Hotel Reservation System Design Document

#### Authors Name, Surname: Kerem Kuru

#### Number: 220303016

# Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
|  | 0.0 | Initial draft | Prof. Dr. Taner Çevik |
| 25/05/2024 | 0.1 | Report is done. | Kerem Kuru |

# 1. Introduction

## 1.1 Purpose

This document provides a design and implementation guide for the Hotel Reservation System. Target users are hotel customers and managers.

## 1.2 SYSTEM FUNCTIONS

The system allows customers to make reservations and managers to oversee all transactions.

## 1.3 Design Objectives

The primary objective is to create a user-friendly desktop application that enhances the efficiency of hotel operations. The system is designed using Object-Oriented Programming (OOP) principles and is easily adaptable to use databases for data storage.

## 1.4 References

Hotel Management Guidelines

NTU Website

W3Schools

Some Java Forums

ChatGPT

# 2. Design Overview

## 2.1 Introduction

This section explains the design principles of the system. It aims to provide an interface for the Hotel Reservation System and management.

## 2.2 Environment Overview

The application is designed to run in a desktop environment, allowing users to access local databases without requiring an internet connection.

## 2.3 System Architecture

**User Roles:**  
- Customer: Makes reservations.  
- Administrator: Oversees all transactions.

**Main Screens**

**Hotel Management Login**

**Hotel Reservation Login**

**Reservation Information List**

**Hotel Management Screens**

**Registered List**

**Login Screen**

**Login**

**Register**

**Forgot Password**

**Reservation Screen**

**Reservation Options**

# 3. Interfaces and Data Stores

## 3.1 System Interfaces

The system provides several interfaces to facilitate different functionalities for various user roles. Key interfaces include:  
- Login Screen  
- Reservation Entry Screen  
- Customer Registration Screen  
- Reservation List Screen  
- Password Recovery Screen

## 3.2 Data Stores

All data in the Hotel Reservation System is stored in files as objects.

After the customer registers, the information is saved in the ziyaretciListesi.txt file. The features the customer chooses when making a reservation are saved in the ziyarecibilgisi.txt file.

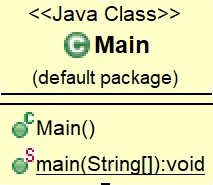
# 4. Structural Design

## 4.1 Design Explanation and Rationale

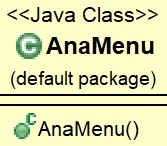
The main concept and the structure type of the program is OOP. Which means the program is working with objects only. In the database management, due to the benefits of OOP, system does not need to create a file for each field. Another usage of OOP is the polymorphism and abstract structures. It makes the program more easy to handle, readable and compatible with the Object Stream operations.

## 4.2 Class Diagram

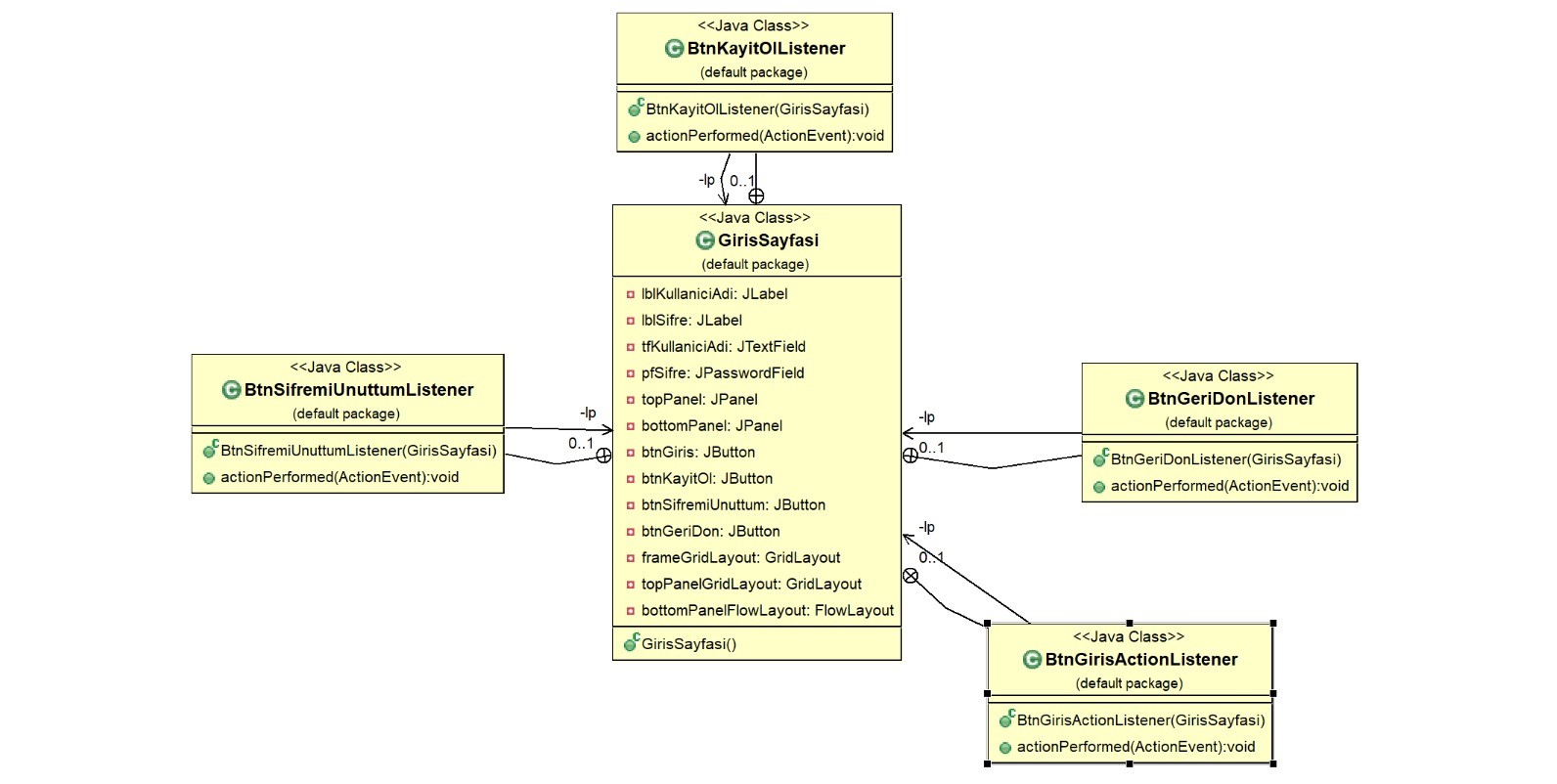
**Main**



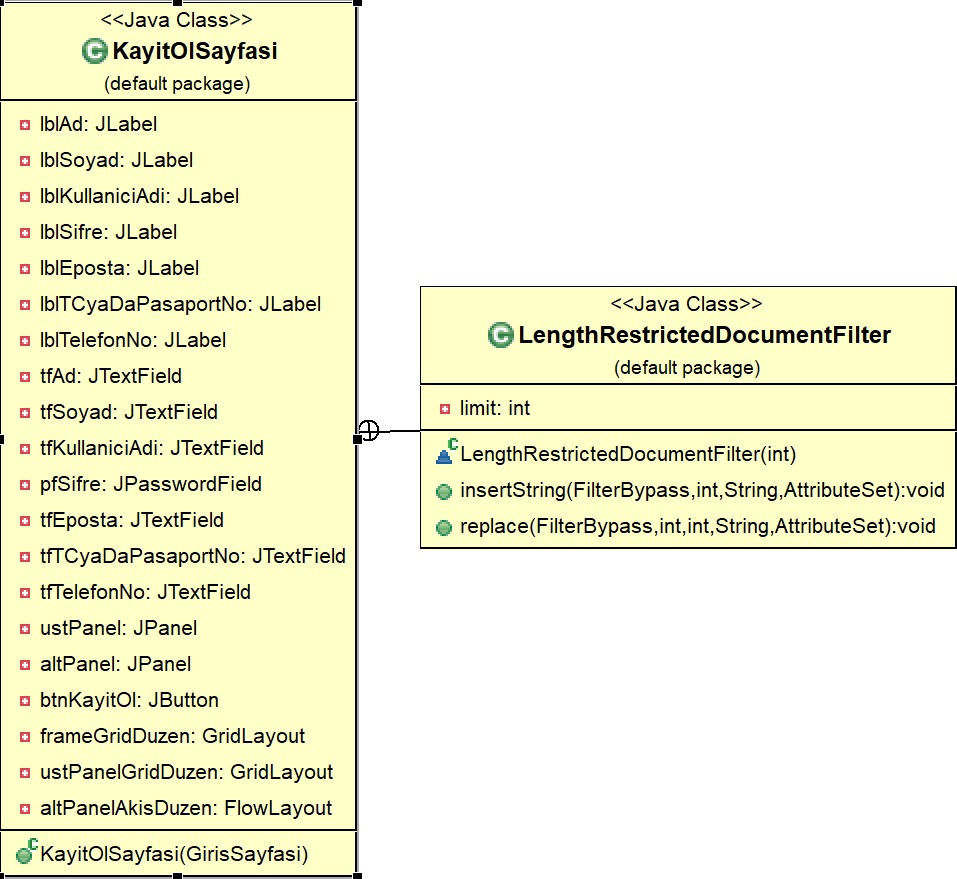
**Ana Menu**



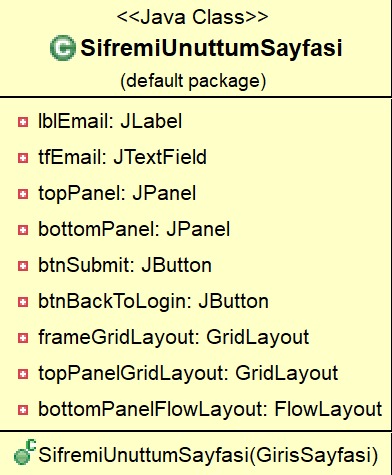
**Login Page**



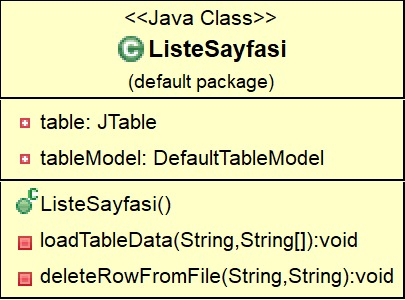
**Register Page**



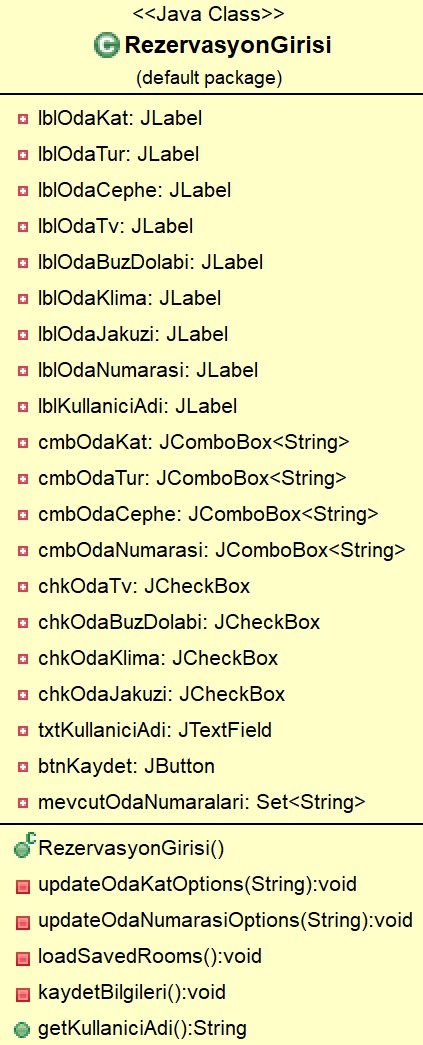
**Forgot Password Page**



**List Page**



**Reservation Page**



## 4.3 Class Descriptions

**AnaMenu:**

**Attributes**

* **JLabel lbl**: Label for the welcome message.
* **JButton otelYonetimiButton**: Button for hotel management login.
* **JButton rezervasyonButton**: Button for hotel reservation login.

**Methods:**

* **AnaMenu()**: Constructor method; sets the window title, defines window close operation, creates the user interface components, and manages button click events.

**GirisSayfasi:**

**Attributes**

* **private JLabel lblKullaniciAdi;**: Label for "Username".
* **private JLabel lblSifre;**: Label for "Password".
* **private JTextField tfKullaniciAdi;**: Text field for entering username.
* **private JPasswordField pfSifre;**: Password field for entering password.
* **private JPanel topPanel;**: Panel containing components for username and password.
* **private JPanel bottomPanel;**: Panel containing buttons.
* **private JButton btnGiris;**: Button for "Login".
* **private JButton btnKayitOl;**: Button for "Register".
* **private JButton btnSifremiUnuttum;**: Button for "Forgot Password".
* **private JButton btnGeriDon;**: Button for "Go Back".
* **private GridLayout frameGridLayout;**: Grid layout for the frame.
* **private GridLayout topPanelGridLayout;**: Grid layout for the top panel.
* **private FlowLayout bottomPanelFlowLayout;**: Flow layout for the bottom panel.

**Methods**

* **GirisSayfasi()**: Constructor method that sets up the user interface components, layouts, and handles button actions.
* **private class BtnGirisActionListener implements ActionListener { ... }**: Action listener class for the "Login" button, responsible for reading a text file (**ziyaretciListesi.txt**) to verify username and password.
* **private class BtnKayitOlListener implements ActionListener { ... }**: Action listener class for the "Register" button, hides the login window and opens the registration window (**KayitOlSayfasi**).
* **private class BtnSifremiUnuttumListener implements ActionListener { ... }**: Action listener class for the "Forgot Password" button, hides the login window and opens the password recovery window (**SifremiUnuttumSayfasi**).
* **private class BtnGeriDonListener implements ActionListener { ... }**: Action listener class for the "Go Back" button, disposes of the login window and opens the main menu window (**AnaMenu**).

**KayitOlSayfasi:**

**Attributes**

* **private JLabel lblAd;**: Label for "Name".
* **private JLabel lblSoyad;**: Label for "Surname".
* **private JLabel lblKullaniciAdi;**: Label for "Username".
* **private JLabel lblSifre;**: Label for "Password".
* **private JLabel lblEposta;**: Label for "E-mail".
* **private JLabel lblTCyaDaPasaportNo;**: Label for "TC or Passport No".
* **private JLabel lblTelefonNo;**: Label for "Phone Number".
* **private JTextField tfAd;**: Text field for entering name.
* **private JTextField tfSoyad;**: Text field for entering surname.
* **private JTextField tfKullaniciAdi;**: Text field for entering username.
* **private JPasswordField pfSifre;**: Password field for entering password.
* **private JTextField tfEposta;**: Text field for entering e-mail.
* **private JTextField tfTCyaDaPasaportNo;**: Text field for entering TC or Passport number.
* **private JTextField tfTelefonNo;**: Text field for entering phone number.
* **private JPanel ustPanel;**: Panel containing labels and text fields for personal information.
* **private JPanel altPanel;**: Panel containing the "Register" button.
* **private JButton btnKayitOl;**: Button for registering a new user.
* **private GridLayout frameGridDuzen;**: Grid layout for the frame.
* **private GridLayout ustPanelGridDuzen;**: Grid layout for the top panel.
* **private FlowLayout altPanelAkisDuzen;**: Flow layout for the bottom panel.

**Methods**

* **KayitOlSayfasi(GirisSayfasi lp)**: Constructor method that initializes the user interface components, layouts, and handles the registration process when the "Register" button is clicked.
* **class LengthRestrictedDocumentFilter extends DocumentFilter { ... }**: Inner class implementing **DocumentFilter** to restrict the length of input for specific text fields (**tfTCyaDaPasaportNo** and **tfTelefonNo**). It overrides **insertString** and **replace** methods to enforce the length restriction.

**ListeSayfasi:**

**Attributes**

* **private JTable table;**: The table component to display data.
* **private DefaultTableModel tableModel;**: The model that manages data for the table.

**Constructor**

* **ListeSayfasi()**: Initializes the JFrame window with a title and default close operation. Sets up panels for buttons, a table, and handles button actions for loading and deleting data.

**Methods**

* **void loadTableData(String filePath, String[] columnNames)**: Loads data from a specified file (**filePath**) into the table using **DefaultTableModel**. Clears existing data and sets column names provided in **columnNames**.
* **void deleteRowFromFile(String filePath, String kullaniciAdi)**: Deletes a row from the file (**filePath**) based on the **kullaniciAdi** (username). It reads all lines from the file except the one matching the username, writes them back to the file, and updates the table.

**RezervasyonGirişi:**

**Attributes**

* **Labels (JLabel):**
  + **private JLabel lblOdaKat;**
  + **private JLabel lblOdaTur;**
  + **private JLabel lblOdaCephe;**
  + **private JLabel lblOdaTv;**
  + **private JLabel lblOdaBuzDolabi;**
  + **private JLabel lblOdaKlima;**
  + **private JLabel lblOdaJakuzi;**
  + **private JLabel lblOdaNumarasi;**
  + **private JLabel lblKullaniciAdi;**
* **ComboBoxes (JComboBox<String>):**
  + **private JComboBox<String> cmbOdaKat;**
  + **private JComboBox<String> cmbOdaTur;**
  + **private JComboBox<String> cmbOdaCephe;**
  + **private JComboBox<String> cmbOdaNumarasi;**
* **Checkboxes (JCheckBox):**
  + **private JCheckBox chkOdaTv;**
  + **private JCheckBox chkOdaBuzDolabi;**
  + **private JCheckBox chkOdaKlima;**
  + **private JCheckBox chkOdaJakuzi;**
* **Text Field (JTextField):**
  + **private JTextField txtKullaniciAdi;**
* **Button (JButton):**
  + **private JButton btnKaydet;**
* **Set (HashSet<String>):**
  + **private Set<String> mevcutOdaNumaralari;**

**Constructor**

* **RezervasyonGirisi()**: Initializes the JFrame window titled "Rezervasyon Girişi" with components like labels, combo boxes, check boxes, text field, and button. Sets up event listeners for combo boxes and buttons to handle user interactions.

**Methods**

* **private void updateOdaKatOptions(String selectedTur)**: Updates the options in **cmbOdaKat** based on the selected room type (**selectedTur**).
* **private void updateOdaNumarasiOptions(String selectedKat)**: Updates the options in **cmbOdaNumarasi** based on the selected room floor (**selectedKat**).
* **private void loadSavedRooms()**: Loads existing room numbers from **"ziyaretcibilgisi.txt"** into **mevcutOdaNumaralari** using a **HashSet**.
* **private void kaydetBilgileri() throws IOException**: Saves reservation details entered by the user into **"ziyaretcibilgisi.txt"**. Checks if the user already has a reservation based on the username (**kullaniciAdi**) and prevents duplicate entries.
* **public String getKullaniciAdi()**: Getter method that returns the text entered in **txtKullaniciAdi**.

**SifremiUnuttumSayfasi:**

**Attributes**

* **Labels (JLabel):**
  + **private JLabel lblEmail;**
* **Text Field (JTextField):**
  + **private JTextField tfEmail;**
* **Panels (JPanel):**
  + **private JPanel topPanel;**
  + **private JPanel bottomPanel;**
* **Buttons (JButton):**
  + **private JButton btnSubmit;**
  + **private JButton btnBackToLogin;**
* **Layout Managers:**
  + **private GridLayout frameGridLayout;**
  + **private GridLayout topPanelGridLayout;**
  + **private FlowLayout bottomPanelFlowLayout;**

**Constructor**

* **SifremiUnuttumSayfasi(GirisSayfasi lp)**: Constructs the password recovery page (**SifremiUnuttumSayfasi**) with components for entering an email address (**tfEmail**), submitting the request (**btnSubmit**), and returning to the login page (**btnBackToLogin**). It sets up event listeners for button clicks to handle password retrieval and navigation back to the login page.

**Methods**

* **void actionPerformed(ActionEvent evt)**: Event handler method for **btnSubmit** that retrieves the email address entered (**tfEmail.getText().trim()**), reads through **"ziyaretciListesi.txt"** to find a matching email address, and displays the corresponding username and password if found. It shows appropriate error messages if the email is not found or if there are file reading errors.

**Main:**

**Attributes**

* No explicit attributes are declared in the **Main** class.

**Methods**

* **main(String[] args)**:
  + Entry point of the application. It schedules the creation of a new **AnaMenu** instance using **SwingUtilities.invokeLater()**, ensuring that Swing components are initialized and updated on the Event Dispatch Thread (EDT), which is crucial for Swing applications to maintain responsiveness and avoid threading issues.

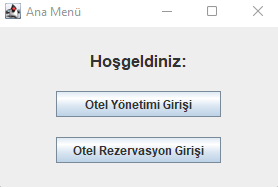
**Description**

The **Main** class serves as the starting point (**main** method) for a Swing-based Java application. It initializes the Swing components by invoking **SwingUtilities.invokeLater()** with an anonymous **Runnable** object that creates an instance of **AnaMenu**.

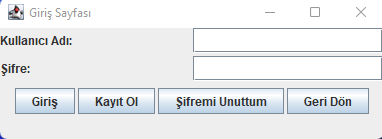
# 5. Dynamic Model

## 5.1 Scenarios

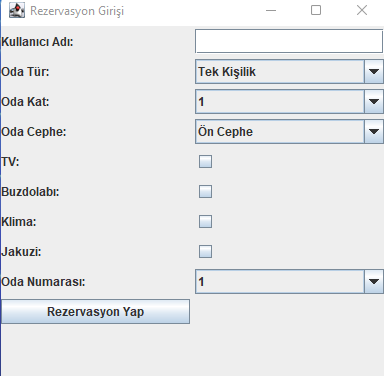
- Scenario 1: Customer Reservation  
 - Steps:  
 1. Customer logs in. (Otel Rezervasyon Girişi)



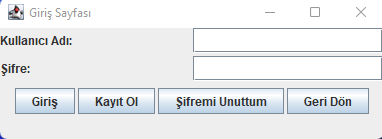
2. Navigates to the Reservation Entry screen. (Customer information is entered)



3. Enters reservation details. Saves the reservation.



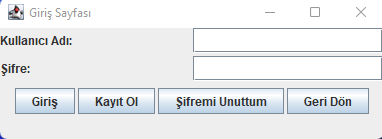
- Scenario 2: Customer Registration  
 - Steps:  
 1. Navigates to the Customer Registration screen.



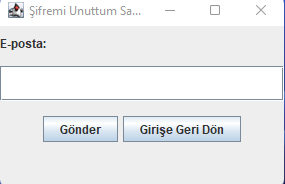
2. Enters customer details and registers the customer.



- Scenario 3: Password Recovery  
 - Steps:  
 1. User navigates to the Password Recovery screen.



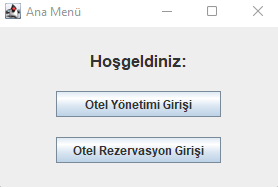
2. Enters their email address. User can learn username and password.



-Scenario 4: Admin Login

-Steps:

1. Admin log in. (Otel Yönetimi Girişi)



2. Navigates to the Hotel Management Entry screen. (Admin information is entered)



3. Can view Customer Information and Customer Reservation Information.

