Chapter 1- Introduction

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.

System designing in terms of software engineering has its own value and importance in the system development process as a whole. To mention it may though seem as simple as anything or simply the design of systems, but in a broader sense it implies a systematic and rigorous approach to design such a system which fulfils all the practical aspects including flexibility, efficiency and security.

Before there is any further discussion of system design, it is important that some points be made clear. As it goes without saying that nothing is created that is not affected by the world in which it's made. So, the systems are not created in a vacuum.

They are created in order to meet the needs of the users. They are not only intended to solve the existing problems, but they also come up with acceptable solutions to the problems that may arise in the future. The whole process of system development, from blueprint to the actual product, involves considering all the relevant factors and taking the required specifications and creating a useful system based on strong technical, analytical and development skills of the professionals.

Let's get back to our discussion about what the system design phase is and the importance of system design in the process of system development. Being another important step in the system development process, system designing phase commences after the system analysis phase is completed. It's appropriate to mention that the output or the specifications taken through the phase of system analysis become an input in the system design phase which in turn leads to workout based on the user defined estimations.

The importance of this phase may be understood by reason of the fact that it involves identifying data sources, the

nature and type of data that is available. For example, in order to design a salary system, there is a need for using

inputs, such as, attendance, leave details, additions or deductions etc. This facilitates understanding what kind of

data is available and by whom it is supplied to the system so that the system may be designed considering all the

relevant factors. In addition, system designing leads to ensure that the system is created in such a way that it fulfils

1.1 Project Overview:

the overview provides a succinct summary of the "Tourism Management System" project, encapsulating its core aspects and objectives. It delineates the project's aim to streamline tourism-related activities through a comprehensive software solution. The key components, including user authentication, package and hotel management, destination exploration, and a centralized dashboard, are highlighted. Additionally, it emphasizes the target audience, comprising both tourists and administrators, and underscores the utilization of Java Swing, JDBC, and MySQL for development. The overview concludes by outlining potential future enhancements aimed at improving user experience and expanding system capabilities. Overall, the overview offers a concise yet informative glimpse into the project's scope, functionality, and technological underpinnings.

1.1.1 Mission and Vision:

Our mission at the Tourism Management System project is to redefine tourism management by offering an intuitive platform that empowers travelers to seamlessly plan and book their adventures. We envision a future where travelers can effortlessly explore destinations, discover unique experiences, and make informed decisions with ease. By integrating with hotels, destinations, and other travel services, our vision is to simplify the entire travel process and enhance the overall experience for tourists, setting new standards for efficiency, accessibility, and innovation in the industry.

1.1.2 Scope and Objectives:

The scope of the Tourism Management System project encompasses the development of a comprehensive software solution that facilitates various aspects of tourism management. This includes features such as user authentication, personal details management, package and hotel booking, destination exploration, and a centralized dashboard for comprehensive management. The system will provide functionality for both tourists, enabling them to plan and book their trips, and administrators, allowing them to manage user data, packages, hotels, and destinations efficiently.

- Develop a user-friendly interface for tourists to easily access and utilize the system's features.
- Implement secure user authentication mechanisms to protect user data and ensure privacy.

- Provide functionality for tourists to manage their personal details, preferences, and booking history.
- Enable tourists to explore available travel packages, view detailed itineraries, and book desired packages seamlessly.

1.1.3 Key Features and Functionality

The Tourism Management System project offers a range of key features and functionalities aimed at enhancing the tourism experience for both tourists and administrators:

- User Authentication and Management
- Package Management
- Hotel Booking:
- Additional Tools and Utilities

1.1.4 Technological Stack:

The aims of Projecyt is to create a robust and efficient software solution that meets the needs of both tourists and administrators in managing various aspects of tourism effectively.

- **Java Swing:** Utilized for developing the graphical user interface (GUI) components, providing a robust and platform-independent interface for users.
- **JDBC (Java Database Connectivity):** Employed for database connectivity, enabling interaction with the backend MySQL database to store and retrieve data efficiently.
- **MySQL Database:** Backend storage solution for user data, package information, hotel details, destination content, and other system data, ensuring reliability and scalability.
- **Multithreading:** Implemented for managing concurrent processes, such as loading screens and slideshow animations, enhancing user experience and system performance.
- **External APIs (Optional Future Enhancement):** Integration with external services for additional functionalities, such as mapping services for destination exploration or weather APIs for real-time updates, enhancing the overall user experience.

1.1.5 Project Timeline and Milestones:

The project will be executed in multiple phases, with each phase focusing on the development and integration of specific modules and functionalities. Key milestones.include

- Requirement Analysis
- Design and Planning
- Implementation
- Testing and Quality Assurance
- Training and Documentation

1.2 Software Requirements:

The Tourism Management System relies on a set of essential software components for its development and functionality. Compatibility across major operating systems ensures accessibility, while the Java Development Kit (JDK) enables programming in Java, the system's primary language. Integrated Development Environments (IDEs) like Eclipse or IntelliJ IDEA facilitate efficient coding and project management. The MySQL Database Management System serves as the backend for data storage, necessitating the MySQL JDBC Driver for connectivity. Additionally, optional components such as graphics libraries and web browsers may enhance visual elements or support web integration. Documentation tools, version control systems, and project management tools further streamline development processes. These software requirements collectively underpin the creation of a robust and user-friendly Tourism Management System.

- **Operating System:** The system should be compatible with major operating systems such as Windows, macOS, and Linux to ensure broad accessibility.

- **Java Development Kit (JDK):** JDK version 8 or higher is required for development to utilize Java programming language features and libraries.
- **Integrated Development Environment (IDE):** An IDE such as Eclipse, IntelliJ IDEA, or NetBeans is recommended for coding, debugging, and managing project files efficiently.
- **MySQL Database Management System:** MySQL Server should be installed to serve as the backend database for storing user data, package details, hotel information, destination content, and other system data.
- **MySQL JDBC Driver:** JDBC driver for MySQL should be included in the project's dependencies to enable connectivity and interaction with the MySQL database.
- **Graphics Library (Optional):** If implementing advanced graphical features, a graphics library such as JavaFX or OpenGL may be required to enhance the visual aspects of the user interface.

1.3 Hardware Requirements:

- System type: 64-bit Operating System, x64-bassed processor.

- installed memory (RAM):8.00 GB (7.43 GB Usable)
- Total size of Hard disk: 1 TB

Chapter 2- Limitation of Existing System

The existing tourism management systems face several limitations that impede their effectiveness. These include compatibility issues, scalability challenges, reliance on outdated technology, security concerns, limited functionality, poor user experience, and inadequate support services. These shortcomings hinder accessibility, compromise data security, and lead to inefficiencies in system operations. Addressing these limitations through the development of an updated and more user-friendly system is crucial for enhancing user satisfaction, improving system performance, and ensuring competitiveness in the tourism industry.

1. **Limited Accessibility:** Some systems may not be accessible across various devices and operating systems, restricting user access and usability.

2. **Scalability Challenges:** Systems may struggle to handle increased loads during peak periods, leading to performance issues or system downtime.

3. **Outdated Technology:** Reliance on outdated technologies may limit system capabilities and hinder integration with modern features and services.

4. **Security Vulnerabilities:** Inadequate security measures or vulnerabilities in the system architecture may expose user data to risks such as unauthorized access, data breaches, or cyberattacks.

5. **Limited Functionality:** Some systems may lack essential features or functionalities, reducing their effectiveness in meeting user requirements and industry standards.

6. **Poor User Experience:** Complex interfaces, slow response times, or cumbersome processes can result in a poor user experience, leading to frustration and reduced user adoption.

7. **Lack of Customization:** Limited customization options may restrict users' ability to tailor the system to meet their specific needs or preferences.

8. **Inadequate Support and Maintenance:** Insufficient support services or lack of regular maintenance may result in prolonged downtime, unresolved issues, or difficulty in adapting to changing requirements.

Chapter 3 – System Analysis

System analysis involves a comprehensive examination of an existing system or the requirements for a new system to identify its objectives, functionalities, processes, and constraints. This process typically includes the following steps:

3.1 Requirement Analysis:

Requirement analysis is a crucial phase in the software development process, encompassing the gathering, documentation, and validation of requirements for a software system. This process involves several key steps:

3.1.1 Requirement Elicitation:

This involves gathering requirements from stakeholders, including users, customers, and domain experts. Techniques such as interviews, surveys, workshops, and brainstorming sessions are used to elicit requirements effectively.

3.1.2 Requirement Documentation: Once requirements are gathered, they need to be documented in a clear, concise, and unambiguous manner. This documentation typically includes functional requirements (what the system should do) and non-functional requirements (qualities the system should possess, such as performance, security, and usability)

3.1.3 Requirement Analysis: During this phase, gathered requirements are analyzed to ensure consistency, completeness, and feasibility. Conflicting or ambiguous requirements are identified and resolved through discussions with stakeholders

3.1.4 Requirement Prioritization: Not all requirements are of equal importance. Therefore, prioritizing requirements based on their importance and urgency is essential. Techniques such as MoSCoW (Must have, Should have, Could have, Won't have) are often used for prioritization.

3.1.5 Requirement Validation: Once requirements are documented and prioritized, they need to be validated to ensure they accurately reflect the stakeholders' needs and expectations. Validation techniques such as reviews, walkthroughs, and prototyping are used to confirm that the requirements are correct and complete.

3.1.6 Requirement Management:

Requirements are subject to change throughout the software development lifecycle due to evolving business needs, technological advancements, or changes in regulations. Therefore, a robust requirement management process is essential to track changes, assess their impact, and ensure that the system remains aligned with stakeholders' needs.

3.2 Technical Feasibility:

Technical feasibility assesses the practicality of implementing a proposed system or project from a technical perspective. It evaluates whether the proposed solution can be developed using the available technology, resources, and expertise. Here are the key aspects of technical feasibility analysis:

3.2.1 Technology Assessment: Evaluate the availability and suitability of technology platforms, tools, and frameworks required for implementing the proposed system. Consider factors such as compatibility, scalability, and performance.

3.2.2 Resource Evaluation: Assess the availability of technical resources, including hardware, software, infrastructure, and human resources (such as developers, engineers, and technical experts). Determine if the necessary resources are accessible and if any additional resources need to be acquired or trained

3.2.3 Expertise and Skills: Evaluate the technical expertise and skills required for developing, implementing, and maintaining the proposed system. Determine if the existing team possesses the necessary skills or if additional training or hiring is necessary.

3.2.4 Integration Compatibility: Assess the compatibility of the proposed system with existing systems, databases, and infrastructure. Determine if the new system can seamlessly integrate with the existing technology ecosystem without significant disruptions or modifications.

3.3 Operation Feasibility:

The user authentication process, a critical aspect of any software system, appears feasible within the tourism management application. Validating user credentials against a MySQL database is a common practice, though ensuring robust security measures like password hashing and protection against SQL injection is imperative.

3.3.1 User Authentication:

- **Feasibility Assessment**: Feasible
- **Details:** User authentication is a common and essential operation in software systems. Verifying user credentials against a database is a standard practice. However, ensuring

security measures such as password hashing and protection against SQL injection is crucial for robust authentication.

3.3.2 Loading Animation:

- **Feasibility Assessment:** Feasible
- **Details**: Displaying a loading animation provides visual feedback to users during application initialization or processing tasks. It enhances user experience by indicating that the system is active. Implementing loading animations is straightforward and feasible using Java Swing components like JProgressBar.

3.3.3 Slideshow Display (Hotels and Destinations):

- **Feasibility Assessment:** Feasible
- **Details**: Slideshow displays of images are common in graphical applications. Implementing slideshows for showcasing hotels and destinations involves cycling through images at predefined intervals. This operation is feasible using Java's javax.swing components to manage image display and transitions.

3.3.4 Database Connectivity:

- **Feasibility Assessment :** Feasible
- **Details :** Establishing database connectivity using JDBC (Java Database Connectivity) is a well-supported feature in Java. Connecting to a MySQL database, executing SQL queries, and processing results are standard operations. The provided code demonstrates feasibility by initializing a connection to a MySQL database using JDBC.

3.3.5 User Interface Layout:

- **Feasibility Assessment :** Feasible with Considerations
- **Details :** Defining the user interface layout using absolute positioning (null layout) is feasible but may have limitations. While it provides precise control over component placement, it lacks responsiveness to different screen sizes and resolutions. Consideration should be given to implementing more flexible layout managers for improved usability across various devices.

Chapter 4 - Conceptual Modules

4.1 Data Flow Diagram :

Data Flow Diagrams (DFDs) provide a visual representation of how data moves through the Tribe webapp. These diagrams help us understand the flow of information and processes within the system. DFDs are instrumental in modelling the interactions between different components and the overall architecture of the app.

4.1.1 Context Diagram (Level 0 DFD) :

The context diagram presents an overview of the Imagin app's interactions with external entities such as users, posts and databases. It will illustrate the high-level flow of data and processe

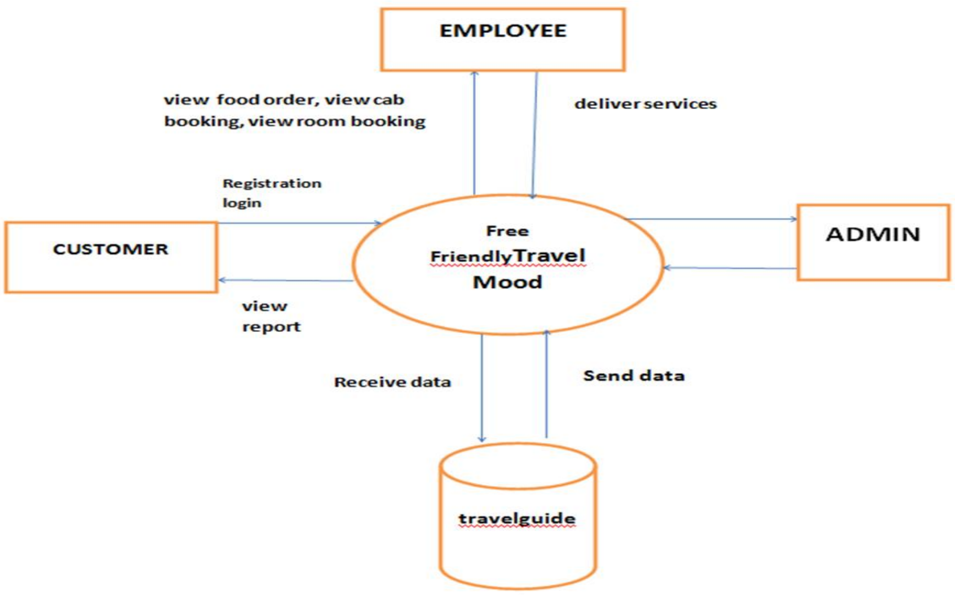


Fig 1: Context diagram (Level 0 DFD) of the Tribe app explaining the data flow in a broader sense

4.1.2 Level 1 DFD:

This diagram breaks down the system into major processes and data stores within the app, showing how they interact. It will provide a more detailed view of the app's unctioning, ranging from the steps involved in authentication and authorization of user n the app and how form validation is done to the process of creation of posts and browsing through the blogs and spaces

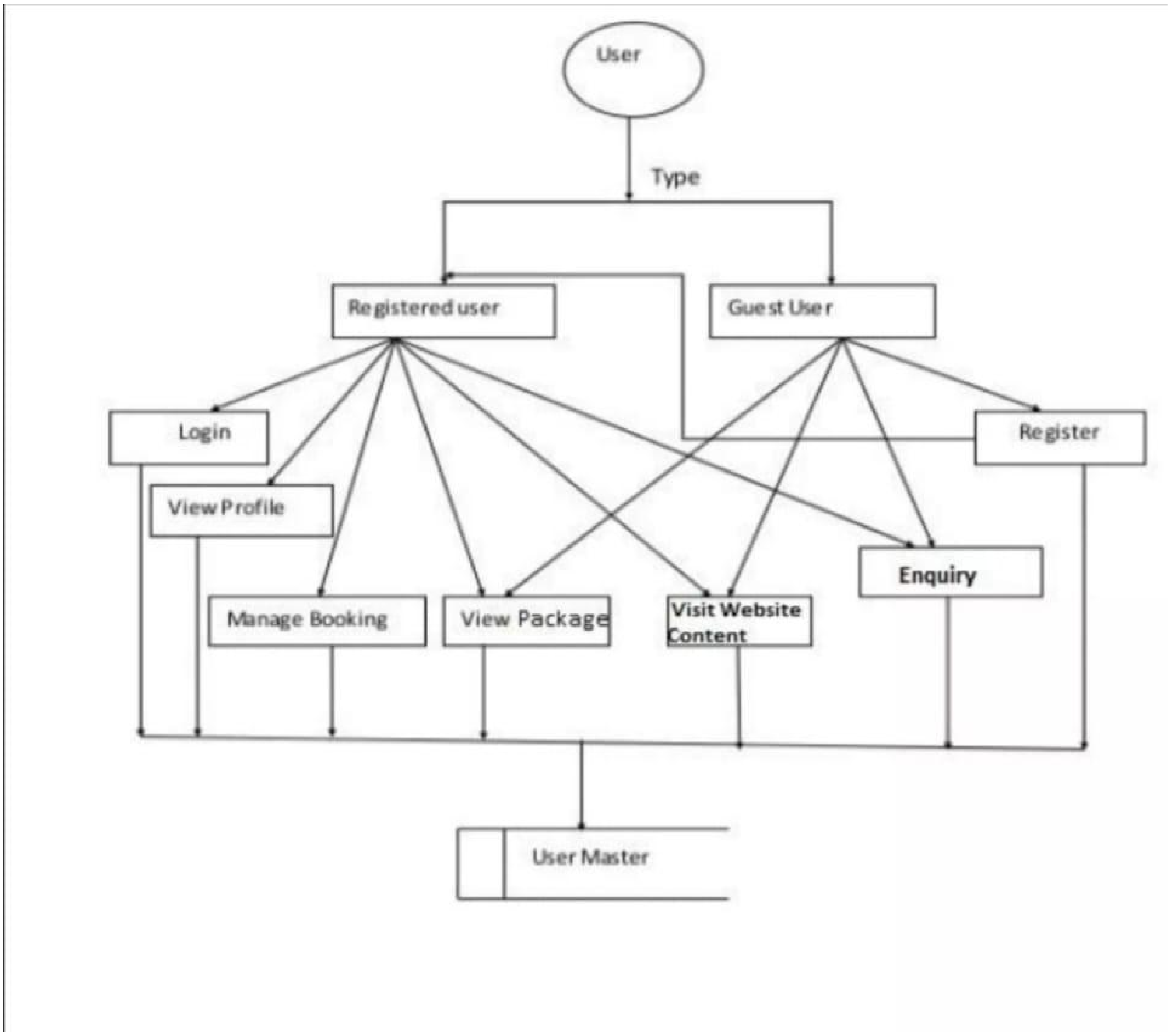


Fig 2: Level 1 DFD showing detailed flow of the data within the app during each and every service provided by the app.

There are three major processes in the tribe Web app namely authentication, blog posts and spaces. The data flow during authentication can be understood by the upper block which depicts how a user can login or sign up for which the data is validated from the back end and if the credentials are successfully verified in authorized user is returned to the front end this authorized user can now create or browse through blog which are fetched through the blog storage and presented to the user. Authorized user can also create a space where he becomes the admin, he can now add other members to this space or post a thread the space anonymously this thread gets stored in the thread storage which any member of the space can fetch and see they can also like and reply to this thread

4.2 Entity-Relationship Diagram :

Entity-Relationship diagrams visually represent the data model and relationships within the Imagin app. They are instrumental in defining the app's data structure.

➢ Entities:

We'll identify and define the core entities in the system. In this proposed model of ours ,there are two fundamental entities :

• User

• Blogs

• Spaces

• Threads

➢ Relationships:

This outlines relationships between these entities, such as how users create images, interact with each other, and how prompts are associated with generated images.

➢ Attributes:

For each entity, we'll detail the specific attributes or properties that are associated with them, providing insights into the information store

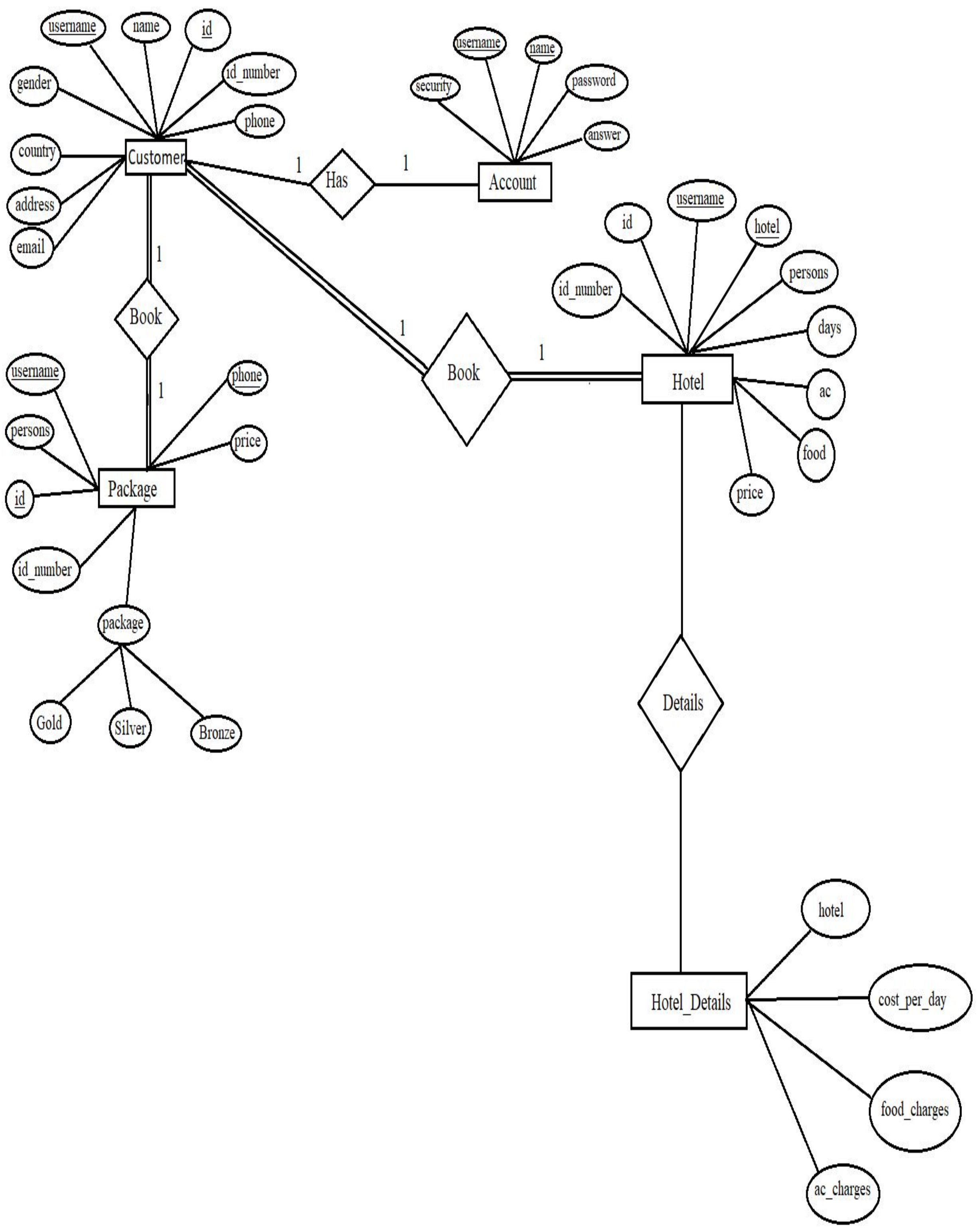


Fig 3: Entity-Relationship diagram explaining how the proposed system will have interdependency between entities.

- User ID
- UserName
- Password:
- Email Address:
- Full Name:
- Date of Birth:
- Address:
- Phone Number:
- Payment Information:
- Preferences:
- Booking History:
- Role or Access Leve:

The attributes for Blog entity are as follows :

- Author
- Content
- Publication Date
- Last Updated
- Tags or Categories
- Comments
- Likes or Reactions
- Visibility
- Featured Image
- URL or Permalink

The attributes for Space entity are as follows

- Space ID
- Name
- Description
- Location
- Capacity
- Amenities
- Availability
- Price

The attributes for Post entity are as follows:

- Post ID
- Title
- Author
- Content
- Publication Date
- Last Updated
- Tags or Categories
- Comments

These conceptual modules and diagrams help us better understand and illustrate the system level design and architecture of our app by the visualizing the relationship among the entities and the actual process of data flow during the execution of each and every microservice. As in this case, the user entity can create a number of blogs which is stored as a reference. Similarly, the space entity stores the reference of all its members (users). Post contains its comments which in turn reference to other posts. A blog also contains author (user) and likes (array of users)

Chapter 5 - Proposed System

5.1 Overview:

The proposed system aims to prioritize user experience by offering a seamless, intuitive, and personalized interface. Users will have access to a comprehensive set of features designed to meet their specific needs and preferences. The system will focus on enhancing user engagement, facilitating easy navigation, and providing efficient access to relevant information and services. With robust authentication and authorization mechanisms, users will experience enhanced security and privacy protection. Additionally, the proposed system will offer extensive customization options, allowing users to tailor their experience according to their preferences. Overall, the user-centric approach of the proposed system aims to ensure satisfaction, efficiency, and enjoyment for all users.

5.2 Module Description:

The module description section provides an overview of the various modules or components within the system, outlining their functionalities and interactions. Each module is described in detail, highlighting its purpose, features, and relationship with other modules. This section serves as a roadmap for understanding the system's architecture and functionality, helping stakeholders grasp how different parts of the system work together to achieve the overall objectives.In this section, each module of the system is comprehensively outlined, detailing its specific functionalities and interactions with other modules. The description includes a breakdown of tasks or operations performed by each module, along with any dependencies or interdependencies with other system components. Additionally, the module description may encompass information on inputs, outputs, data flow, and processing logic within each module. This detailed overview aids in understanding the system's architecture and operation, facilitating effective development, testing, and maintenance processes.

The Java code you provided seems to be for a tourism management system application. Let's break down the modules and their descriptions:

1. **Splash Screen Module**:
    - **Splash.java**: This module displays a splash screen when the application starts, providing a visually appealing introduction to the system.
2. **User Authentication Module**:

- **Login.java**: Manages user authentication by providing a login interface. It connects to a MySQL database to verify user credentials and allows users to log in to the system.

3. **Loading Module**:
    - **Loading.java**: Displays a loading screen while the application is processing tasks, providing visual feedback to users during operations that may take some time.
4. **Hotel Management Module**:
    - **CheckHotels.java**: This module displays a slideshow of hotel images, allowing users to view available hotels.
5. **Dashboard Module**:
    - **Dashboard.java**: The main dashboard of the application where users can access various functionalities such as adding personal details, checking packages, booking hotels, viewing destinations, and more.
6. **Destination Module**:
    - **Destination.java**: Displays a slideshow of destination images, allowing users to explore different travel destinations.
7. **Database Connectivity Module**:
    - **Conn.java**: Handles the connection to the MySQL database. It provides functionality for establishing a connection and executing SQL queries.

5.3 Algorithm Used

- **Step 1**: User enters their username and password in the login interface (`Login.java`).

- **Step 2**: When the "Login" button is clicked, an action event triggers the authentication process.
- **Step 3**: The application retrieves the entered username and password from the text fields.
- **Step 4**: It constructs an SQL query to select records from the database where the username and password match the entered values.
- **Step 5**: The application connects to the MySQL database using the `Conn` class and executes the SQL query.
- **Step 6**: If the query returns a result (i.e., the username and password match), the user is granted access to the system, and the dashboard (`Dashboard.java`) is displayed.
- **Step 7**: If the query returns no result, indicating invalid credentials, the application displays an error message.

Chapter 6-Source Code and Output

The source code for this project Tourism managment System.

Splash.java

```
package tourismmanagementsystem;

import javax.swing.*;
import java.awt.*;

public class Splash {
   public static void main(String[] args) {
      SplashFrame frame=new SplashFrame();
      frame.setVisible(true);
      int x=1;
      for (int i=1;i<=550;i+=9,x+=7) {
         frame.setLocation(650-(x+i)/2,370-(i/2));
         frame.setSize(x+i, i);
         try{
            Thread.sleep(10);
         }catch (Exception e){

         }
      }
      frame.setVisible(true);
   }
}

class SplashFrame extends JFrame implements Runnable {
   Thread t1;
   SplashFrame() {
      ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("tourismmanagementsystem/icons/splash.jpg"));
      Image i2=i1.getImage().getScaledInstance(1000,550,Image.SCALE_DEFAULT);
      ImageIcon i3=new ImageIcon(i2);
      JLabel l1=new JLabel(i3);
      add(l1);
      setUndecorated(true);

      t1=new Thread(this);
      t1.start();
   }
   public void run() {
      try {
         Thread.sleep(5000);
         this.setVisible(false);
         new Login().setVisible(true);
      } catch (Exception e) {

      }

   }
}
```

Login.java

```
package tourismmanagementsystem;

import javax.swing.*;
import javax.swing.border.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class Login extends JFrame implements ActionListener{
 JButton b1,b2,b3;
 JTextField t1;
 JPasswordField t2;
      Login(){
      setBounds(220,130,900,400);
      getContentPane().setBackground(Color.white);
       setLayout(null);

       JPanel p1=new JPanel();
       p1.setBackground(new Color(131,193,233));
       p1.setBounds(0,0,400,400);
       p1.setLayout(null);
       add(p1);

       ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("tourismmanagementsystem/icons/login.png"));
       Image i2=i1.getImage().getScaledInstance(200,200,Image.SCALE_DEFAULT);
       ImageIcon i3=new ImageIcon(i2);
       JLabel l1=new JLabel(i3);
       l1.setBounds(100,80,200,200);
       p1.add(l1);

       JPanel p2=new JPanel();
       p2.setBounds(400,30,460,300);
       p2.setLayout(null);
       add(p2);

       JLabel l2=new JLabel("Username");
       l2.setBounds(60,20,200,25);
       l2.setFont(new Font("SAN_SERIF",Font.PLAIN,20));
       p2.add(l2);

       t1=new JTextField();
       t1.setBounds(60,60,300,30);
       t1.setBorder(BorderFactory.createEmptyBorder());
       p2.add(t1);

       JLabel l3=new JLabel("Password");
       l3.setBounds(60,100,200,25);
       l3.setFont(new Font("SAN_SERIF",Font.PLAIN,20));
       p2.add(l3);

       t2=new JPasswordField();
       t2.setBounds(60,140,300,30);
       t2.setBorder(BorderFactory.createEmptyBorder());
       p2.add(t2);

       b1=new JButton("Login");
       b1.setFont(new Font("SAN_SERIF",Font.PLAIN,20));
       b1.setBackground(new Color(131,193,233));
       b1.setForeground(Color.white);
       b1.setBorder(BorderFactory.createEmptyBorder());
       b1.setBounds(60,200,150,30);
       b1.addActionListener(this::actionPerformed);
       p2.add(b1);

       b2=new JButton("Signup");
```

```
        b2.setFont(new Font("SAN_SERIF",Font.PLAIN,20));
        b2.setForeground(new Color(131,193,233));
        b2.setBackground(Color.white);
        b2.setBorder(new LineBorder(new Color(131,193,233)));
        b2.setBounds(230,200,150,30);
        b2.addActionListener(this::actionPerformed);
        p2.add(b2);

        b3=new JButton("Forget Password");
        b3.setFont(new Font("SAN_SERIF",Font.PLAIN,20));
        b3.setForeground(new Color(131,193,233));
        b3.setBackground(Color.white);
        b3.setBorder(new LineBorder(new Color(131,193,233)));
        b3.setBounds(130,250,200,30);
        b3.addActionListener(this::actionPerformed);
        p2.add(b3);

        JLabel l4=new JLabel("Trouble in Login...");
        l4.setForeground(Color.RED);
        l4.setBounds(340,250,150,20);
        p2.add(l4);

//        setUndecorated(true);
        setVisible(true);
    }

 public void actionPerformed(ActionEvent ae) {
     if (ae.getSource() == b1) {
         try{
             String username=t1.getText();
             String password=t2.getText();
             String sql="Select * from accounts where username = '"+username+"' And password = '"+password+"'";
             Conn c=new Conn();
             ResultSet rs=c.s.executeQuery(sql);
             if(rs.next()){
                 this.setVisible(false);
                 new Loading(username).setVisible(true);
             }else{
                 JOptionPane.showMessageDialog(null,"Invalid Login");
             }
         }catch(Exception e){}
     }else if (ae.getSource() == b2)
     {
      this.setVisible(false);
      new Signup().setVisible(true);
     }
     else if(ae.getSource() == b3)
     {
      this.setVisible(false);
      new ForgetPassword().setVisible(true);
     }
     }
    public static void main(String args[]){

        new Login();
    }

}
```

Loading.java

```
package tourismmanagementsystem;
```

```
import java.awt.*;
import javax.swing.*;

public class Loading extends JFrame implements Runnable{
    JProgressBar p;
    Thread t;
     String username;

    public void run(){
        try{
          for(int i=1;i<=101;i++){
              int m=p.getMaximum();
              int n=p.getValue();
              if(n<m){
                  p.setValue(p.getValue() + 1);
              }
              else{
                  i=101;
                  setVisible(false);
                  new Dashboard(username).setVisible(true);
              }
              Thread.sleep(50);
          }
        }
        catch(Exception e){

        }
    }
    Loading(String user){
        username=user;
        t=new Thread(this);
        setBounds(400,150,500,400);
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);
     JLabel l1=new JLabel("Travel And Tourism Application");
     l1.setBounds(50,20,600,40);
     l1.setFont(new Font("Raleway",Font.BOLD,25));
     l1.setForeground(Color.blue);
     add(l1);

     p= new JProgressBar();
     p.setStringPainted(true);
     p.setBounds(100,80,300,25);
     add(p);

     JLabel l2=new JLabel("Please Wait...");
     l2.setBounds(200,110,100,25);
     l2.setFont(new Font("Tahoma",Font.BOLD,14));
     l2.setForeground(Color.red);
     add(l2);

     JLabel l3=new JLabel("Welcome " + username);
     l3.setBounds(20,320,600,25);
     l3.setFont(new Font("Tahoma",Font.BOLD,16));
     l3.setForeground(Color.red);
     add(l3);

        t.start();
    }
    public static void main(String[] args) {
        new Loading("").setVisible(true);
    }
}
```

CheckHotels.java

```
package tourismmanagementsystem;
import javax.swing.*;
import java.awt.*;
public class CheckHotels extends JFrame implements Runnable{
    Thread t1;
    JLabel l1,l2,l3,l4,l5,l6,l7,l8,l9,l10;
    JLabel[] label =new JLabel[]{l1,l2,l3,l4,l5,l6,l7,l8,l9,l10};
    JLabel caption;
    public void run(){
        String[] text =new String[]{"Jw Marriott Hotel","Mandarin Oriental Hotel","Four Seasons Hotel","Radisson
Hotel","Classio Hotel","The Bay Club Hotel","Breeze Blows Hotel","Quick Stop Hotel","Quick Stop Hotel","Happy
Mornings Motel","Moss View Hotel"};
        try{
        for(int i=0;i<=9;i++){
            this.label[i].setVisible(true);
            caption.setText(text[i]);
            this.label[i].add(caption);
            Thread.sleep(2800);
            this.label[i].setVisible(false);
        }
        }catch(Exception e){

        }
          }

    CheckHotels(){
        setBounds(250,75,700,550);

        ImageIcon i1=null,i2=null,i3=null,i4=null,i5=null,i6=null,i7=null,i8=null,i9=null,i10=null;
        ImageIcon[] image=new ImageIcon[]{i1,i2,i3,i4,i5,i6,i7,i8,i9,i10};

        Image j1=null,j2=null,j3=null,j4=null,j5=null,j6=null,j7=null,j8=null,j9=null,j10=null;
        Image[] jimage=new Image[]{j1,j2,j3,j4,j5,j6,j7,j8,j9,j10};

        ImageIcon i11=null,i12=null,i13=null,i14=null,i15=null,i16=null,i17=null,i18=null,i19=null,i20=null;
        ImageIcon[] iimage=new ImageIcon[]{i11,i12,i13,i14,i15,i16,i17,i18,i19,i20};

        caption=new JLabel();
        caption.setBounds(50,420,1000,70);
        caption.setForeground(Color.WHITE);
        caption.setFont(new Font("Tahoma", Font.PLAIN,30));

        for(int i=0;i<=9;i++){
        image[i]=new
ImageIcon(ClassLoader.getSystemResource("tourismmanagementsystem/icons/hotel"+(i+1)+".jpg"));
        jimage[i]=image[i].getImage().getScaledInstance(700,550,Image.SCALE_DEFAULT);
        iimage[i]=new ImageIcon(jimage[i]);
        this.label[i]=new JLabel(iimage[i]);
        this.label[i].setBounds(0,0,700,550);
        add(this.label[i]);
        }

        t1=new Thread(this);
        t1.start();

    }
    public static void main(String[] args) {
        new CheckHotels().setVisible(true);
    }
}
```

Dashboard.java

```
package tourismmanagementsystem;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Dashboard extends JFrame implements ActionListener{
    JButton b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14;
    String username;
    Dashboard(String username){
        this.username=username;
       setExtendedState(JFrame.MAXIMIZED_BOTH);
       setLayout(null);

       JPanel p1=new JPanel();
       p1.setBounds(0,0,1950,65);
       p1.setLayout(null);
       p1.setBackground(new Color(0,0,102));
       add(p1);

       JPanel p2=new JPanel();
       p2.setBounds(0,65,250,1000);
       p2.setLayout(null);
       p2.setBackground(new Color(0,0,102));
       add(p2);

       b1=new JButton("Add Personal Details");
       b1.setBackground(new Color(0,0,102));
       b1.setFont(new Font("Tahoma",Font.PLAIN,15));
       b1.setForeground(Color.white);
       b1.setMargin(new Insets(0,0,0,80));
       b1.setBounds(0,0,250,35);
       b1.addActionListener(this);
       p2.add(b1);

       b2=new JButton("Update Personal Details");
       b2.setBackground(new Color(0,0,102));
       b2.setFont(new Font("Tahoma",Font.PLAIN,15));
       b2.setForeground(Color.white);
       b2.setMargin(new Insets(0,0,0,60));
       b2.setBounds(0,30,250,35);
       b2.addActionListener(this);
       p2.add(b2);

       b3=new JButton("View Details");
       b3.setBackground(new Color(0,0,102));
       b3.setFont(new Font("Tahoma",Font.PLAIN,15));
       b3.setForeground(Color.white);
       b3.setMargin(new Insets(0,0,0,135));
       b3.setBounds(0,60,250,35);
       b3.addActionListener(this);
       p2.add(b3);

       b4=new JButton("Delete Personal Details");
       b4.setBackground(new Color(0,0,102));
       b4.setFont(new Font("Tahoma",Font.PLAIN,15));
```

```
b4.setForeground(Color.white);
b4.setMargin(new Insets(0,0,0,65));
b4.setBounds(0,90,250,35);
p2.add(b4);

b5=new JButton("Check Package");
b5.setBackground(new Color(0,0,102));
b5.setFont(new Font("Tahoma",Font.PLAIN,15));
b5.setForeground(Color.white);
b5.setMargin(new Insets(0,0,0,120));
b5.setBounds(0,120,250,35);
b5.addActionListener(this);
p2.add(b5);

b6=new JButton("Book Package");
b6.setBackground(new Color(0,0,102));
b6.setFont(new Font("Tahoma",Font.PLAIN,15));
b6.setForeground(Color.white);
b6.setMargin(new Insets(0,0,0,125));
b6.setBounds(0,150,250,35);
b6.addActionListener(this);
p2.add(b6);

b7=new JButton("View Package");
b7.setBackground(new Color(0,0,102));
b7.setFont(new Font("Tahoma",Font.PLAIN,15));
b7.setForeground(Color.white);
b7.setMargin(new Insets(0,0,0,125));
b7.setBounds(0,180,250,35);
b7.addActionListener(this);
p2.add(b7);

b8=new JButton("View Hotels");
b8.setBackground(new Color(0,0,102));
b8.setFont(new Font("Tahoma",Font.PLAIN,15));
b8.setForeground(Color.white);
b8.setMargin(new Insets(0,0,0,140));
b8.setBounds(0,210,250,35);
b8.addActionListener(this);
p2.add(b8);

b9=new JButton("Book Hotel");
b9.setBackground(new Color(0,0,102));
b9.setFont(new Font("Tahoma",Font.PLAIN,15));
b9.setForeground(Color.white);
b9.setMargin(new Insets(0,0,0,145));
b9.setBounds(0,240,250,35);
b9.addActionListener(this);
p2.add(b9);

b10=new JButton("View Booked Hotel");
b10.setBackground(new Color(0,0,102));
b10.setFont(new Font("Tahoma",Font.PLAIN,15));
b10.setForeground(Color.white);
b10.setMargin(new Insets(0,0,0,90));
b10.setBounds(0,270,250,35);
b10.addActionListener(this);
p2.add(b10);

b11=new JButton("Destinations");
b11.setBackground(new Color(0,0,102));
b11.setFont(new Font("Tahoma",Font.PLAIN,15));
b11.setForeground(Color.white);
```

```
b11.setMargin(new Insets(0,0,0,130));
b11.setBounds(0,300,250,35);
b11.addActionListener(this);
p2.add(b11);

b12=new JButton("Payment");
b12.setBackground(new Color(0,0,102));
b12.setFont(new Font("Tahoma",Font.PLAIN,15));
b12.setForeground(Color.white);
b12.setMargin(new Insets(0,0,0,155));
b12.setBounds(0,330,250,35);
p2.add(b12);

b13=new JButton("Calculator");
b13.setBackground(new Color(0,0,102));
b13.setFont(new Font("Tahoma",Font.PLAIN,15));
b13.setForeground(Color.white);
b13.setMargin(new Insets(0,0,0,150));
b13.setBounds(0,360,250,35);
b13.addActionListener(this);
p2.add(b13);

b14=new JButton("Notepad");
b14.setBackground(new Color(0,0,102));
b14.setFont(new Font("Tahoma",Font.PLAIN,15));
b14.setForeground(Color.white);
b14.setMargin(new Insets(0,0,0,155));
b14.setBounds(0,390,250,35);
b14.addActionListener(this);
p2.add(b14);

JButton b15=new JButton("About");
b15.setBackground(new Color(0,0,102));
b15.setFont(new Font("Tahoma",Font.PLAIN,15));
b15.setForeground(Color.white);
b15.setMargin(new Insets(0,0,0,165));
b15.setBounds(0,420,250,35);
p2.add(b15);

ImageIcon i4=new ImageIcon(ClassLoader.getSystemResource("tourismmanagementsystem/icons/dash.png"));
Image i5 =i4.getImage().getScaledInstance(70,70,Image.SCALE_DEFAULT);
ImageIcon i6=new ImageIcon(i5);
JLabel l2=new JLabel(i6);
l2.setBounds(0,0,70,70);
p1.add(l2);

JLabel l3=new JLabel("Dashboard");
l3.setFont(new Font("Tahome",Font.BOLD,30));
l3.setForeground(Color.white);
l3.setBounds(80,10,300,40);
p1.add(l3);

ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("tourismmanagementsystem/icons/home.jpg"));
Image i2 =i1.getImage().getScaledInstance(1950,1000,Image.SCALE_DEFAULT);
ImageIcon i3=new ImageIcon(i2);
JLabel l1=new JLabel(i3);
l1.setBounds(0,0,1950,1000);
add(l1);

JLabel l4=new JLabel("Travel And Tourism Management System");
l4.setBounds(400,80,1000,60);
l4.setForeground(Color.white);
l4.setFont(new Font("Tahoma",Font.PLAIN,40));
```

```
        l1.add(l4);

    }

    public void actionPerformed(ActionEvent ae){
        if(ae.getSource()==b13){
            try{
                Runtime.getRuntime().exec("calc.exe");
            }catch(Exception e){

            }
        }else if(ae.getSource()==b14){
            try{
                Runtime.getRuntime().exec("notepad.exe");
            }catch(Exception e){

            }
        }else if(ae.getSource()==b1){
            new AddCustomer(username).setVisible(true);
        }else if(ae.getSource()==b2){
            new UpdateCustomer(username).setVisible(true);
        }
        else if(ae.getSource()==b3){
            new ViewCustomer(username).setVisible(true);
        }else if(ae.getSource()==b5){
            new CheckPackage().setVisible(true);
        }else if(ae.getSource()==b6){
            new BookPackage(username).setVisible(true);
        }else if(ae.getSource()==b7){
            new ViewPackage(username).setVisible(true);
        }else if(ae.getSource()==b8){
            new CheckHotels().setVisible(true);
        }else if(ae.getSource()==b11){
            new Destination().setVisible(true);
        }else if(ae.getSource()==b9){
           new BookHotel(username).setVisible(true);
        }else if(ae.getSource()==b10){
            new ViewBookedHotel(username).setVisible(true);
        }
    }
    public static void main(String[] args){
        new Dashboard("").setVisible(true);
    }
}
```

Destination.java

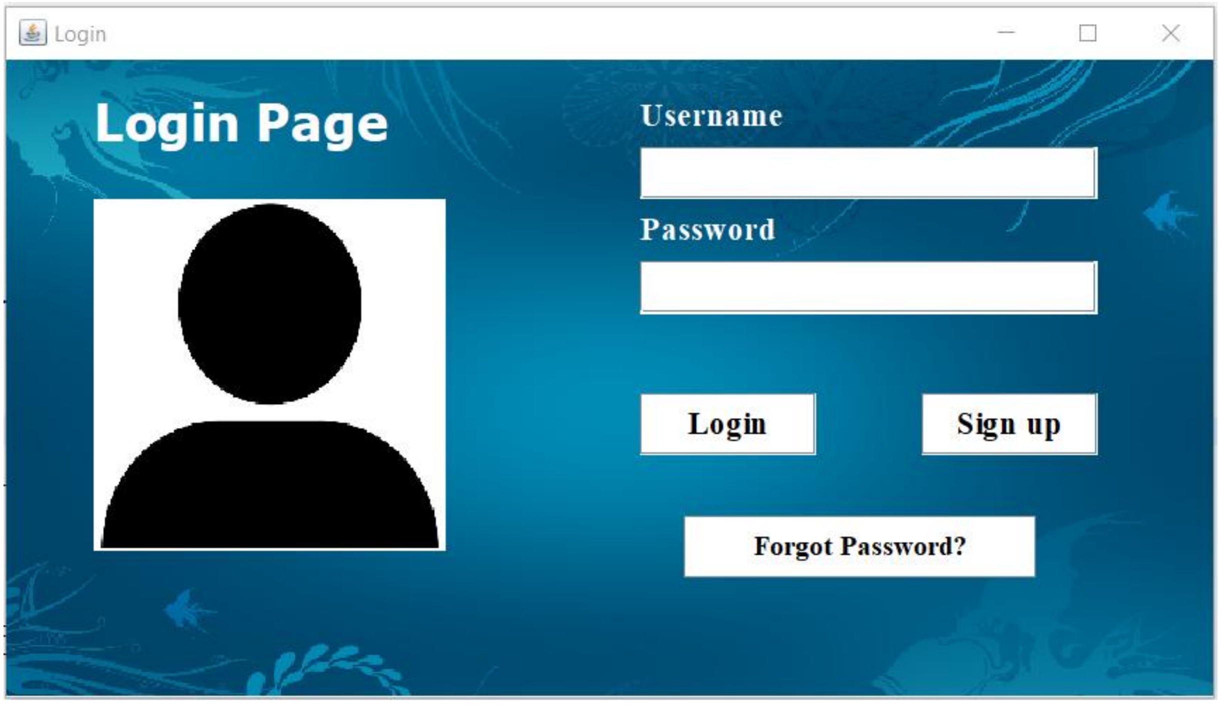
```
package tourismmanagementsystem;
import javax.swing.*;
import java.awt.*;
public class Destination extends JFrame implements Runnable{
    Thread t1;
    JLabel l1,l2,l3,l4,l5,l6,l7,l8,l9,l10;
    JLabel[] label =new JLabel[]{l1,l2,l3,l4,l5,l6,l7,l8,l9,l10};
    public void run(){
        String[] text =new String[]{"Jw Marriott Hotel","Mandarin Oriental Hotel","Four Seasons
Hotel","Radisson Hotel","Classio Hotel","The Bay Club Hotel","Breeze Blows Hotel","Quick Stop
Hotel","Quick Stop Hotel","Happy Mornings Motel","Moss View Hotel"};
        try{
        for(int i=0;i<=9;i++){
            this.label[i].setVisible(true);
            Thread.sleep(2800);
            this.label[i].setVisible(false);
        }
        }catch(Exception e){

        }
        }

    Destination(){
        setBounds(250,75,700,550);

        ImageIcon i1=null,i2=null,i3=null,i4=null,i5=null,i6=null,i7=null,i8=null,i9=null,i10=null;
        ImageIcon[] image=new ImageIcon[]{i1,i2,i3,i4,i5,i6,i7,i8,i9,i10};

        Image j1=null,j2=null,j3=null,j4=null,j5=null,j6=null,j7=null,j8=null,j9=null,j10=null;
        Image[] jimage=new Image[]{j1,j2,j3,j4,j5,j6,j7,j8,j9,j10};

        ImageIcon i11=null,i12=null,i13=null,i14=null,i15=null,i16=null,i17=null,i18=null,i19=null,i20=null;
        ImageIcon[] iimage=new ImageIcon[]{i11,i12,i13,i14,i15,i16,i17,i18,i19,i20};

        for(int i=0;i<=9;i++){
        image[i]=new
ImageIcon(ClassLoader.getSystemResource("tourismmanagementsystem/icons/dest"+(i+1)+".jpg"));
        jimage[i]=image[i].getImage().getScaledInstance(700,550,Image.SCALE_DEFAULT);
        iimage[i]=new ImageIcon(jimage[i]);
        this.label[i]=new JLabel(iimage[i]);
        this.label[i].setBounds(0,0,700,550);
        add(this.label[i]);
        }

        t1=new Thread(this);
        t1.start();

    }
    public static void main(String[] args) {
        new Destination().setVisible(true);
    }
}
```

Conn.java

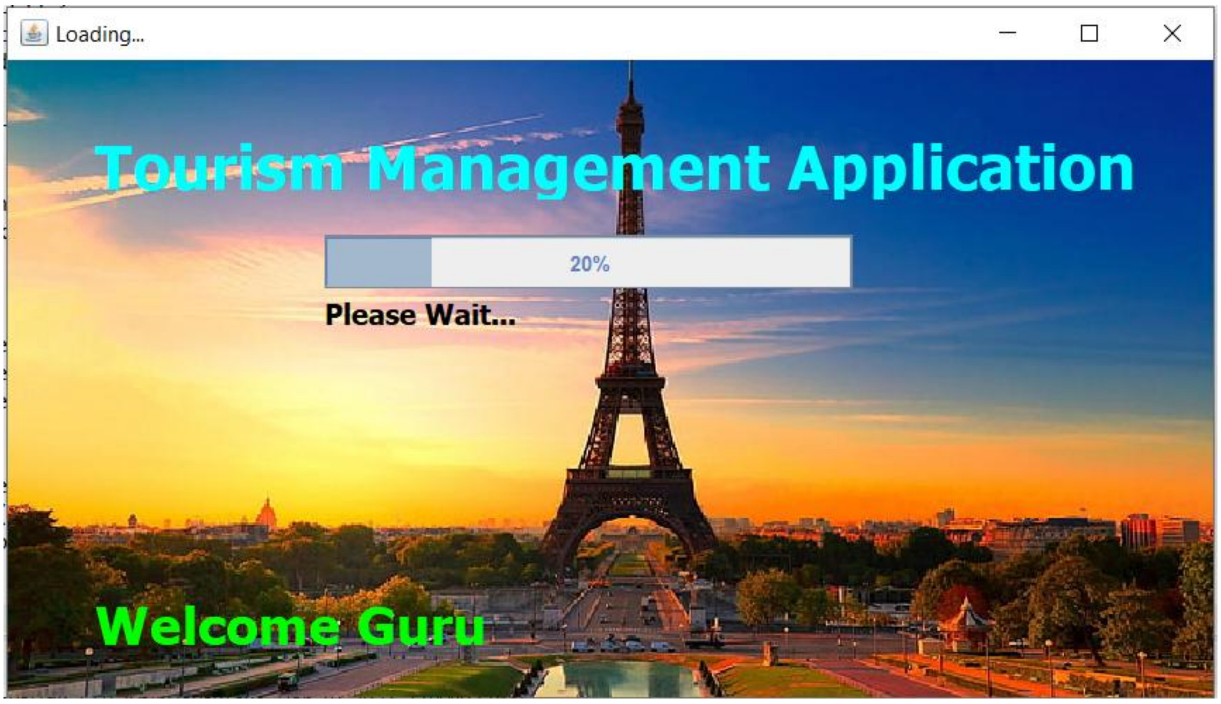
```
package tourismmanagementsystem;

import java.sql.*;

public class Conn {
    Connection c;
    Statement s;
    public Conn(){
        try{
          Class.forName("com.mysql.jdbc.Driver");
          c=DriverManager.getConnection("jdbc:mysql:///dms","root","root");
          s=c.createStatement();
        }
        catch(Exception e){

        }
    }

}
```

Chapter 7:Output

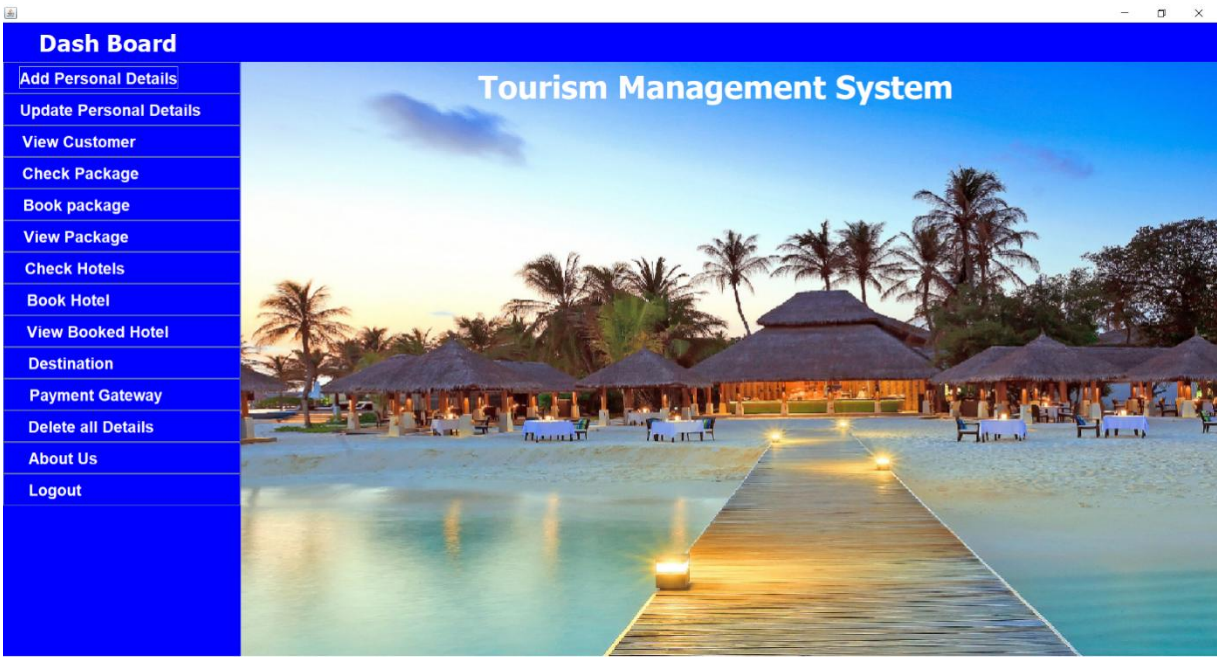
LOGIN PAGE:



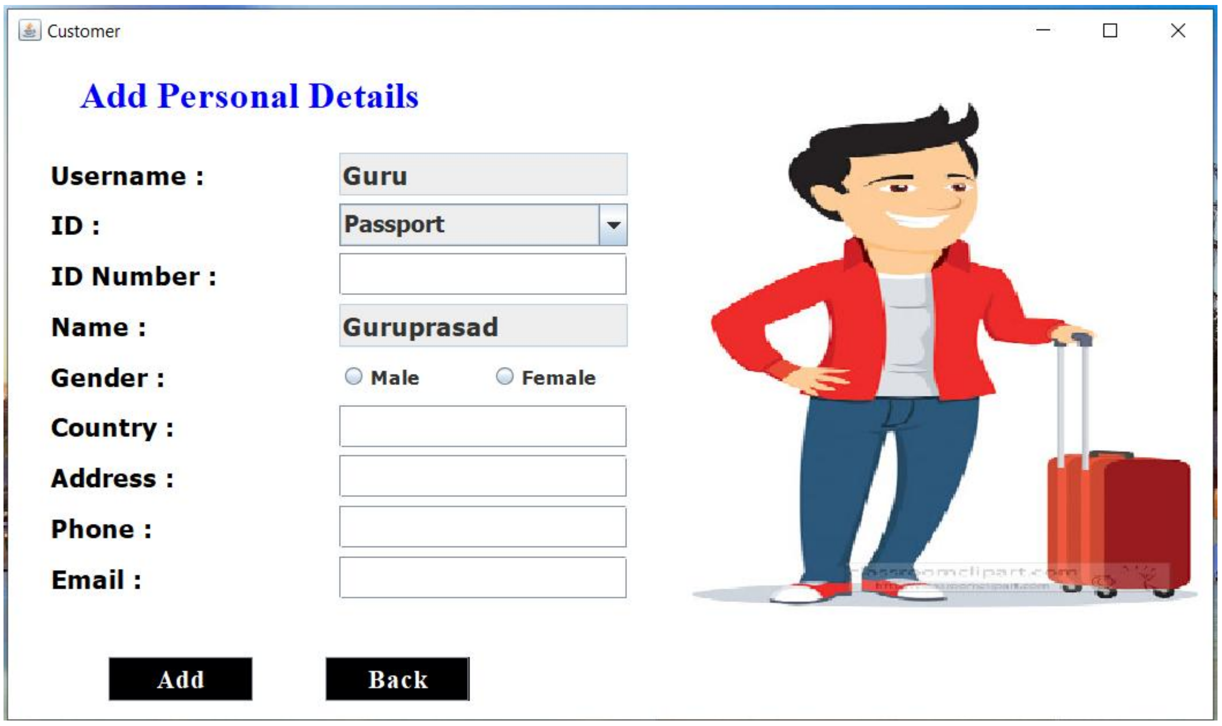
LOADING PAGE:



MAIN FRAME:

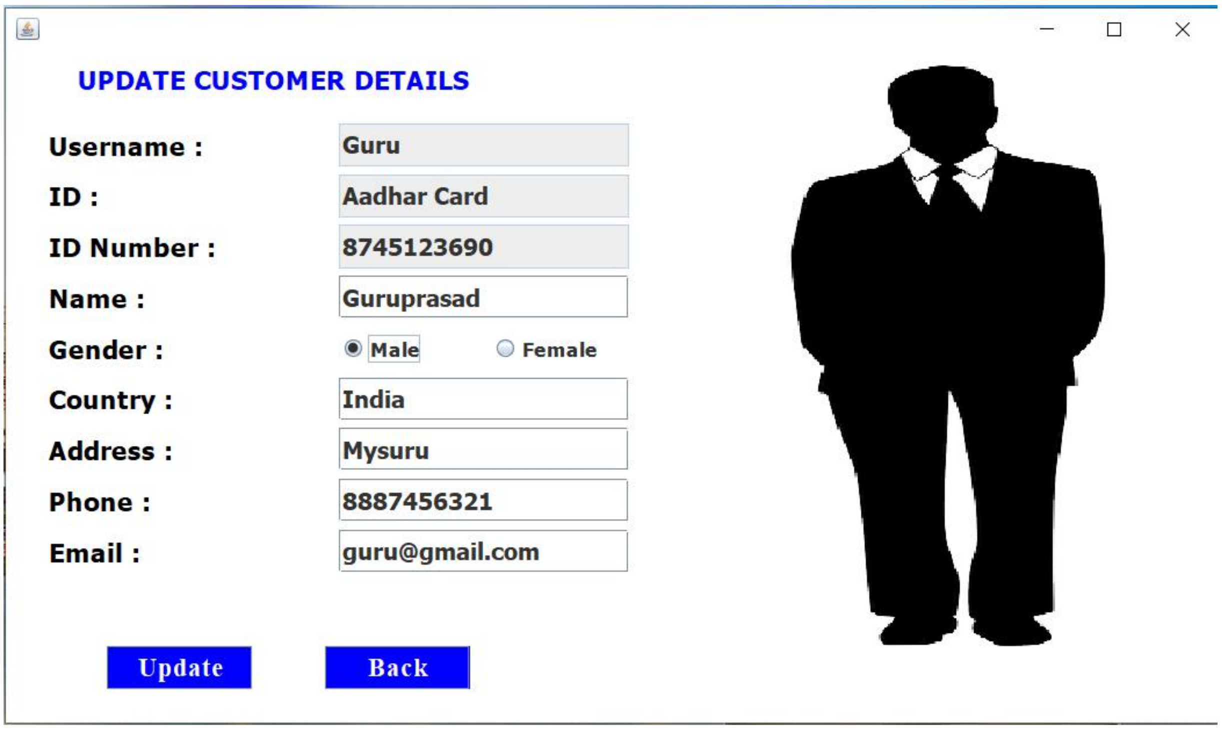


PERSONAL DETAILS:



Username :	Guru
ID :	Passport
ID Number :	
Name :	Guruprasad
Gender :	Male Female
Country :	
Address :	
Phone :	
Email :	

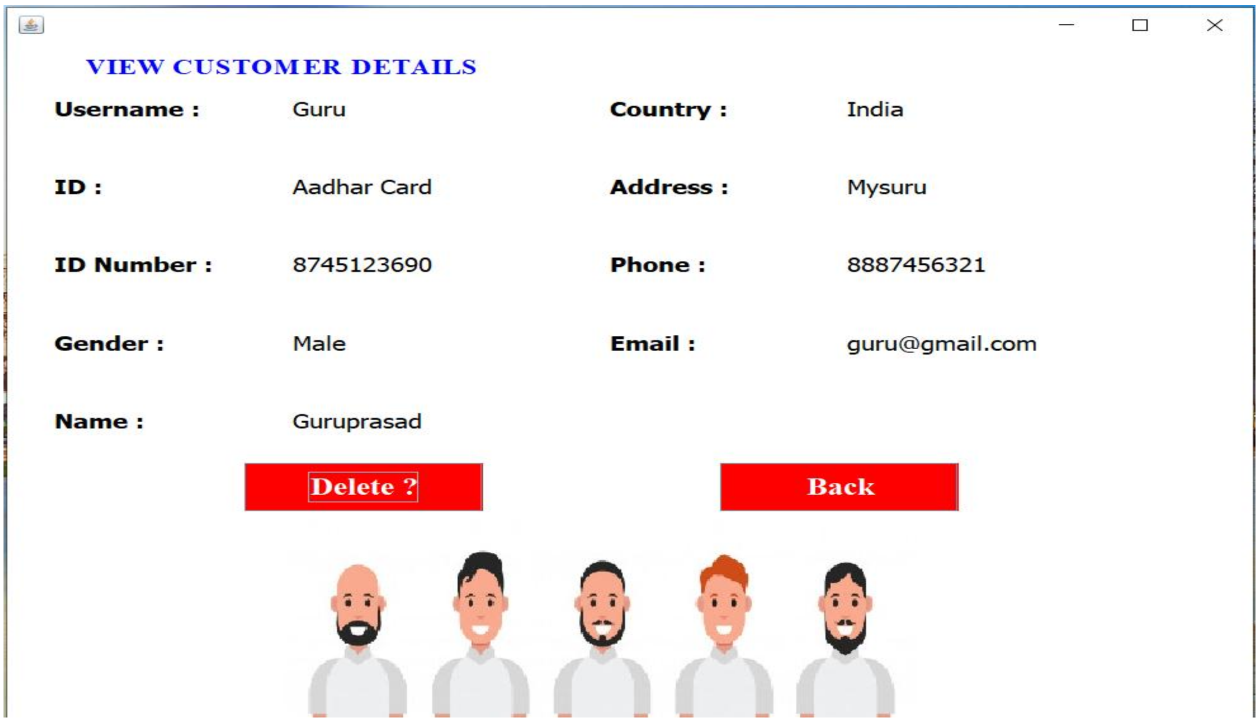
UPDATE CUSTOMER PAGE:



UPDATE CUSTOMER DETAILS

Username :	Guru
ID :	Aadhar Card
ID Number :	8745123690
Name :	Guruprasad
Gender :	[✓ Male] Female
Country :	India
Address :	Mysuru
Phone :	8887456321
Email :	guru@gmail.com

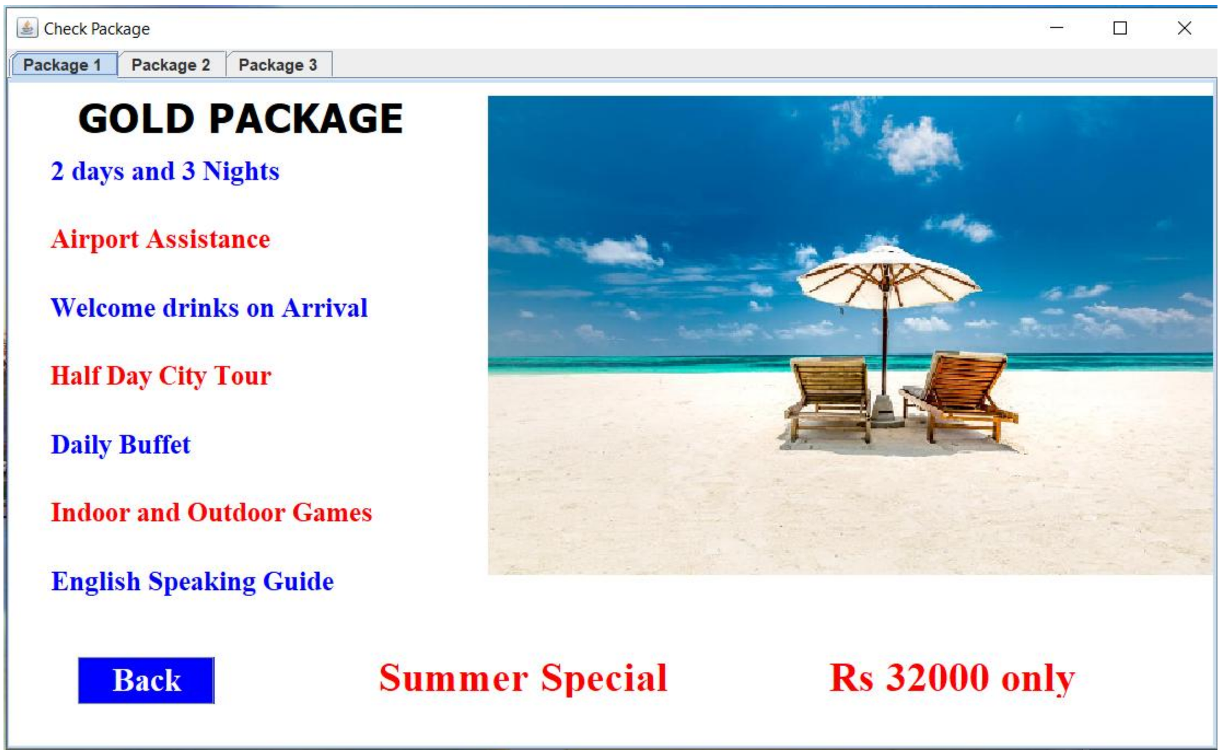
VIEW CUSTOMER PAGE:



VIEW CUSTOMER DETAILS

Username :	Guru	Country :	India
ID :	Aadhar Card	Address :	Mysuru
ID Number :	8745123690	Phone :	8887456321
Gender :	Male	Email :	guru@gmail.com
Name :	Guruprasad		

PACKAGE PAGE:



Check Package

Package 1 | Package 2 | Package 3

GOLD PACKAGE

2 days and 3 Nights

Airport Assistance

Welcome drinks on Arrival

Half Day City Tour

Daily Buffet

Indoor and Outdoor Games

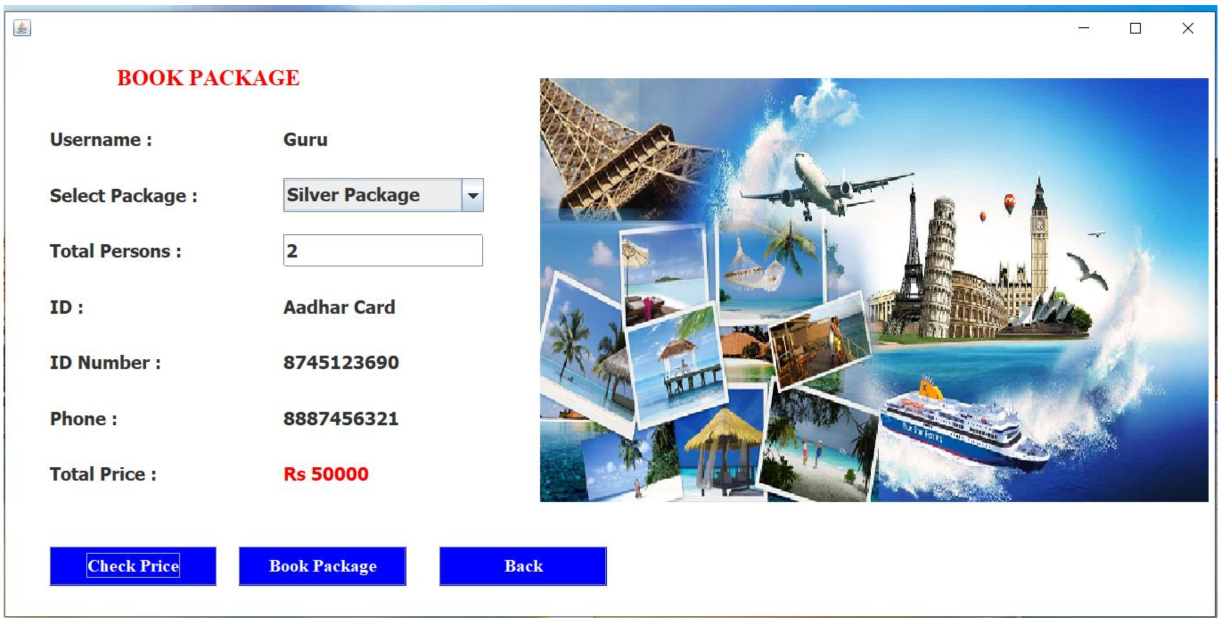
English Speaking Guide

Back

Summer Special

Rs 32000 only

BOOK PACKAGE:

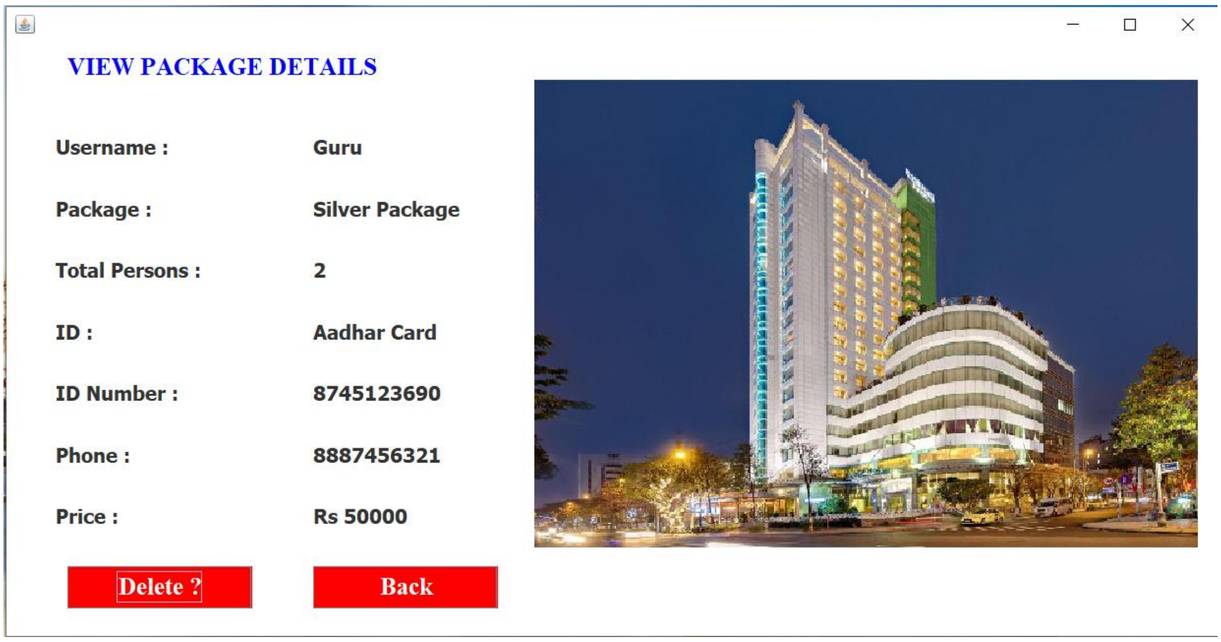


BOOK PACKAGE

Username :	Guru
Select Package :	Silver Package
Total Persons :	2
ID :	Aadhar Card
ID Number :	8745123690
Phone :	8887456321
Total Price :	Rs 50000

Check Price | Book Package | Back

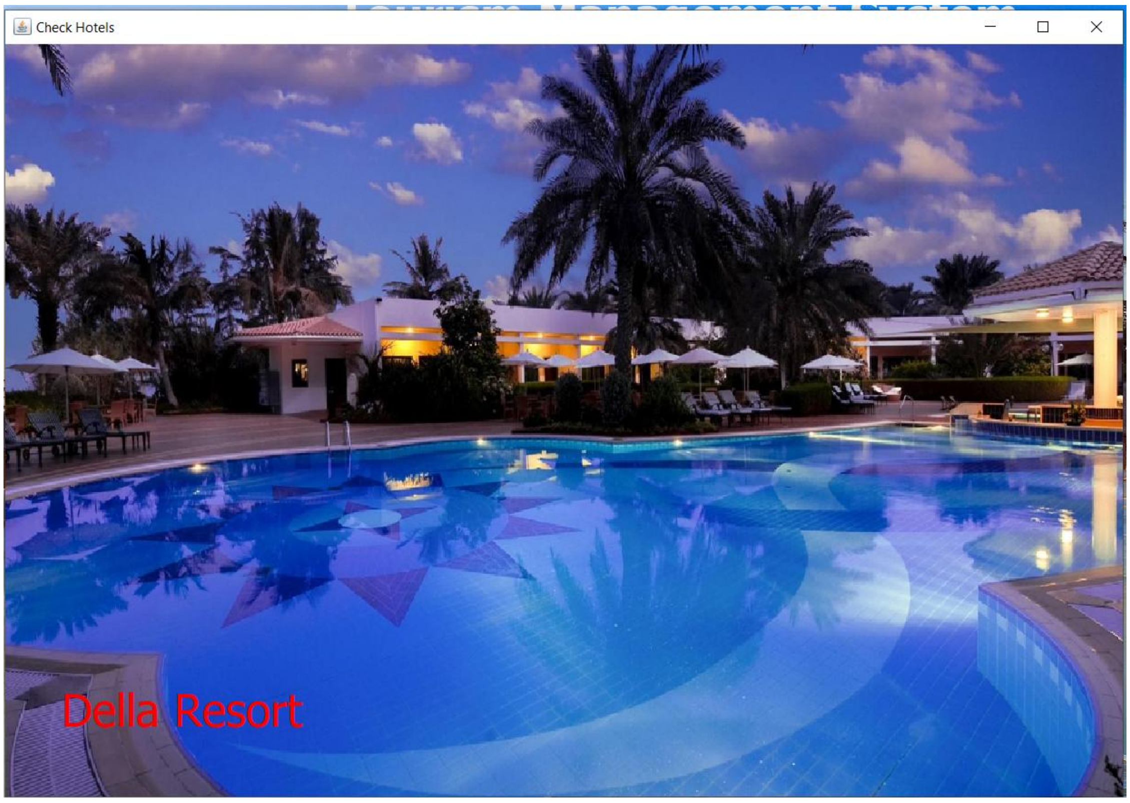
VIEW PACKAGE:



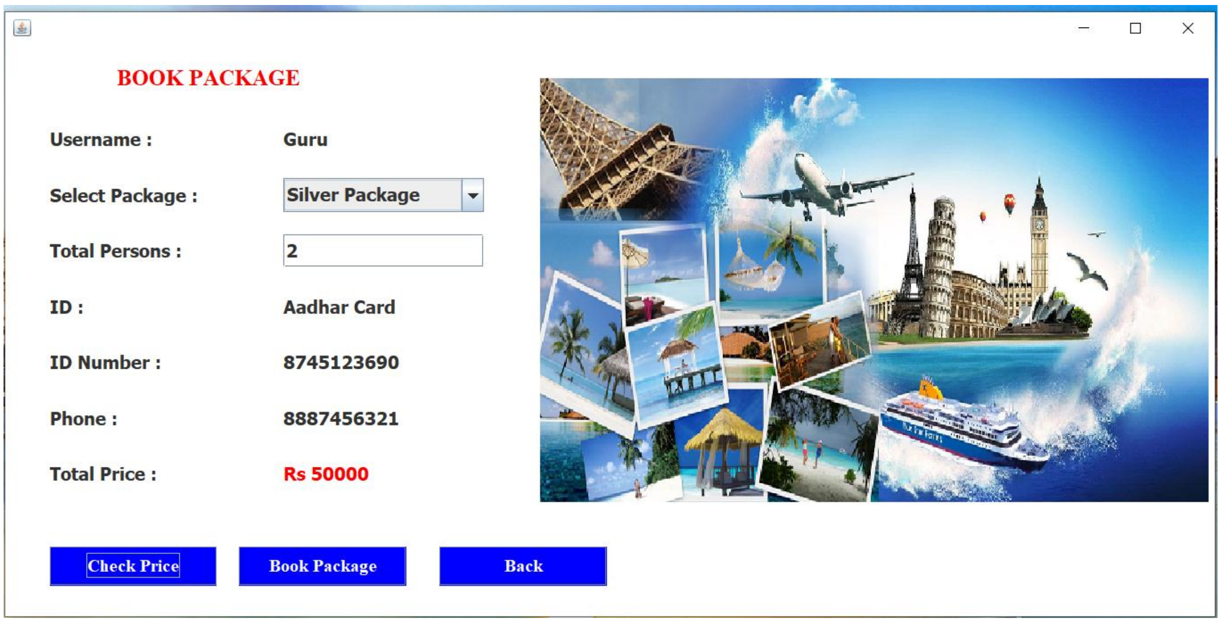
VIEW PACKAGE DETAILS

Username :	Guru
Package :	Silver Package
Total Persons :	2
ID :	Aadhar Card
ID Number :	8745123690
Phone :	8887456321
Price :	Rs 50000

HOTEL/RESORT PAGE:



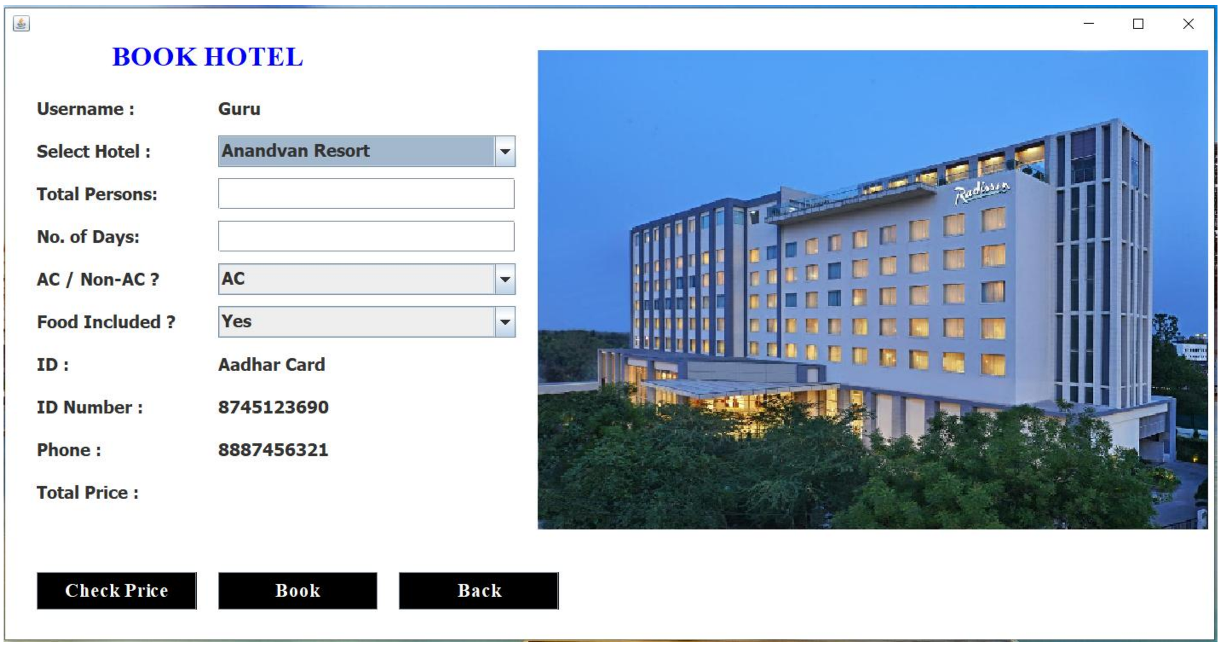
BOOK PACKAGE:



BOOK PACKAGE

Username :	Guru
Select Package :	Silver Package
Total Persons :	2
ID :	Aadhar Card
ID Number :	8745123690
Phone :	8887456321
Total Price :	Rs 50000

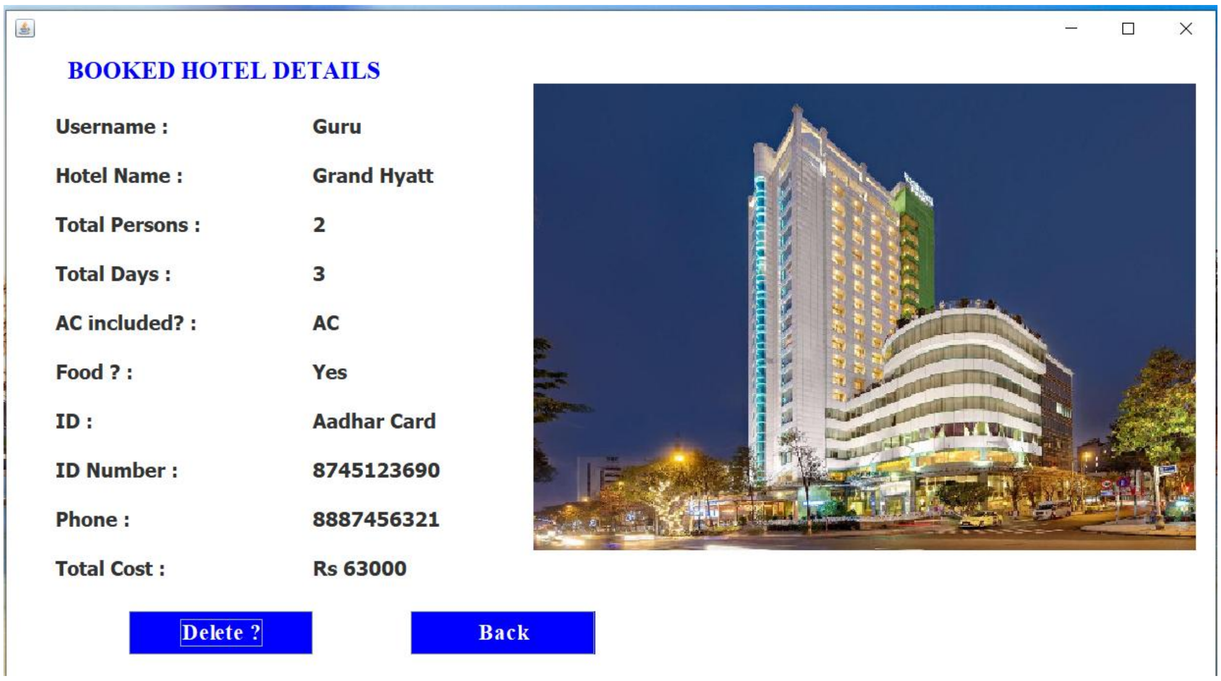
BOOK HOTEL:



BOOK HOTEL

Username :	Guru
Select Hotel :	Anandvan Resort
Total Persons:	
No. of Days:	
AC / Non-AC ?	AC
Food Included ?	Yes
ID :	Aadhar Card
ID Number :	8745123690
Phone :	8887456321
Total Price :	

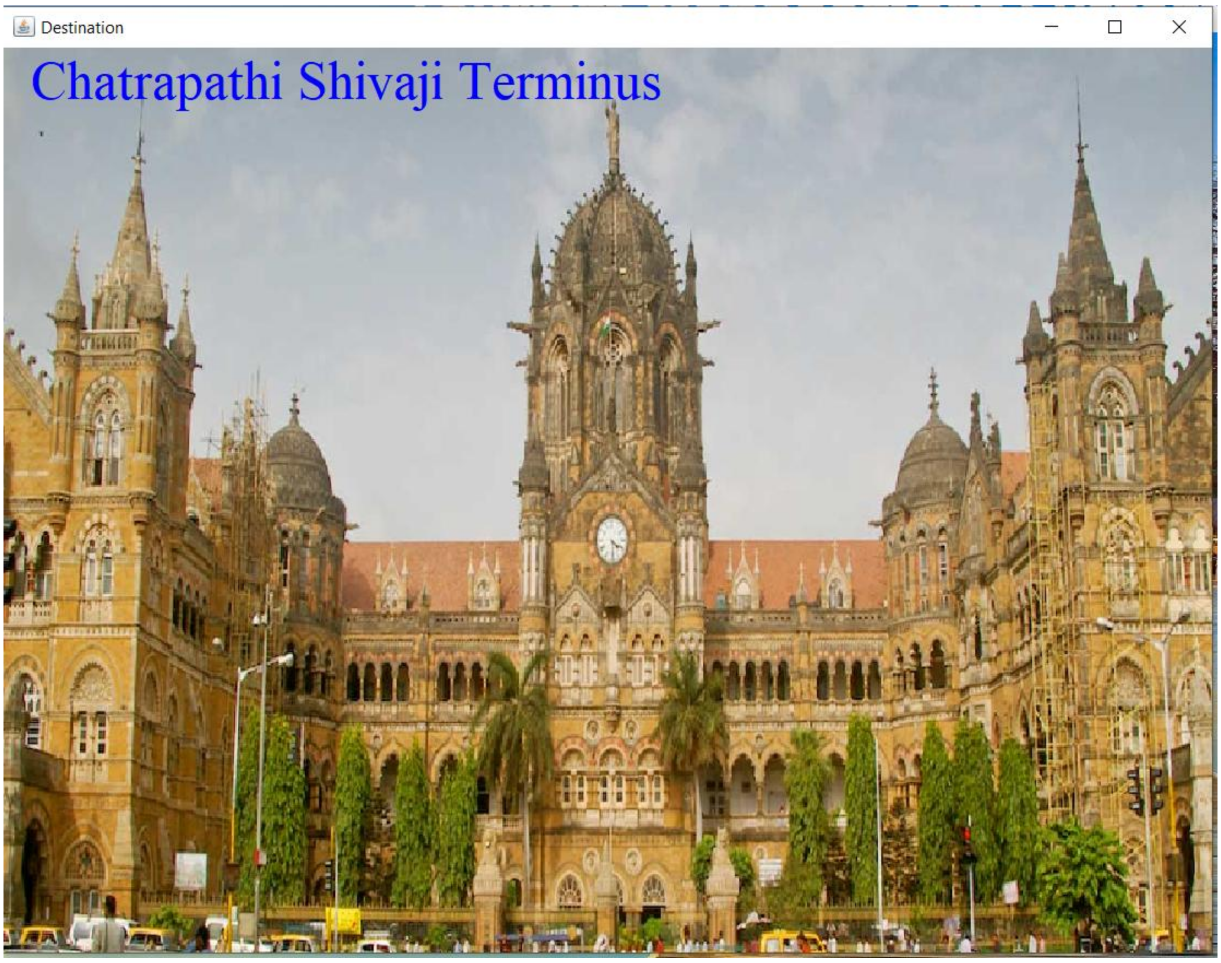
VIEW HOTEL:



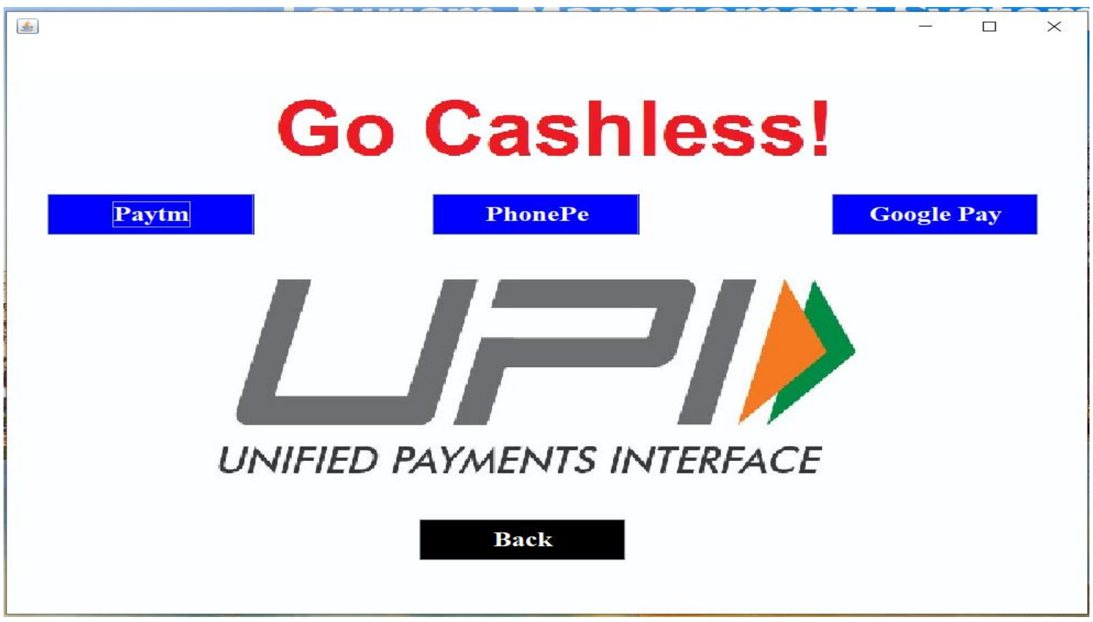
BOOKED HOTEL DETAILS

Username :	Guru
Hotel Name :	Grand Hyatt
Total Persons :	2
Total Days :	3
AC included? :	AC
Food ? :	Yes
ID :	Aadhar Card
ID Number :	8745123690
Phone :	8887456321
Total Cost :	Rs 63000

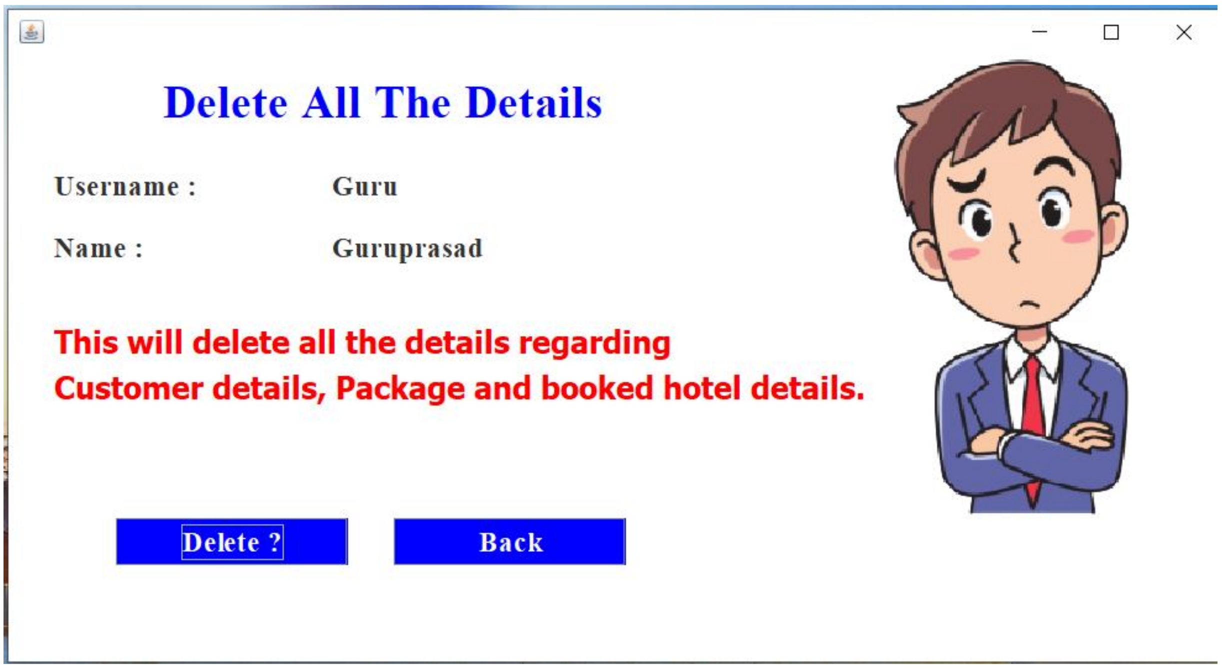
DESTINATION PAGE:



PAYMENT:



DELETE ALL:



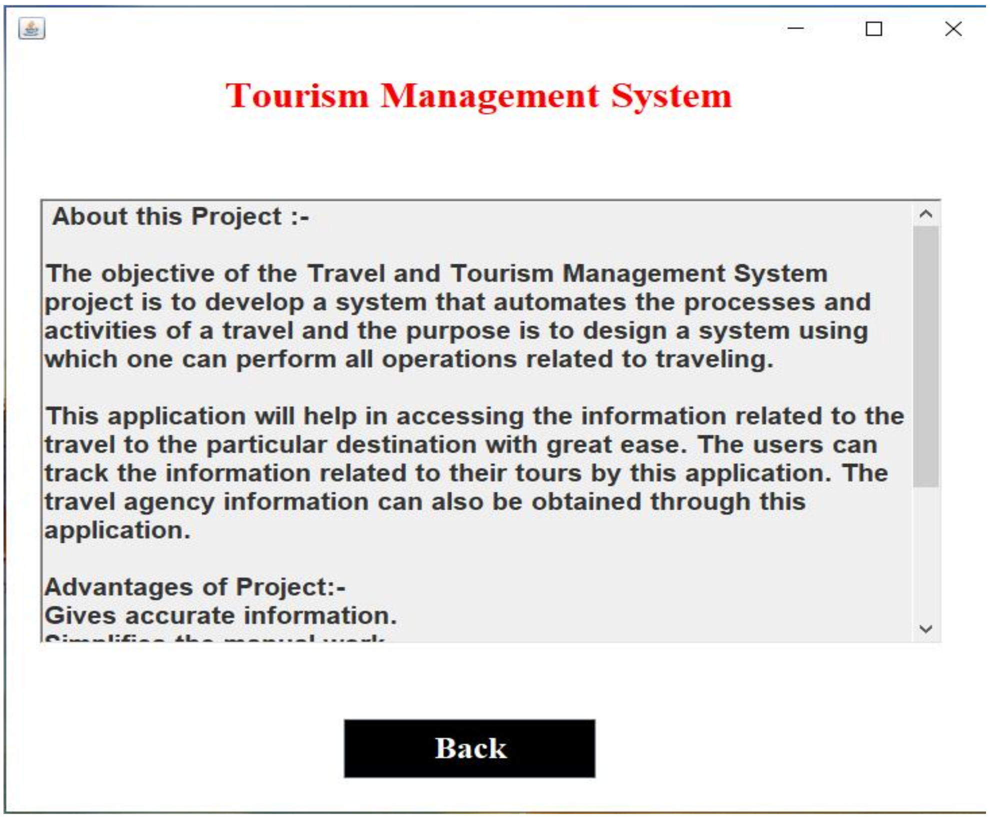
Delete All The Details

Username : Guru

Name : Guruprasad

This will delete all the details regarding
Customer details, Package and booked hotel details.

ABOUT PAGE:



About this Project :-

The objective of the Travel and Tourism Management System project is to develop a system that automates the processes and activities of a travel and the purpose is to design a system using which one can perform all operations related to traveling.

This application will help in accessing the information related to the travel to the particular destination with great ease. The users can track the information related to their tours by this application. The travel agency information can also be obtained through this application.

Advantages of Project:-
Gives accurate information.

Chapter 8 - Conclusion

In conclusion, the tourism management system exhibits promising feasibility across its key operations. User authentication, facilitated through validation against a MySQL database, is a common and essential aspect of software systems, albeit requiring robust security measures. The inclusion of loading animations during application initialization enhances user experience by providing visual feedback on system activity, a feature easily achievable using Java Swing components. Similarly, implementing slideshows for displaying hotel and destination images is feasible with Java's javax.swing components. Database connectivity, enabled through JDBC, demonstrates standard operations for establishing connections, executing queries, and processing results. However, the reliance on absolute positioning for user interface layout, while offering precise control, may benefit from exploration of more responsive layout strategies to accommodate diverse screen sizes. Despite these considerations, with careful attention to security, usability, and performance enhancements, the tourism management system holds the potential to deliver a reliable and user-friendly experience.

In further consideration, the feasibility of the tourism management system's operations underscores its potential to effectively cater to users' needs within the travel and tourism domain. The robustness of user authentication procedures, when coupled with stringent security measures like password hashing and defense against SQL injection, ensures the protection of sensitive user information. Additionally, the implementation of loading animations not only enhances the application's visual appeal but also provides valuable feedback to users, assuring them of ongoing system activity during potentially lengthy processes. The utilization of slideshows to showcase hotels and destinations adds a dynamic and engaging element to the user interface, fostering user interest and exploration. Moreover, the seamless integration with a MySQL database through JDBC affirms the system's capability to handle data operations efficiently, contributing to a smooth user experience. While the current user interface layout may benefit from refinement to improve responsiveness across different devices, this aspect presents an opportunity for enhancement rather than a fundamental obstacle. Overall, the tourism management system demonstrates substantial feasibility across its core functionalities, laying a solid foundation for its potential success in serving the needs of travelers and tourism enthusiasts.

Chapter 9: Future Enhancement

The process of the system we can consider here, can maintain the databases of the system. We can insert to the databases and retrieve all the information.

The main aim of this project is to help the tourists to manage their trip. It makes all operation of the tour company easy and accurate. The standalone platform makes tourism management easy by handling requests ad providing servers for the customers located at different parts of the various cities.

Different modules have been incorporated in this project to handle different parts and sector of the tour management field.

References

We have taken references from many resources like YouTube and many websites.

Websites:

- https://www.w3schools.com
- https://www.javatpoint.com
- https://www.codecademy.com
- https://www.stackoverflow.com

YouTube video links:

1. https://youtu.be/5vzCjvUwMXg
2. https://youtu.be/dwVj_g3TpZ4
3. https://youtu.be/L5RpqspNAuc