ElizabethRestaurant: Wedding Package Booking System

A Java implementation for a dynamic, feature-rich wedding package reservation system.

Project Overview

The Challenge of Dynamic Pricing

The core goal of the "ElizabethRestaurant" project was to design and implement a robust Java application capable of managing multiple wedding packages and dynamically calculating the total cost based on user selections and guest counts.

Accurate Cost Calculation

Ensuring precise total charge based on complex, tiered pricing structures.

Seamless User Input

Allowing customers to easily select packages, specify guest counts, and choose add-ons.

Modular Java Implementation

Structuring the program using clear classes and methods for maintainability.

Key Functional Features

Multiple Packages

The system offers four distinct wedding packages, each catering to different needs and group sizes.

Dynamic Pricing

Calculations automatically adjust costs based on the number of guests and optional services (e.g., fireworks, gifts).

Validation & Termination

Handles both successful termination (inputting -99) and graceful exit upon invalid package selection.

Detailed Wedding Package Structures

Each of the four packages presents a unique pricing model, requiring careful implementation of conditional logic in the Java code.

1	LAVINIA WEDDING PACKAGE	Simple Per-Guest Rate: **Rs. 5,500.00** per guest.
2	GOVERNOR'S WEDDING PACKAGE	Tiered Pricing: **Rs. 5,700.00** (up to 350 guests); **Rs. 6,200.00** (guests 351–450). Max 450 guests.
3	MESTIZO WEDDING PACKAGE	Base Rate (**Rs. 6,400.00**) + Fixed Cake Fee (**Rs. 25,000**) + Optional Fireworks (**Rs. 12,000**).
4	VICTORIAN WEDDING PACKAGE	Base Rate (**Rs. 6,800.00**) + Optional Gifts (**Rs. 900** per guest) with a 15% discount if gifts are selected.

Implementation Milestones

The project followed a standard development lifecycle, moving from conceptualization to a fully tested application.

Idea Generation

Conceptualizing core features and user flows for the booking system.

System Design

Defining package structures, pricing rules, and Java class organization.

Implementation

Writing and compiling the Java code to realize the defined functionalities.

User Testing

Conducting verification to ensure accurate output for all package inputs and scenarios.

Core Java Structure: Main Method

The central control flow is managed within the `main` method, utilizing essential Java features for input, iteration, and conditional execution.

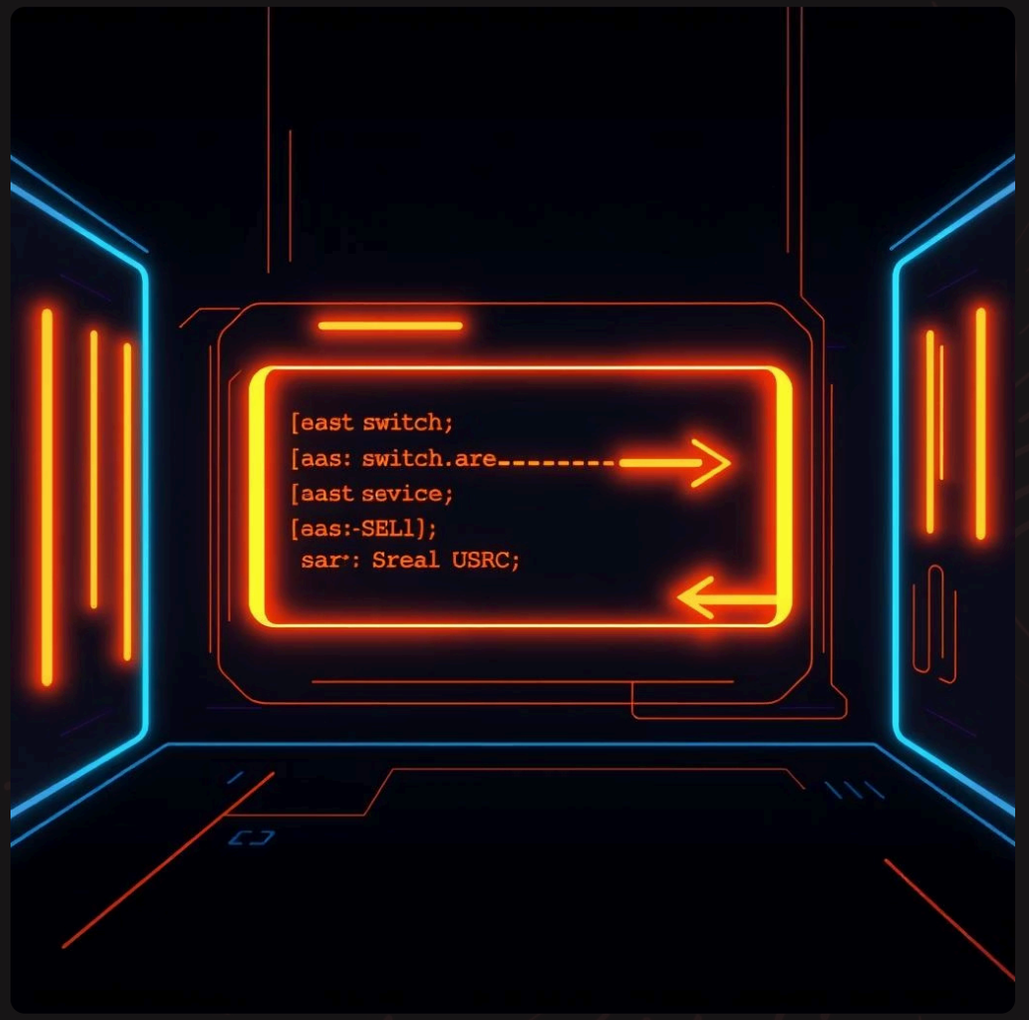
```
import java.util.Scanner;
public class ElizabethHotel {
 public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);
 int packageId;

 while (true) {
 // Prompts and Input Reading...
 packageId = scanner.nextInt();
 if (packageId == -99) { break; }

 // Read number of guests...
 int numberOfGuests = scanner.nextInt();
 double totalCharge = 0;
 boolean giftsChosen = false;

 // Switch statement for package logic...
 }
 scanner.close();
 System.out.println("Thank you for using the Elizabeth Hotel Wedding Booking System!");
 }
}
```

Control Flow: Loops, Conditionals, and Input



- `Scanner` Class

  Used for robust console input handling, capturing both integer (IDs, guests) and string (yes/no responses) data types.

- `while(true)` Loop

  Enables continuous booking functionality, allowing the user to process multiple packages until the explicit exit command (`-99`) is entered.

- `switch` Statement

  The core logic handler that directs the program to the correct, package-specific calculation block based on the user's input ID.

Focus on Package 2 & 4 Calculations

These packages demonstrate the most complex conditional logic, requiring precise mathematical application for tiered rates and optional discounts.

Package 2: Tiered Guest Count Logic

The program uses nested `if-else if` statements to determine the price tier. If the guest count exceeds 350 (up to 450), a composite calculation is performed:

`totalCharge = (350 * 5700) + ((numberOfGuests - 350) * 6200);`

Package 4: Optional Discount Calculation

Guests pay a base rate plus Rs. 900 per guest for gifts. If they choose gifts, a 15% discount is applied to the gift charge:

totalCharge += (900 * numberOfGuests) * 0.85;

Project Outcome and Success

The implemented Java program successfully achieves all defined objectives, providing a reliable and extensible foundation for a real-world booking system.

Requirements Met

Successfully calculates costs for all four packages, including complex optional add-ons and tiered pricing.

User-Friendly

The console interface is clear, requesting input sequentially and providing validated feedback.

Solid Foundation

The modular code structure ensures it serves as a strong starting point for further development.

Future Enhancements: Scaling the System

To elevate the booking system from a console application to a commercial-grade solution, several key improvements are planned.

Implement a Graphical User Interface (GUI)

Transition from console input to a modern GUI (e.g., using JavaFX or Swing) for improved aesthetics and user experience.

Database Integration

Store booking data, customer details, and historical pricing in a relational database for robust record-keeping and data analysis.

Payment and Confirmation Module

Integrate secure payment processing and automated confirmation/receipt generation for finalized bookings.

Configuration & Extensibility

Move package details and pricing from hard-coded values into external configuration files (e.g., JSON/XML) to allow dynamic updates without recompilation.