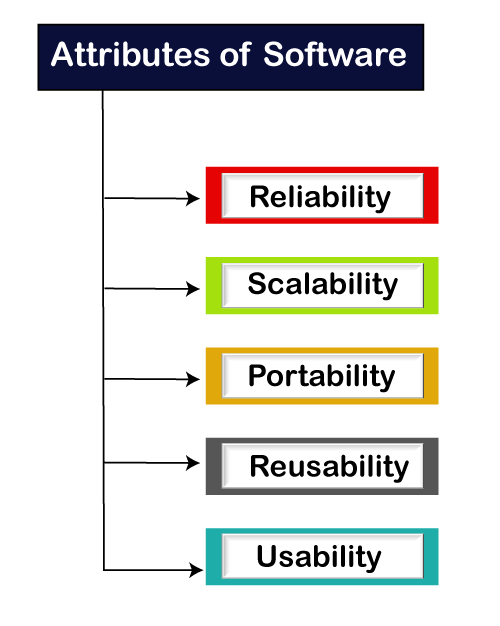
**CHAPTER-5**

**Testing**

* 1. **What is Software Testing ?**

Software testing is a process of identifying the correctness of software by considering its all attributes (Reliability, Scalability, Portability, Re-usability, Usability) and evaluating the execution of software components to find the software bugs or errors or defects.

Testing is mandatory because it will be a dangerous situation if the software fails any of time due to lack of testing. So, without testing software cannot be deployed to the end user.



**Fig-2**

###### LEVELS OF TESTING

* Unit testing
* Component integration testing
* System testing
* Acceptance testing



**Fig-3**

5.1.1 Unit testing

# A level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

5.1.2 Component Integration Testing

A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

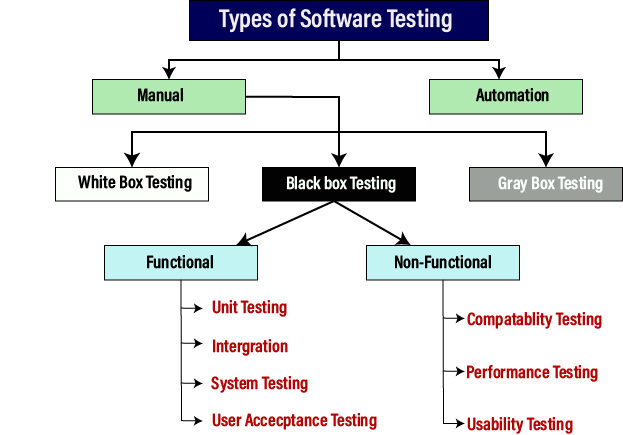
5.1.3 System Testing

A level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements.

5.1.4 Acceptance Testing

A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery.

* 1. **Types of Software Testing**



**Fig-4**

We are going to adopt **BLACK BOX TESTING**, Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

* + - Types of Black Box Testing
* **Functional Testing** – This black box testing type is related to the functional requirements of a system; it is done by software testers.
* **Non-Functional Testing** – This type of black box testing is not related to testing of specific functionality, but non-functional requirements such as performance, scalability, and usability.
* **Regression Testing** – Regression Testing is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

**CHAPTER-6**

**ROLES AND RESPONSIBILITY**

**ROLE**

Work as a developer, designer, tester of the application.

**RESPONSIBILITIES**

* Work on definition of development requirements and priorities.
* Data migration.
* Interfaces with other systems.
* Reporting configuration and deployment • Set up and maintenance of security rights and access permission.
* Contributing to technical strategy. Policy and procedure.
* Development and operation of technical testing programs.
* Production of technical documentation to agreed quality standards.
* Reporting on progress/issues to management and users.

**CHAPTER-7**

**CONCLUSION AND FUTURE ENHANCEMENTS**

It is safe to say bus or travel e-tickets are here to stay. They already make up a majority of the market, and the industry is committed to increasing that percentage. Airlines are moving toward a "self-service model," with passengers researching, [booking](https://coupons.howstuffworks.com/promo-codes/booking), buying, checking in and boarding virtually without airline assistance.

As all this occurs, analysts foresee great improvements in standardizing e-ticketing among travel and tourism sector.

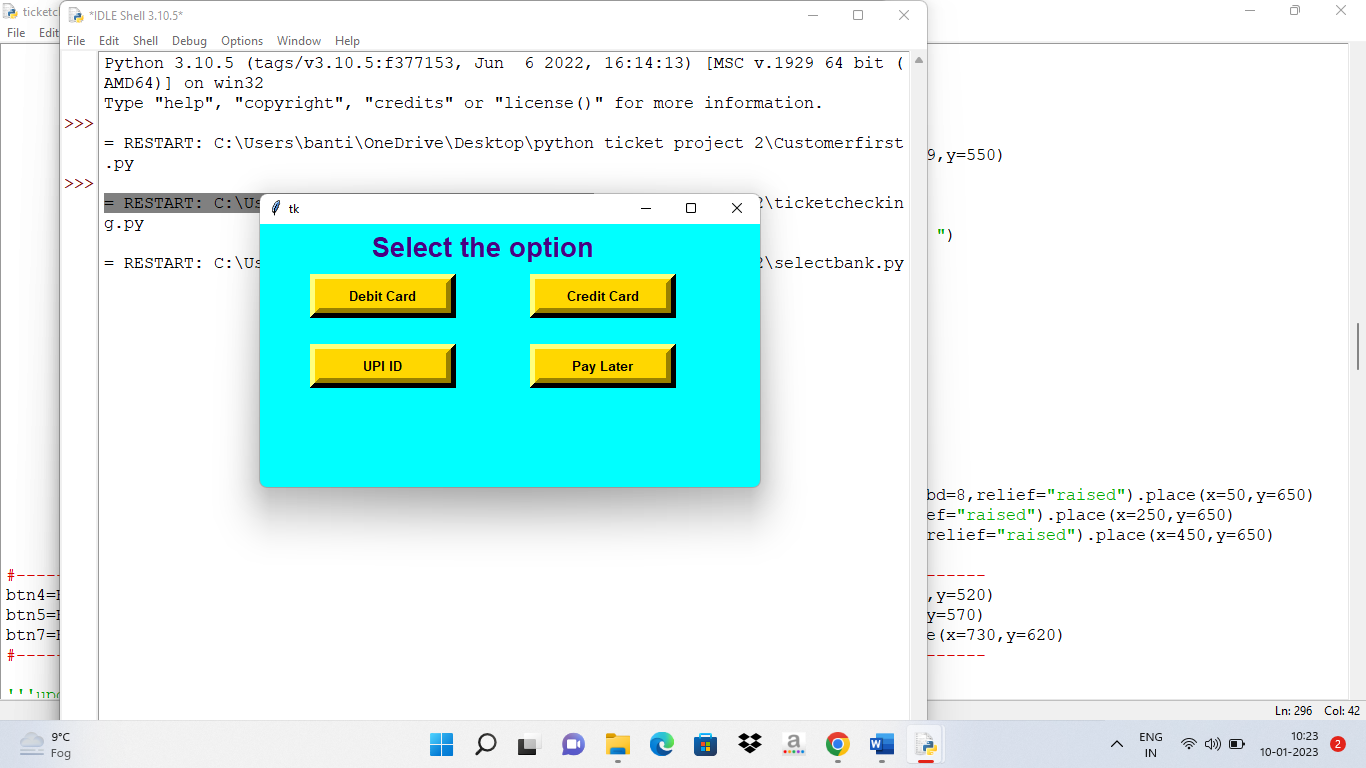
Though airline, train and accommodation, these services have not been included in the project but these can be inculpated in the project at any stage and point of time. This makes our project, scalable which is a good feature.

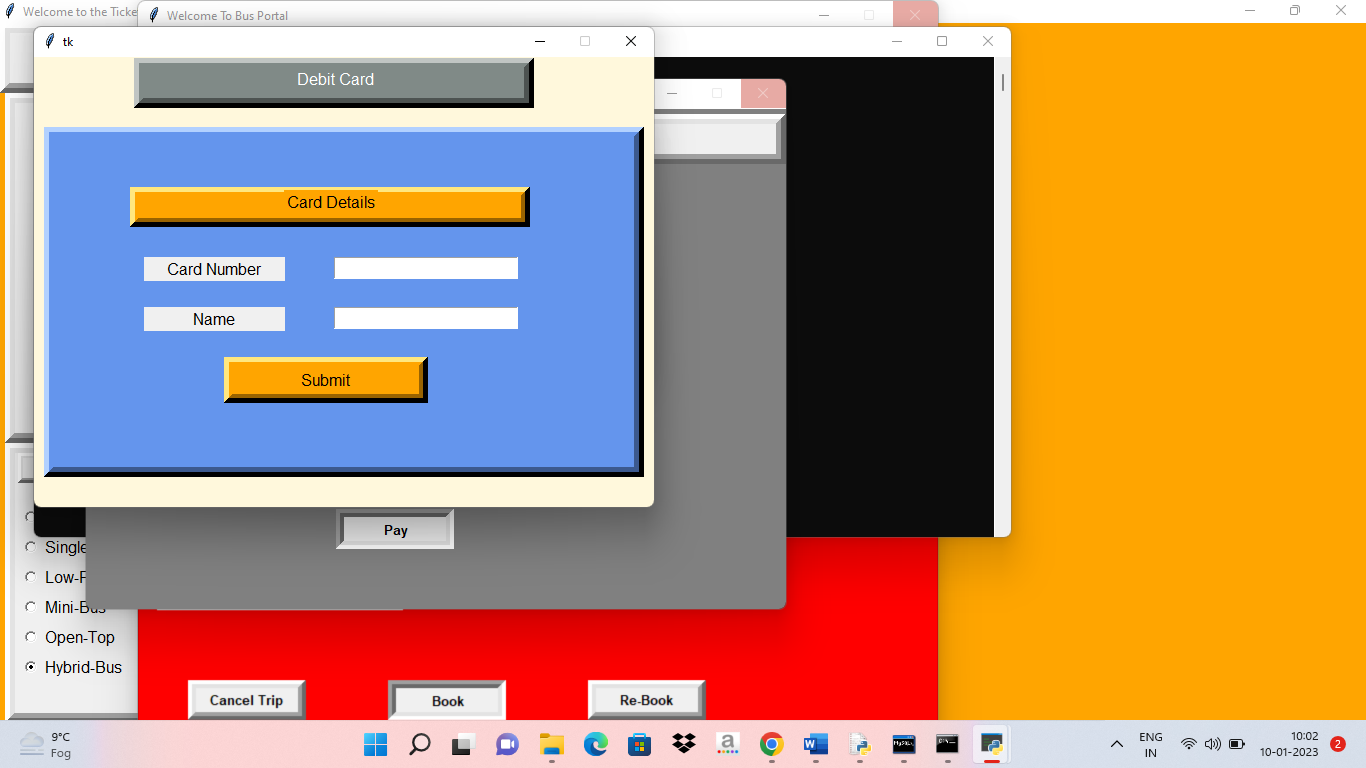
**APPENDICES (CODING SNIPPETS)**

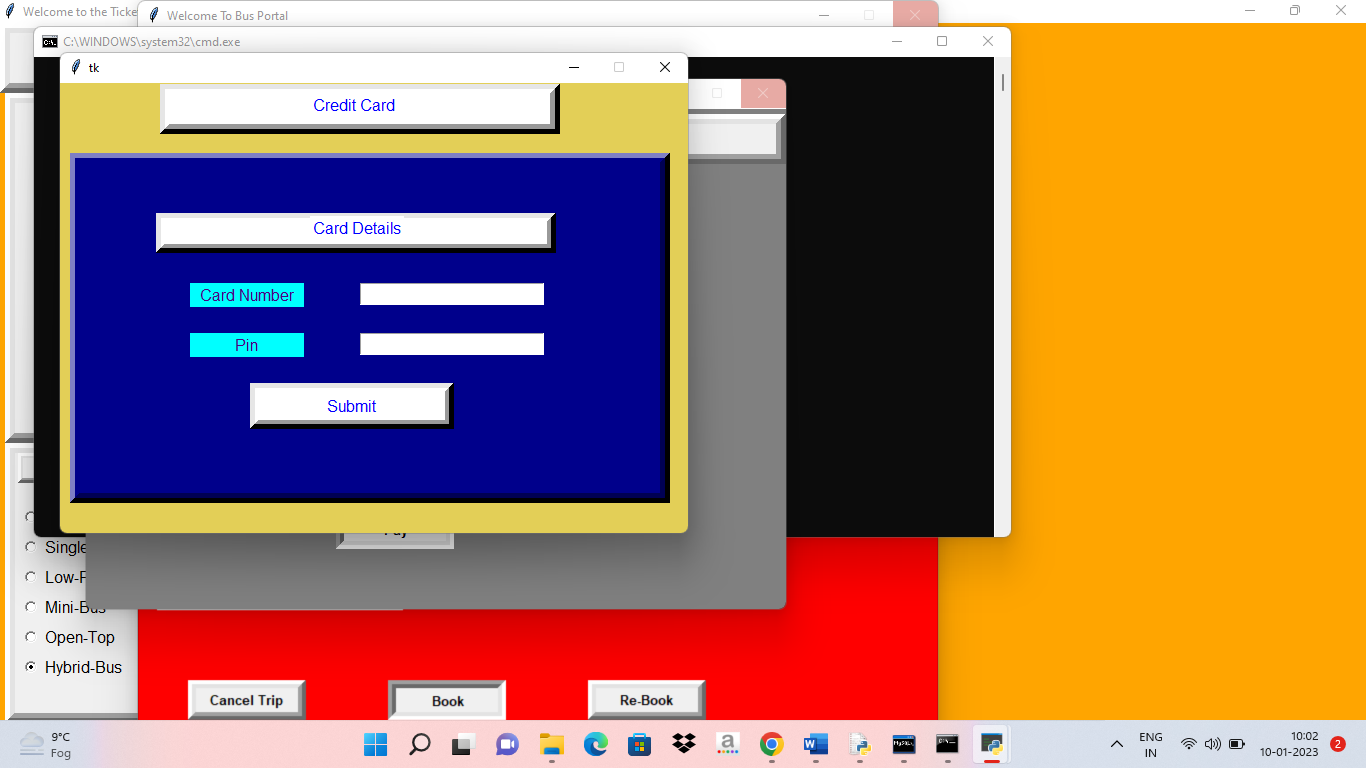
**Graphical user interface

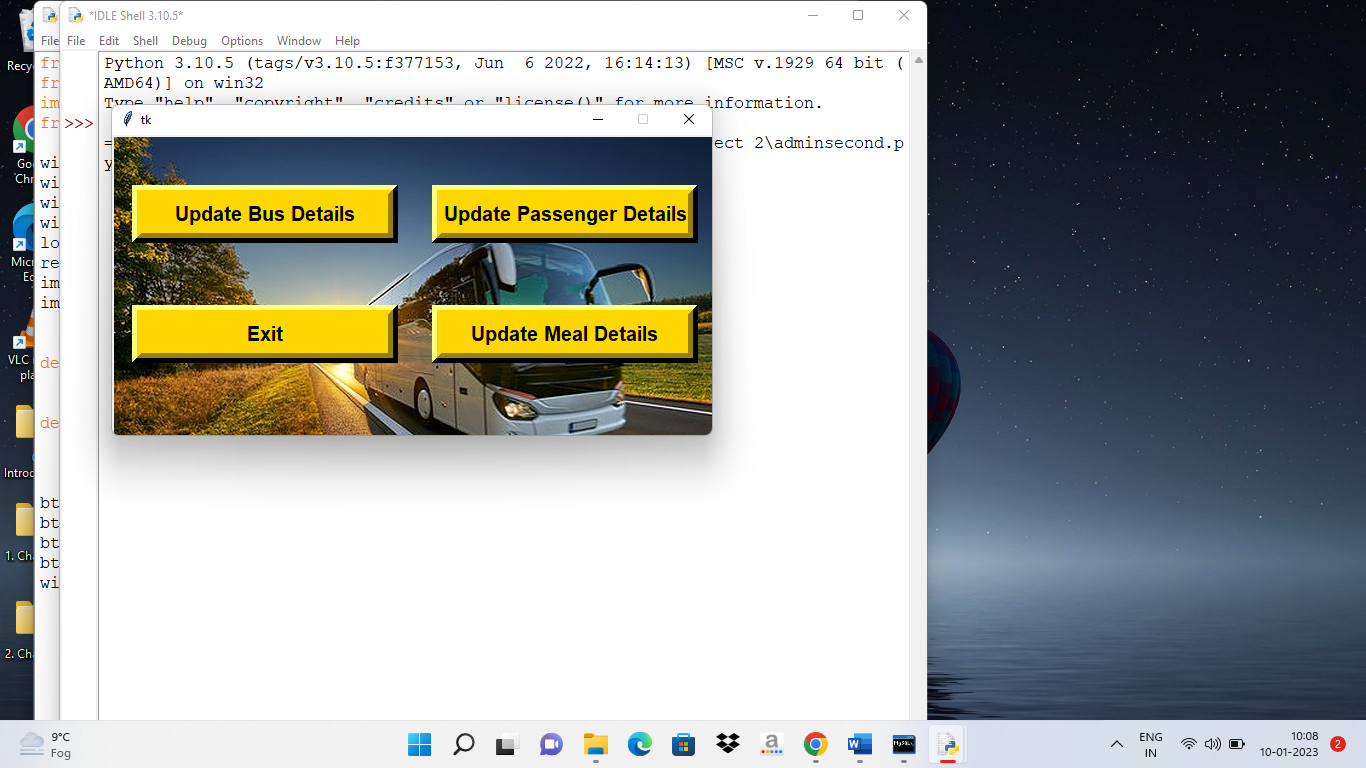
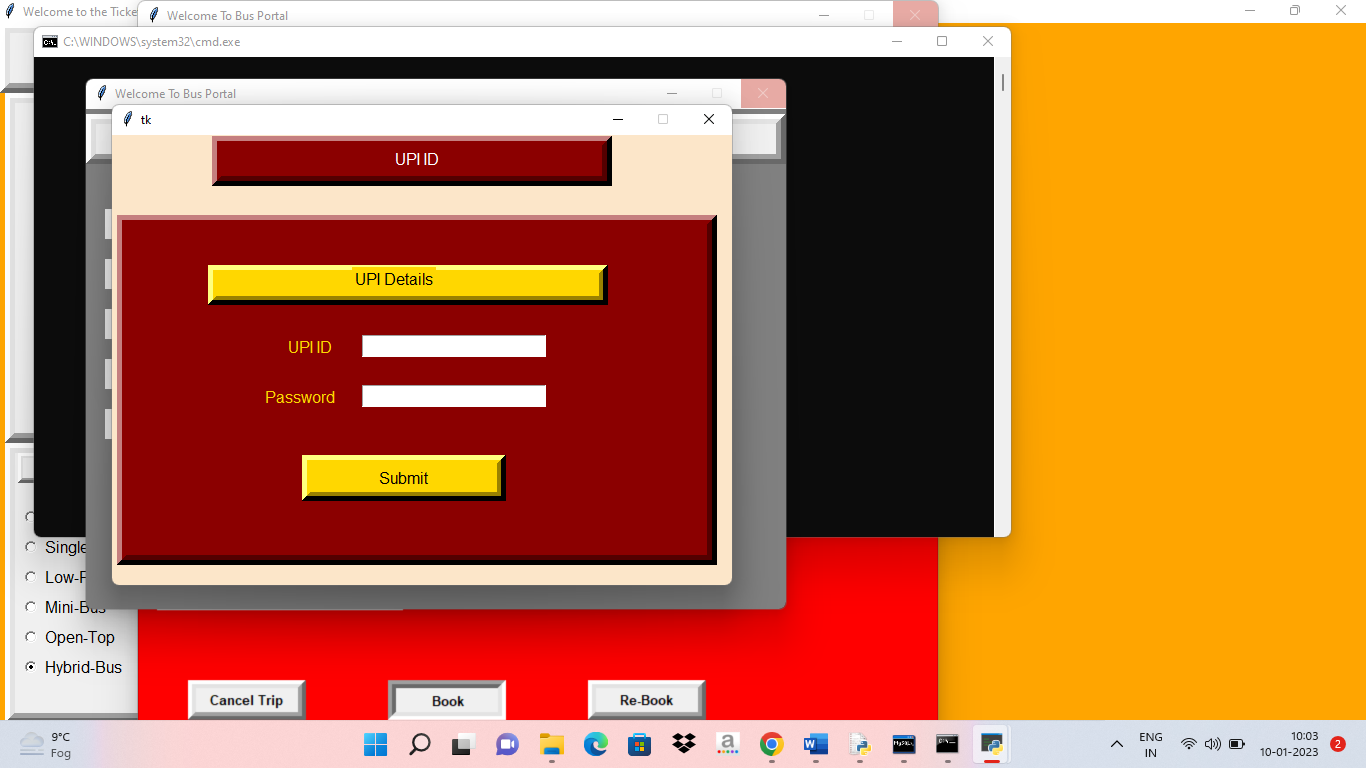
Description automatically generated**

****

****





****

from tkinter import\*

from tkinter import messagebox

import pymysql

import pymysql.cursors

import os,time

import random

value=random.randint(101,550)

win=Tk()

win.geometry("734x800")

win.config(bg="orange")

'''win.resizable(False,False)'''

win.title("Welcome to the Ticketing Portal")

frame1=Frame(win,bd=10,relief="raised",width=730,height=70).grid(row=0)

#-----------------------------------------------------------------------------

def data():

a=num1.get()

b=num2.get()

c=num3.get()

d=num4.get()

try:

conn = pymysql.connect(host='localhost',user='root',password='bantijha',db='ticketing')

mydb=conn.cursor()

mydb.execute("insert into passenger(pname,lname,phone,email) values('"+a+"','"+b+"','"+d+"','"+c+"')")

conn.commit()

messagebox.showinfo("Message","Data Submitted")

except:

conn.rollback()

messagebox.showerror("Message","Not Submitted")

conn.close()

#Customer Details

num1=StringVar()

num2=StringVar()

num3=StringVar()

num4=StringVar()

lb1=Label(frame1,text="Bus Ticketing Portal",font=('Helvetica', 18, 'bold')).grid(row=0)

frame2=Frame(win,bd=10,relief="raised",width=380,height=350).place(x=5,y=70)

lb2=Label(frame2,text="Customer Details",bd=5,relief="raised",width=28,font=10).place(x=35,y=80)

lb3=Label(frame2,text="First\_Name",font=5).place(x=135,y=126)

tx6=Entry(frame2,font=20,textvariable=num1).place(x=85,y=150)

lb4=Label(frame2,text="Last\_Name",font=5).place(x=135,y=175)

tx1=Entry(frame2,font=20,textvariable=num2).place(x=85,y=200)

lb4=Label(frame2,text="E\_Mail\_ID",font=5).place(x=140,y=225)

tx2=Entry(frame2,font=20,textvariable=num3).place(x=85,y=250)

lb4=Label(frame2,text="Contact",font=5).place(x=152,y=275)

tx2=Entry(frame2,font=20,textvariable=num4).place(x=85,y=300)

bt=Button(frame2,text="Submit",command=data,font=20,width=10,bd=10,relief="raised").place(x=120,y=350)

#---------------------------------------------------------------------------------------------

#Submit\_all

def submit\_all():

fro=from\_From.get()

too=to\_To.get()

passen=passe.get()

agea=age\_adult.get()

nomeal=feed.get()

agec=child.get()

ageo=old.get()

try:

conn = pymysql.connect(host='localhost',user='root',password='bantijha',db='ticketing')

mydb=conn.cursor()

mydb.execute("insert into passenger(fromD,toD,nbseats,nrseats,noofmeals,childnrseats,oldnrseats) values('"+fro+"','"+too+"','"+passen+"','"+agea+"','"+nomeal+"','"+agec+"','"+ageo+"')")

conn.commit()

messagebox.showinfo("Message"," Submitted")

except:

conn.rollback()

messagebox.showerror("Message"," not Submitted")

conn.close()

#Bus type

def bus():

m=d.get()

n=g.get()

try:

conn = pymysql.connect(host='localhost',user='root',password='bantijha',db='ticketing')

mydb=conn.cursor()

mydb.execute("insert into passenger(type,class) values('"+m+"','"+n+"')")

conn.commit()

messagebox.showinfo("Message","designation Submitted")

except:

conn.rollback()

messagebox.showerror("Message"," designation Not Submitted")

conn.close()

frame3=Frame(win,bd=10,relief="raised",width=380,height=280).place(x=5,y=420)

lb3=Label(frame3,text="Bus Type",bd=5,relief="raised",font=10,width=30).place(x=18,y=430)

d=StringVar()

d.set("Single-Decker")

rd=Radiobutton(frame3,text="Double-Decker",font=10,value="Double-Decker",variable=d).place(x=20,y=480)

rd2=Radiobutton(frame3,text="Single-Decker",font=10,value="Single-Decker",variable=d).place(x=20,y=510)

rd3=Radiobutton(frame3,text="Low-Floor",font=10,value="Low-Floor",variable=d).place(x=20,y=540)

rd4=Radiobutton(frame3,text="Mini-Bus",font=10,value="Mini-Bus",variable=d).place(x=20,y=570)

rd5=Radiobutton(frame3,text="Open-Top",font=10,value="Open-Top",variable=d).place(x=20,y=600)

rd6=Radiobutton(frame3,text="Hybrid-Bus",font=10,value="Hybrid-Bus",variable=d).place(x=20,y=630)

#CLASS

g=StringVar()

g.set("AC Class")

lb=Label(frame3,text="Class",font=("bold",18),relief="sunken").place(x=230,y=480)

rd7=Radiobutton(frame3,text="AC Class",font=10,value="AC Class",variable=g).place(x=220,y=510)

rd8=Radiobutton(frame3,text="Non-AC",font=10,value="Non-AC",variable=g).place(x=220,y=540)

bt=Button(frame3,text="Submit",command=bus,bd=10,relief="raised",font=10).place(x=220,y=620)

#-----------------------------------------------------------------------------------------------------------

#Bus Route

frame4=Frame(win,bd=10,relief="raised",width=350,height=350).place(x=380,y=70)

lb4=Label(frame4,text="Bus Route",bd=5,relief="raised",font=10,width=28).place(x=390,y=80)

#from

lb5=Label(frame4,text="From",width=5,font=10).place(x=510,y=115)

option=['Andhra Pradesh','Arunachal Pradesh','Assam','Bihar','Chhattisgarh','Goa','Gujarat','Haryana','Himachal Pradesh',

'Jammu and Kashmir','Jharkhand','Karnataka','Kerala','Madhya Pradesh','Maharashtra','Manipur','Meghalaya','Mizoram','Odisha',

'Nagaland','Punjab','Rajasthan','Sikkim','Tamil Nadu','Telangana','Tripura','Uttar Pradesh','Uttarakhand','West Bengal']

from\_From=StringVar()

from\_From.set("Select Destination")

drop=OptionMenu(frame4,from\_From,\*option).place(x=480,y=140)

#to

lb6=Label(frame4,text="To",width=5,font=10).place(x=510,y=172)

option=['Andhra Pradesh','Arunachal Pradesh','Assam','Bihar','Chhattisgarh','Goa','Gujarat','Haryana','Himachal Pradesh',

'Jammu and Kashmir','Jharkhand','Karnataka','Kerala','Madhya Pradesh','Maharashtra','Manipur','Meghalaya','Mizoram','Odisha',

'Nagaland','Punjab','Rajasthan','Sikkim','Tamil Nadu','Telangana','Tripura','Uttar Pradesh','Uttarakhand','West Bengal']

to\_To=StringVar()

to\_To.set("Select Destination")

drop=OptionMenu(frame4,to\_To,\*option).place(x=480,y=195)

#age

lb7=Label(frame4,text="Adult",font=10).place(x=410,y=308)

option=['1',

'2',

'3',

'4',

'5',

'6',

]

age\_adult=StringVar()

drop=OptionMenu(frame4,age\_adult,\*option).place(x=410,y=350)

lb7=Label(frame4,text="Child",font=10).place(x=510,y=308)

child=StringVar()

drop=OptionMenu(frame4,child,\*option).place(x=510,y=350)

lb7=Label(frame4,text="Old",font=10).place(x=620,y=308)

old=StringVar()

drop=OptionMenu(frame4,old,\*option).place(x=610,y=350)

#no of pssenger

passe=StringVar()

lb7=Label(frame4,text="Number of Passengers",width=20,font=10).place(x=450,y=235)

'''tx8=Label(frame4,font=10,text=passe)'''

tx8=Entry(frame4,font=10,textvariable=passe)

tx8.place(x=450,y=265)

#Meal

frame5=Frame(win,bd=10,relief="raised",width=350,height=280).place(x=380,y=420)

lb8=Label(frame5,text="Would you like Something?",width=22,font=10).place(x=380,y=440)

meals=StringVar()

meals.set("Yes")

meal=Radiobutton(frame5,text="Want Meal",font=10,variable=meals,value="Yes").place(x=510,y=470)

nomeal=Radiobutton(frame5,text="No Meal",font=10,variable=meals,value="No").place(x=510,y=500)

lb=Label(frame5,text="Number of Meals:",font=20,width=15).place(x=380,y=535)

feed=StringVar()

tx5=Entry(frame5,width=20,font=20,textvariable=feed).place(x=450,y=576)

bt=Button(frame3,text="Submit",command=submit\_all,bd=10,relief="raised",font=10).place(x=500,y=620)

'''def updatecontent():

agea=age\_adult.get()

agec=child.get()

ageo=old.get()

passe=int(agea)+int(agec)+int(ageo)

tx8.config(text=passe)

tx8.after (500,updatecontent)'''

#Reset

#---------------------------------------------------------------------------------------------

def res():

tx1=Entry(text="")

num1.set("")

num2.set("")

num3.set("")

num4.set("")

age\_adult.set("0")

child.set("0")

old.set("0")

feed.set("")

from\_From.set("Select Destination")

passe.set("")

to\_To.set("Select Destination")

#---------------------------------------------------------------------------------------------------

#2nd Module

def new():

submit\_all()

if num1.get()=="" and num2.get()=="" and num3.get()=="" and num4.get()=="" and passe.get()=="" and feed.get()=="":

messagebox.showerror("Error","Check all Credentials")

else:

messagebox.showinfo("Error","Welcome to Fare Page")

try:

conn = pymysql.connect(host='localhost',user='root',password='bantijha',db='ticketing')

mydb=conn.cursor()

mydb.execute("insert into passenger(pname,lname,phone,email) values('"+a+"','"+b+"','"+d+"','"+c+"')")

conn.commit()

messagebox.showinfo("Message","Data Submitted")

except:

conn.rollback()

messagebox.showerror("Message","Not Submitted")

conn.close()

root=Tk()

root.geometry("800x750")

root.title("Welcome To Bus Portal")

root.config(bg="Red")

root.resizable(False,False)

fr=Frame(root,width=800,bd=10,height=50,relief="raised")

fr.place(x=0,y=5)

frlb=Label(fr,text="Fare Portal",font=('arial',15,'bold'),width=15)

frlb.place(x=270,y=0)

lb=Label(root,text="Bus Route",font=('arial',15,'bold'),width=20).place(x=19,y=100)

fr=StringVar()

tx=Label(root,text=from\_From.get(),font=('arial',15,'bold'),width=20)

tx.place(x=300,y=100)

to=StringVar()

tx2=Label(root,text=to\_To.get(),font=('arial',15,'bold'),width=20)

tx2.place(x=550,y=100)

typ=StringVar()

lb2=Label(root,text="Bus Type",font=('arial',15,'bold'),width=20).place(x=19,y=150)

tx2=Label(root,text=d.get(),font=('arial',15,'bold'),width=20)

tx2.place(x=300,y=150)

rn=StringVar()

lb3=Label(root,text="Route Number",font=('arial',15,'bold'),width=20).place(x=19,y=200)

tx =Label(root,text=value,font=('arial',15,'bold'),width=20)

tx.place(x=300,y=200)

fare=StringVar()

lb4=Label(root,text="Per Seat Fare",font=('arial',15,'bold'),width=20).place(x=19,y=250)

text1=Label(root,text="",font=('arial',15,'bold'),textvariable=fare).place(x=300,y=250)

nop=StringVar()

lb5=Label(root,text="Number of Passenger",font=('arial',15,'bold'),width=20).place(x=19,y=300)

text2=Label(root,text=passe.get(),font=('arial',15,'bold')).place(x=300,y=300)

tsf=StringVar()

lb6=Label(root,text="Total Seat Fare",font=('arial',15,'bold'),width=20).place(x=19,y=350)

tx7=Entry(root,font=10,textvariable=tsf).place(x=300,y=350)

mc=StringVar()

lb7=Label(root,text="Meal Cost",font=('arial',15,'bold'),width=20).place(x=19,y=400)

tx8=Entry(root,font=10,textvariable=mc).place(x=300,y=400)

tf=StringVar()

lb8=Label(root,text="Total Fare",font=('arial',15,'bold'),width=20).place(x=19,y=450)

tx9=Entry(root,font=10,textvariable=tf).place(x=300,y=450)

tax=StringVar()

tax.set("18%")

lb9=Label(root,text="Tax",font=('arial',15,'bold'),width=20).place(x=19,y=500)

tx10=Entry(root,font=10,textvariable=tax).place(x=300,y=500)

tpf=StringVar()

lb10=Label(root,text="Total Payble For",font=('arial',15,'bold'),width=20).place(x=19,y=550)

tx10=Entry(root,font=10,textvariable=tpf).place(x=300,y=550)

def clearb():

qExit=messagebox.askyesno(" Quit System ", " Do you really want to cancel trip ? ")

if qExit > 0:

root.destroy()

return

else:

messagebox.showinfo("Message","Moving to Fare page")

def tick():

os.system("ticket3.py") #pay

def close():

root.destroy() #re book

btn=Button(root,text="Cancel Trip",font=('arial',10,'bold'),command=clearb,width=12,bd=8,relief="raised").place(x=50,y=650)

btn=Button(root,text="Book",command=tick,font=('arial',10,'bold'),width=12,bd=8,relief="raised").place(x=250,y=650)

btn=Button(root,text="Re-Book",command=close,font=('arial',10,'bold'),width=12,bd=8,relief="raised").place(x=450,y=650)

#-------------------------------------------------------------------------------------------------

btn4=Button(win,text="Reset",font=10,bd=10,relief="raised",width=15,command=res).place(x=730,y=520)

btn5=Button(win,text="Fare",command=new,font=10,bd=10,relief="raised",width=15).place(x=730,y=570)

btn7=Button(win,text="Exit",font=10,bd=10,relief="raised",width=15,command=win.destroy).place(x=730,y=620)

#-------------------------------------------------------------------------------------------------

'''updatecontent()'''

win.mainloop()

**Admin Module :**

from tkinter import\*

from tkinter import messagebox

import os

from PIL import ImageTk, Image

win=Tk()

win.config(bg="aqua")

win.geometry("600x300")

win.resizable(False,False)

load=Image.open('bus.jpg')

render=ImageTk.PhotoImage(load)

img=Label(win,image=render)

img.place(x=0,y=0)

def dd():

os.system("updatebus.py")

def ll():

os.system("updatepassengers.py")

btn=Button(win,bg='gold',fg='black',text="Update Bus Details",font=('arial',15,'bold'),width=20,bd=10,relief="raised",command=dd).place(x=20,y=50)

btn=Button(win,bg='gold',fg='black',text="Update Passenger Details",font=('arial',15,'bold'),width=20,bd=10,relief="raised",command=ll).place(x=320,y=50)

btn=Button(win,bg='gold',fg='black',text="Update Meal Details",font=('arial',15,'bold'),width=20,bd=10,relief="raised").place(x=320,y=170)

btn=Button(win,bg='gold',fg='black',text="Exit",font=('arial',15,'bold'),width=20,bd=10,relief="raised",command=win.destroy).place(x=20,y=170)

win.mainloop()

**REFERENCES**

* [**https://www.javatpoint.com/software-engineering-iterative-model**](https://www.javatpoint.com/software-engineering-iterative-model)
* [**https://www.smartdraw.com/**](https://www.smartdraw.com/)
* [**https://www.geeksforgeeks.org/software-engineering-software-design-process/**](https://www.geeksforgeeks.org/software-engineering-software-design-process/)
* [**https://en.wikipedia.org/wiki/Software\_design**](https://en.wikipedia.org/wiki/Software_design)