1. User Interface

Graphical user interface, application

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

1. Adding a Tourist Visa

Graphical user interface

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface, website

Description automatically generated

Notification of Visa Issued (Also added in the database)

Graphical user interface, text, application, chat or text message

Description automatically generated

1. If Visa application Already Exists

Graphical user interface, website

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

1. If Visa Application is rejected by the bank (Decision=F)

Graphical user interface

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

1. If Information is not complete, Missing Information

Graphical user interface, website

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

**Tourist Class:**

**package** MainProject;

**import** java.io.Serializable;

**public** **class** Tourist **implements** Serializable{

**private** **static** **final** **long** ***serialVersionUID*** = 6894463765965627542L;

**private** **int** touristId,touristNumber,touristPassportNo,cardNumber;

**private** String touristName,touristAddress,touristEmail;

**public** Tourist(**int** touristId, **int** touristNumber, **int** touristPassportNo, **int** cardNumber, String touristName,

String touristAddress, String touristEmail) {

**super**();

**this**.touristId = touristId;

**this**.touristNumber = touristNumber;

**this**.touristPassportNo = touristPassportNo;

**this**.cardNumber = cardNumber;

**this**.touristName = touristName;

**this**.touristAddress = touristAddress;

**this**.touristEmail = touristEmail;

}

**public** **int** getTouristId() {

**return** touristId;

}

**public** **void** setTouristId(**int** touristId) {

**this**.touristId = touristId;

}

**public** **int** getTouristNumber() {

**return** touristNumber;

}

**public** **void** setTouristNumber(**int** touristNumber) {

**this**.touristNumber = touristNumber;

}

**public** **int** getTouristPassportNo() {

**return** touristPassportNo;

}

**public** **void** setTouristPassportNo(**int** touristPassportNo) {

**this**.touristPassportNo = touristPassportNo;

}

**public** **int** getCardNumber() {

**return** cardNumber;

}

**public** **void** setCardNumber(**int** cardNumber) {

**this**.cardNumber = cardNumber;

}

**public** String getTouristName() {

**return** touristName;

}

**public** **void** setTouristName(String touristName) {

**this**.touristName = touristName;

}

**public** String getTouristAddress() {

**return** touristAddress;

}

**public** **void** setTouristAddress(String touristAddress) {

**this**.touristAddress = touristAddress;

}

**public** String getTouristEmail() {

**return** touristEmail;

}

**public** **void** setTouristEmail(String touristEmail) {

**this**.touristEmail = touristEmail;

}

@Override

**public** String toString() {

**return** "Tourist [touristId=" + touristId + ", touristNumber=" + touristNumber + ", touristPassportNo="

+ touristPassportNo + ", cardNumber=" + cardNumber + ", touristName=" + touristName

+ ", touristAddress=" + touristAddress + ", touristEmail=" + touristEmail + "]";

}

}

**Issue Tourist Visa Class:**

package MainProject;

import java.io.EOFException;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.util.ArrayList;

public class IssueTouristVisa {

ArrayList<Tourist> tourists;

public static void saveTourist( ArrayList<Tourist> t ) {

ObjectOutputStream out;

try {

if(true) {

out = new ObjectOutputStream(new FileOutputStream ("touristDetails.txt"));

while(t.size()!=0)

{

Object o=t.remove(0);

out.writeObject(o);

}

out.close();

}

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.getMessage();

}catch(EOFException e) {

System.out.println("EOF Exception: This is the end of the file.");

}

catch (IOException e) {

// TODO Auto-generated catch block

e.getMessage();

}

}

public static ArrayList<Tourist> ReadTest (){

ArrayList<Tourist> tourists = new ArrayList<>();

ObjectInputStream in;

Object obj;

try {

in = new ObjectInputStream(new FileInputStream("touristDetails.txt"));

while((obj = in.readObject()) != null)

{

Tourist t= (Tourist) obj;

tourists.add(t);

// System.out.println(t);

}

in.close();

} catch (FileNotFoundException e) {

// // TODO Auto-generated catch block

System.out.println(e.getMessage());

} catch (IOException | ClassNotFoundException e) {

System.out.println( e.getMessage());

}

// System.out.println(tourists.size() + " tourists read from the file.");

return tourists;

}

public IssueTouristVisa(ArrayList<Tourist> tourists) {

this.tourists = tourists;

}

public IssueTouristVisa(){

tourists= ReadTest();

for(int i=0;i<tourists.size();i++) {

System.out.println( tourists.get(i));

}

}

public String addTourist(Tourist t) {

tourists.add(t);

return String.format("added Tourist successfully\n");

}

public Tourist findTourist(int passportno) {

for (var x : tourists)

if (x.getTouristPassportNo()==passportno)

return x;

return null;

}

public ArrayList<Tourist> getTourists() {

return tourists;

}

public void setTourists(ArrayList<Tourist> tourists) {

this.tourists = tourists;

}

}

**Main View:**

package mainview;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.stage.Stage;

import javafx.scene.Scene;

import javafx.scene.layout.AnchorPane;

public class Main extends Application {

@Override

public void start(Stage primaryStage) {

try {

AnchorPane root = FXMLLoader.load(getClass().getResource("MainView.fxml"));

Scene scene = new Scene(root);

primaryStage.setScene(scene);

primaryStage.setTitle("Tourist Application System!");

primaryStage.show();

} catch (Exception e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

launch(args);

}

}

**Main View Controller:**

package mainview;

import java.io.IOException;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.scene.Scene;

import javafx.scene.layout.AnchorPane;

import javafx.stage.Stage;

public class MainViewController {

@FXML

void handleButtonAction(ActionEvent event) {

FXMLLoader loading = new FXMLLoader(getClass().getResource("Tourist\\touristView.fxml"));

try {

AnchorPane pane = (AnchorPane) loading.load();

Scene scene = new Scene(pane);

Stage stage = new Stage();

stage.setScene(scene);

stage.show();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

**Tourist Controller:**

package mainview.Tourist;

import javafx.application.Platform;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.layout.AnchorPane;

import javafx.stage.Stage;

import MainProject.Tourist;

import java.net.URL;

import java.util.ResourceBundle;

import javax.swing.JOptionPane;

import MainProject.IssueTouristVisa;

public class TouristController implements Initializable ,TouristInteractionForm {

@FXML

private Label bankDecisionLabel;

@FXML

private Label creditCardLabel;

@FXML

private Label resultNotice;

@FXML

private TextField taddress;

@FXML

private Button tbutton;

@FXML

private TextField tcardnumber;

@FXML

private TextField tdecision;

@FXML

private TextField temail;

@FXML

private TextField tid;

@FXML

private TextField tname;

@FXML

private TextField tnumber;

@FXML

private TextField tpassport;

@FXML

private Button tbutton1;

ObservableList<Integer> observableTObjectsID = FXCollections.observableArrayList(TouristRepository.getIds());

ObservableList<Integer> observableTObjectsPassport= FXCollections.observableArrayList(TouristRepository.getPassportNumber());

Tourist t;

@FXML

void handleButtonAction(ActionEvent event) {

if(tid.getText()!="" && tname.getText()!="" && tpassport.getText()!="" && tnumber.getText()!="" && temail.getText()!="" && taddress.getText()!="") {

t=new Tourist(0,0,0,0,"null","null","null");

t.setTouristId(Integer.parseInt(tid.getText()));

t.setTouristPassportNo(Integer.parseInt(tpassport.getText()));

t.setTouristNumber(Integer.parseInt(tnumber.getText()));

t.setTouristName(tname.getText());

t.setTouristEmail(temail.getText());

t.setTouristAddress(taddress.getText());

boolean booleanID=true,booleanPassport=true;

if(observableTObjectsID!=null) {

for(int i=0; i<observableTObjectsID.size();i++) {

if(observableTObjectsID.get(i)==Integer.parseInt(tid.getText()) || observableTObjectsPassport.get(i)==Integer.parseInt(tpassport.getText())) {

booleanID=false;

booleanPassport=false;

}

}

}

if(booleanID==true && booleanPassport==true) {

tbutton.setVisible(false);

tid.setEditable(false);

tpassport.setEditable(false);

tnumber.setEditable(false);

temail.setEditable(false);

taddress.setEditable(false);

tname.setEditable(false);

tid.setOpacity(0.7);

tpassport.setOpacity(0.7);

temail.setOpacity(0.7);

tnumber.setOpacity(0.7);

taddress.setOpacity(0.7);

tname.setOpacity(0.7);

creditCardLabel.setVisible(true);

tcardnumber.setVisible(true);

bankDecisionLabel.setVisible(true);

tdecision.setVisible(true);

tbutton1.setVisible(true);

}

else {

JOptionPane.showMessageDialog(null, "This user already exists");

}

}

else {

JOptionPane.showMessageDialog(null, "Missing Information! Try Again Later");

Platform.exit();

}

}

@FXML

void handleButton2Action(ActionEvent event) {

if(tcardnumber.getText()!=""){

t.setCardNumber(Integer.parseInt(tcardnumber.getText()));

if(tdecision.getText()!="") {

if(tdecision.getText().equalsIgnoreCase("f")) {

JOptionPane.showMessageDialog(null, "Bank Decision is Rejected ! Application Rejected !");

Platform.exit();

}

else {

addTourist(t);

JOptionPane.showMessageDialog(null, "Approved ! Visa Issued "+ "\n\nSummary : \n\n" + "Name of Person : "+tname.getText()+"\nID Number : "+tid.getText()+"\nPassport Number : "+tpassport.getText()+

"\nContact Number : "+tnumber.getText()+"\nContact Email ID :"+temail.getText()+"\nAddress of Person : "+taddress.getText());

}

}

else {

JOptionPane.showMessageDialog(null, "No Bank Decision Inputed ! Rejecting Application ... ");

}

}

else {

JOptionPane.showMessageDialog(null, "Credit Card Not Entered by the user!");

}

}

public interface touristInteraction {

public void addTourist(Tourist t);

}

@Override

public void initialize(URL arg0, ResourceBundle arg1) {

// TODO Auto-generated method stub

}

@Override

public void addTourist(Tourist t) {

TouristRepository.saveTourist(t);

}

}

**Tourist Interaction Form:**

**package** mainview.Tourist;

**import** MainProject.Tourist;

**public** **interface** TouristInteractionForm {

**public** **void** addTourist(Tourist t);

}

package mainview.Tourist;

import java.util.ArrayList;

import MainProject.IssueTouristVisa;

import MainProject.Tourist;

**Tourist Repository :**

public class TouristRepository {

public static ArrayList<Integer> getIds() {

ArrayList<Integer> touristIds = new ArrayList<>();

ArrayList<Tourist> t=IssueTouristVisa.ReadTest();

if(t.size()!=0) {

for (var tourist : t) {

touristIds.add(tourist.getTouristId());

}

}

return touristIds;

}

public static ArrayList<Integer> getPassportNumber() {

ArrayList<Integer> touristPassport = new ArrayList<>();

ArrayList<Tourist> t=IssueTouristVisa.ReadTest();

if(t.size()!=0) {

for (var tourist : t) {

touristPassport.add(tourist.getTouristPassportNo());

}

}

return touristPassport;

}

public static void saveTourist(Tourist T) {

ArrayList<Tourist> t = IssueTouristVisa.ReadTest();

t.add(T);

IssueTouristVisa.saveTourist(t);

}

}