HAJJ AND UMRAH BOOKING SYSTEM

By Cem & Hebak

OUR TEAM

Muhammad Faiz Bin Zakaria
A22EC0208

Muhammad Ilman Bin Mohd Khairi
A22EC0215

Alif Aiman Bin Mansor
A22EC0137

Muhammad Iman Amier Bin Abu Bakar
A22EC0128

LIST OF CONTENTS

PROJECT DESCRIPTION

SYNOPSIS

- making a system of Haji and Umrah that more smoothly and flawlessly
- some of our features : jemaah health, visa approval and tour package
- other additional features : Haji and Umrah packages, room size
- got a lot of interesting places that rich in historical knowledge
- all these features are made to be the most convenient system of Umrah and Haji

OBJECTIVE

- allowing any individual to complete the Hajj and Umrah, while also raising the congregation's degree of faith and making sure participants are aware of both the practical and spiritual aspects of these pilgrimages.

- providing pilgrims with aid in organizing their Hajj and Umrah, including help with travel, lodging, health, required documentation, and creating a clear schedule to ensure a hassle-free, well-planned experience.

- providing pilgrims with assistance regarding the Hajj and Umrah procedures through seminars designed to meet their religious and emotional needs.

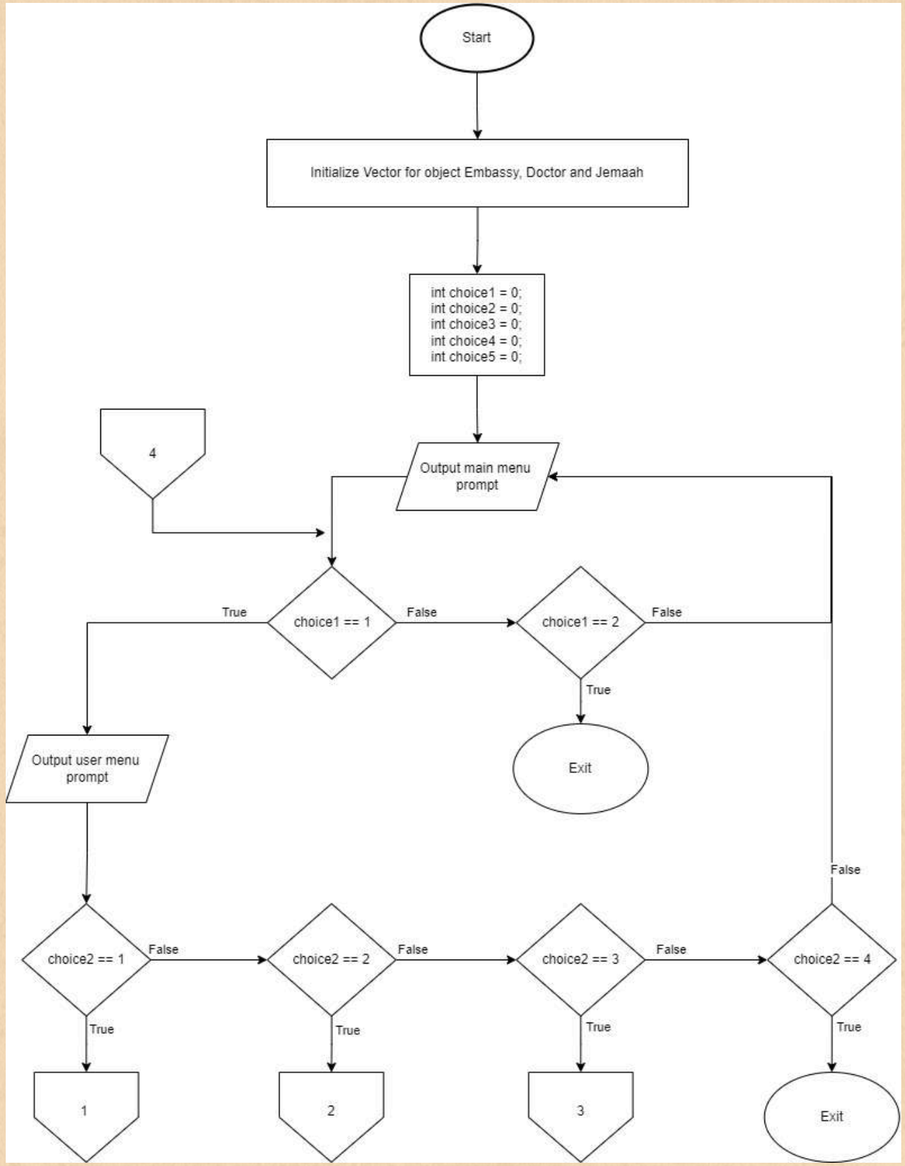
- allowing the audience to experience learning about and getting to know the local culture while also highlighting the significance of the Hajj and Umrah to deepen their understanding of this act of worship.

SCOPE OF PROJECT

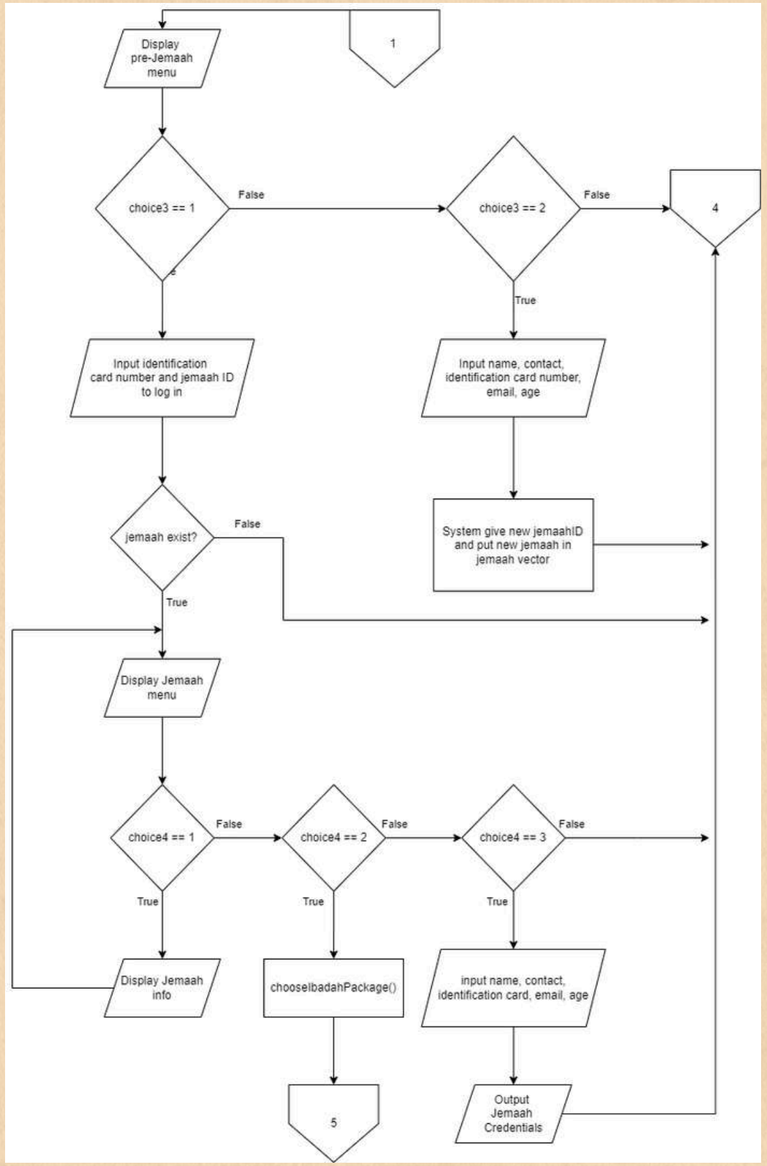
- user can choose packages, update personal info and choose the tourist packages
- assigning doctor to its jemaah, setting any disease and medicine to the jemaah while approve the jemaah itself in continueing the ibadah
- user can know wether their visa is approve or failed and asking them to reapply it back
- contain 5 type of haji and umrah packages including its date and any tour packages they want
- allowing the jemaah to pick the type of room according their preference

PROJECT DESIGNS

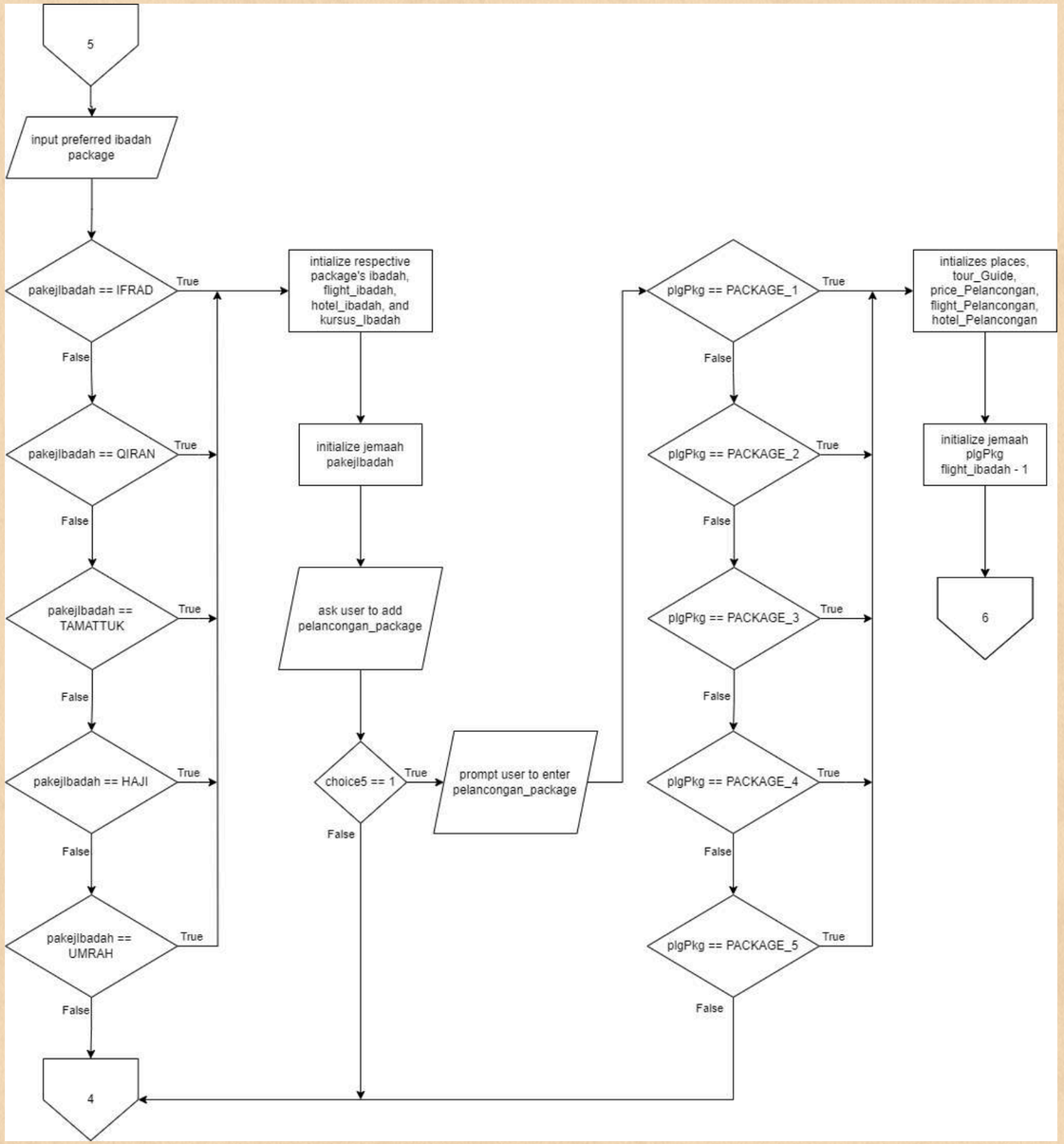
FLOWCHART MAIN



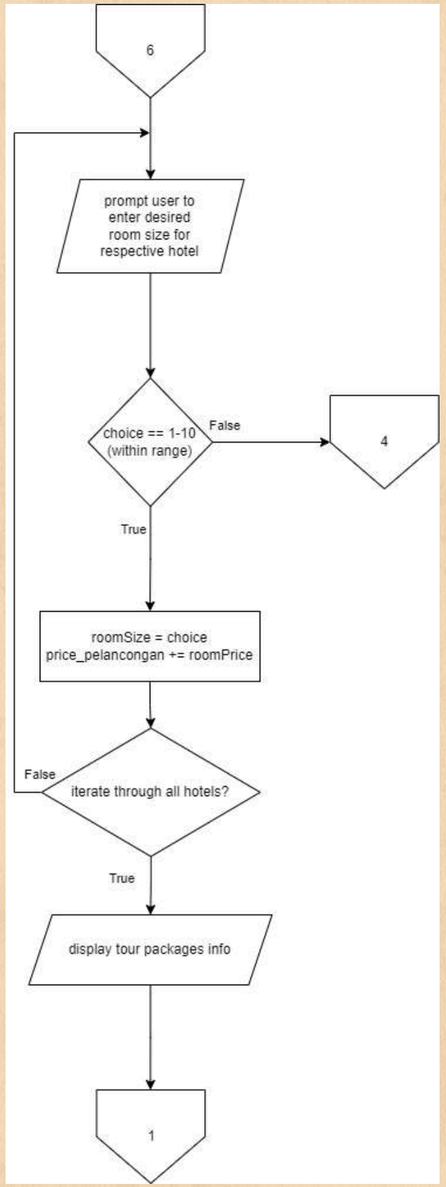
FLOWCHART JEMAAH



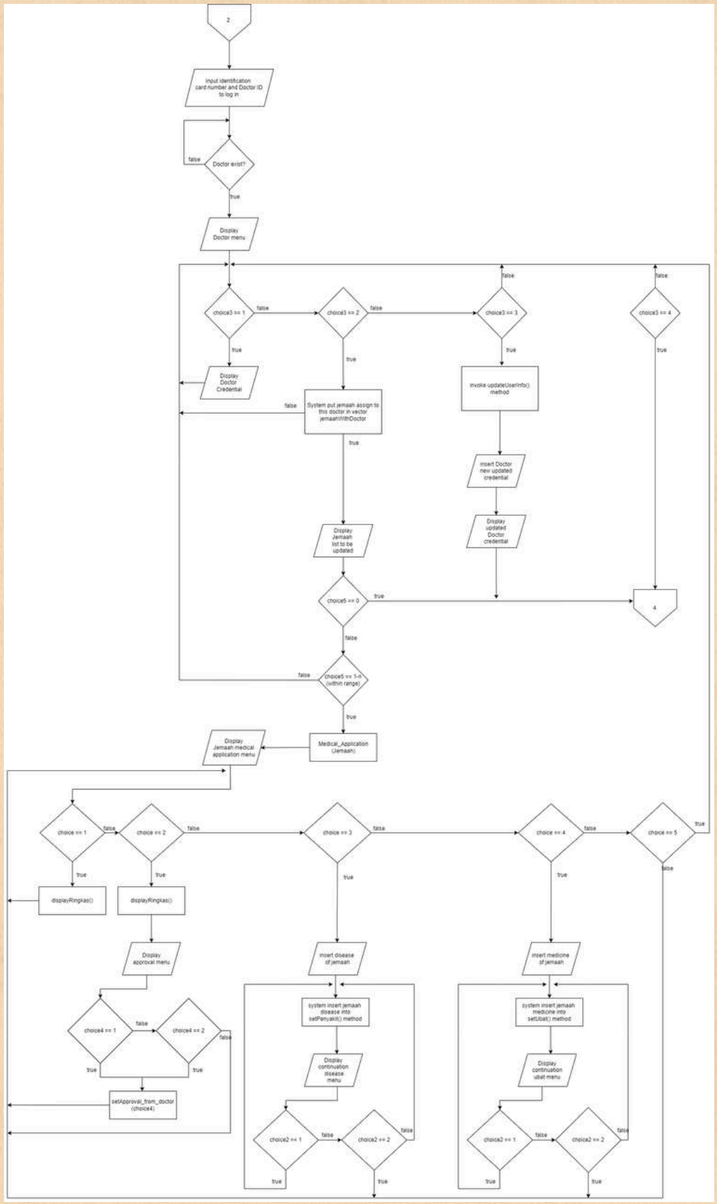
FLOWCHART JEMAAH (CONT)



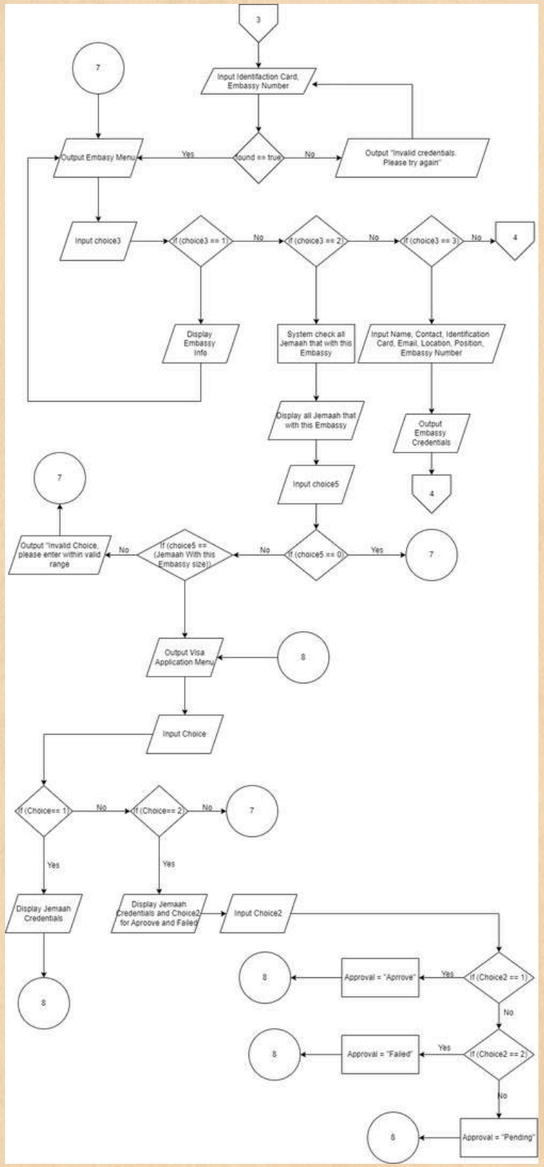
FLOWCHART JEMAAH (CONT)



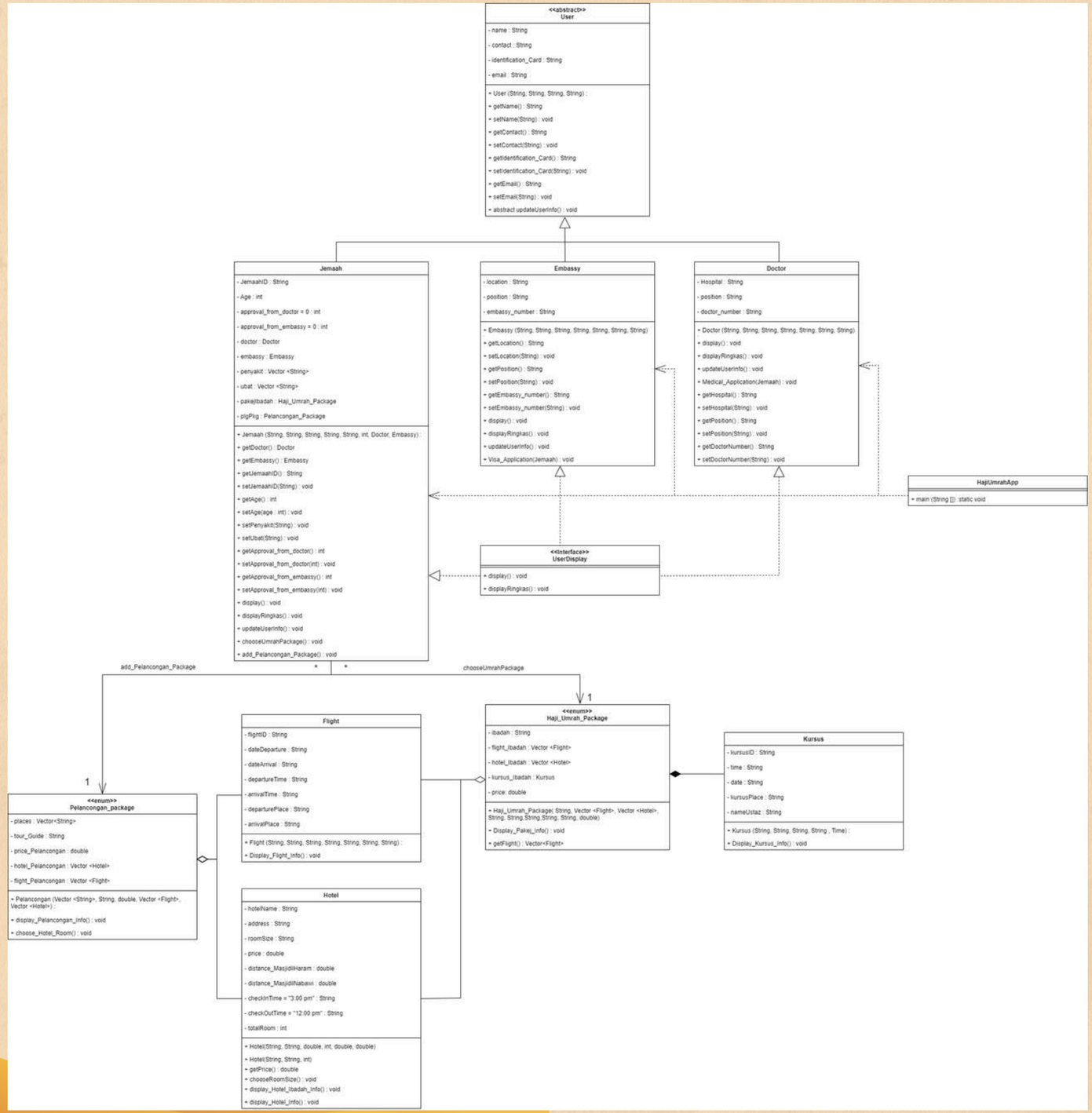
FLOWCHART DOCTOR



FLOWCHART EMBASSY



CLASS DIAGRAM



OBJECT ORIENTED CONCEPTS EMPLOYED

OO CONCEPTS

Data Hiding

```java
public class Kursus {
    private String kursusID;
    private String time;
    private String date;
    private String kursusPlace;
    private String nameUstaz;

    public Kursus(String kursusID, String time, String date, String kursusPlace, String nameUstaz)
        this.kursusID = kursusID;
        this.time = time;
        this.date = date;
        this.kursusPlace = kursusPlace;
        this.nameUstaz = nameUstaz;
    }

    public void Display_Kursus_info() {
        System.out.printf(format:"%-37s: %22s%n", ...args:"Kursus ID", kursusID);
        System.out.printf(format:"%-37s: %22s%n", ...args:"Time", time);
        System.out.printf(format:"%-37s: %22s%n", ...args:"Date", date);
        System.out.printf(format:"%-37s: %22s%n", ...args:"Kursus Location", kursusPlace);
        System.out.printf(format:"%-37s: %22s%n", ...args:"Ustaz", nameUstaz);
    }
}
```

Kursus.java

Encapsulation

```java
public class Flight {
    private String flightID, dateDeparture, dateArrival, departureTime, arrivalTime, departurePlace, arrivalPlace;

    public Flight(String flightID, String dateDeparture, String dateArrival, String departureTime, String arrivalTime,
            String departurePlace, String arrivalPlace) {
        this.flightID = flightID;
        this.dateDeparture = dateDeparture;
        this.dateArrival = dateArrival;
        this.departureTime = departureTime;
        this.arrivalTime = arrivalTime;
        this.departurePlace = departurePlace;
        this.arrivalPlace = arrivalPlace;
    }

    public void display_Flight_Info() {

        System.out.printf(format:"%-35s: %20s%n", ...args:"Flight ID", flightID);
        System.out.printf(format:"%-35s: %20s%n", ...args:"Date of Departure", dateDeparture);
        System.out.printf(format:"%-35s: %20s%n", ...args:"Date of Arrival", dateArrival);
        System.out.printf(format:"%-35s: %20s%n", ...args:"Departure Time", departureTime);
        System.out.printf(format:"%-35s: %20s%n", ...args:"Arrival Time", arrivalTime);
        System.out.printf(format:"%-35s: %20s%n", ...args:"Departure Location", departurePlace);
        System.out.printf(format:"%-35s: %20s%n", ...args:"Arrival Location", arrivalPlace);

    }
}
```

Flight.java

OO CONCEPTS

Aggregation

```java
private Vector<String> places;
private String tour_Guide;
private double price_Pelancongan;
private Vector<Flight> flight_Pelancongan;
private Vector<Hotel> hotel_Pelancongan;
Scanner inp = new Scanner(System.in);

private Pelancongan_Package(Vector<String> places, String tour_Guide, double price_Pelancongan,
        Vector<Flight> flight_Pelancongan, Vector<Hotel> hotel_Pelancongan) {
    this.places = places;
    this.tour_Guide = tour_Guide;
    this.price_Pelancongan = price_Pelancongan;
    this.flight_Pelancongan = flight_Pelancongan;
    this.hotel_Pelancongan = hotel_Pelancongan;
}
```

Pelancongan_package.java

Composition

```java
private String ibadah;
private Vector<Flight> flight_Ibadah;
private Vector<Hotel> hotel_Ibadah;
private Kursus kursus_Ibadah;

private double price;
Scanner inp = new Scanner(System.in);

private Haji_Umrah_Package(String ibadah, Vector<Flight> flight_Ibadah, Vector<Hotel> hotel_Ibadah,
        String kursusID, String time, String date, String kursusPlace, String nameUstaz, double price) {
    this.ibadah = ibadah;
    this.flight_Ibadah = flight_Ibadah;
    this.hotel_Ibadah = hotel_Ibadah;
    kursus_Ibadah = new Kursus(kursusID, time, date, kursusPlace, nameUstaz);
    this.price = price;
}
```

Haji_Umrah_Package.java

OO CONCEPTS

Inheritance

```java
public class Doctor extends User implements userDisplay {
    private String hospital, position, doctor_Number;
    protected Scanner inp = new Scanner(System.in);

    public Doctor(String name, String contact, String identification_Card, String email, String hospital,
            String position, String doctor_Number) {
        super(name, contact, identification_Card, email);
        this.hospital = hospital;
        this.position = position;
        this.doctor_Number = doctor_Number;
    }

    public String getHospital() {
        return hospital;
    }

    public void setHospital(String hospital) {
        this.hospital = hospital;
    }

    public String getPosition() {
        return position;
    }

    public void setPosition(String position) {
        this.position = position;
    }
```

Doctor.java

OO CONCEPTS

Abstract

```java
import java.util.*;

public abstract class User { // using abstarct
    private String name, contact, identification_Card, email;
    protected Scanner inp = new Scanner(System.in);

    public User(String name, String contact, String identification_Card, String email) {
        this.name = name;
        this.contact = contact;
        this.identification_Card = identification_Card;
        this.email = email;
    }

    public String getName() {  return name;  }

    public void setName(String name) {  this.name = name;  }

    public String getContact() { return contact;    }

    public void setContact(String contact) {  this.contact = contact;  }

    public String getIdentification_Card() {  return identification_Card;  }

    public void setIdentification_Card(String identification_Card) {  this.identification_Card = identification_Card;  }

    public String getEmail() {  return email;  }

    public void setEmail(String email) {  this.email = email;  }

    public abstract void updateUserInfo();

}
```

User.java

Interface

```java
public interface userDisplay {
    public void display();
    public void displayRingkas(); // tanpa personal info
}
```

userDisplay.java

OO CONCEPTS

Abstract / Polymorphism

```java
public void updateUserInfo() {

    System.out.println(x:"\n=======================================");
    System.out.printf(format:"%29s\n", ...args:"UPDATE CREDENTIALS");
    System.out.println(x:"=======================================");
    System.out.print(s:"Please enter your credentials: \nName: ");
    setName(inp.nextLine());
    System.out.print(s:"Contact :");
    setContact(inp.nextLine());
    System.out.print(s:"Identification Card : ");
    setIdentification_Card(inp.nextLine());
    System.out.print(s:"Email: ");
    setEmail(inp.nextLine());

    System.out.print(s:"Location: ");
    location = inp.nextLine();
    System.out.print(s:"Position: ");
    position = inp.nextLine();
    System.out.print(s:"Embassy Number: ");
    embassy_number = inp.nextLine();

    // new credentialsdsadasdasdas
    display();
}
```

Embassy.java

Abstract / Polymorphism

```java
// Interface
public void display() {
    System.out.println(x:"\n=======================================");
    System.out.printf(format:"%29s\n", ...args:"EMBASSY CREDENTIALS");
    System.out.println(x:"=======================================");
    System.out.println("Name : " + getName());
    System.out.println("Contact :" + getContact());
    System.out.println("Identification Card : " + getIdentification_Card());
    System.out.println("Email: " + getEmail());
    System.out.println("Location: " + location);
    System.out.println("Position: " + position);
    System.out.println("Embassy Number: " + embassy_number);
    System.out.println();
}

public void displayRingkas() {
    System.out.println(x:"\n=======================================");
    System.out.printf(format:"%29s\n", ...args:"EMBASSY CREDENTIALS");
    System.out.println(x:"=======================================");
    System.out.println("Name : " + getName());
    System.out.println("Contact :" + getContact());
    System.out.println("Email: " + getEmail());
    System.out.println("Location: " + location);
    System.out.println("Position: " + position);
    System.out.println();
}
```

Embassy.java

OO CONCEPTS

Exception Handling

```
case 2: // create new jemaah and insert into vector
        try{
                String name, contact, identification_Card, email,
                                jemaahID;
                int age;
                System.out.println(
                                x:"---------- NEW JEMAAH REGISTRATION ----------");
                System.out.print(s:"Name: ");
                name = inp.nextLine();
                System.out.print(s:"Contact: ");
                contact = inp.nextLine();
                System.out.print(s:"Identification Card : ");
                identification_Card = inp.nextLine();
                System.out.print(s:"Email: ");
                email = inp.nextLine();
                System.out.print(s:"Age: ");
                age = inp.nextInt();
                inp.nextLine();// clear buffer
                jemaahID = "J" + (jemaah.size() + 1);
                System.out.print("Here your jemaah ID (for login purpose): " + jemaahID);
                System.out.println(x:"\n");

                Jemaah newJemaah = new Jemaah(name, contact, identification_Card, email, jemaahID, age, doktor.get(randomNumberGenerator.nextInt(bound:2)),
                                                pegawai.get(randomNumberGenerator.nextInt(bound:2)));

                jemaah.add(newJemaah);

        }catch (InputMismatchException e){
                System.out.println(x:"Invalid input, please enter the correct data.\n");
                inp.nextLine();
        }
        break;
}
```

HajiUmrahApp.java

SYSTEM DEMONSTRATION

THANK YOU