



SOFTWARE DOCUMENTATION

SOFTWARE DESIGN & ARCHITECTURE PROJECT

HABIBA DARWISH

JANNAH TAREK

AYA KHODEIR

YOUSSEF HESHAM

LOJINE AHMED

YOUSSEF NEGMELDIN

ABDULRAHMAN NASR

CONTENT

SECTION 1 : Core Functionality (Functional and Non-functional Requirements)	2
SECTION 2 : Conceptual Design	4
SECTION 3 : Software Decomposition	5
SECTION 3.1 : Functional Decomposition	6
SECTION 3.2 : Object Oriented Decomposition	10
SECTION 4 : Domain Driven Design	14

SECTION 1 : CORE FUNCTIONALITY

This section contains all the functional and non-functional requirements of the system

Functional Requirements:

1. Displaying all hotel locations on a detailed map
2. Search feature includes several filters to allow users to select several criteria when searching for bookings
3. Search feature should validate the availability of rooms
4. Calculate cost of stay for user given details of room chosen , facilities provided and the period of the booking
5. Option to reschedule or cancel a reservation with a cancellation fee that is calculated based on the amount of time until the reservation date
6. Display All events/amenities in the hotel
7. Receptionists should be able to check-in/checkout guest to/from a certain room
8. After the user selects the room he wishes for, they are redirected to a form that requires user to enter his information to follow the booking cycle
9. Guest should be able to book online or in-person
10. Guest should be able to have a daily QR code to enter restaurant/breakfast with, sent on email or to room
11. Hotel Managers should be able to track inventory for room items
12. User should have his own profile in the system with his history of bookings and loyalty points

13. Generate weekly/monthly reports to the hotel manager.
14. Integrate online payment gateways to hotel website

Non-functional Requirements

1. Product
 - 1.1. maintainability: system should be easily maintained and modified with low cost.
 - 1.2. Access to certain data like guests room numbers should be restricted to admins.
 - 1.3. Safety: Protect user login details and ensure the protection of their financial details.
 - 1.4. Scalability: components should be scalable to handle large amounts of users at peak times and dynamically adjust to increased load.
 - 1.5. User can login at any time of the day to view their account (Availability)
 - 1.6. Reliability: System should back up.
 - 1.7. Usability: System should have a nice UI for ease of training for receptionists.
2. External
 - 2.1. Legal : User has to submit national ID
 - 2.2. Legal : User has to accept terms and conditions related to the hotel

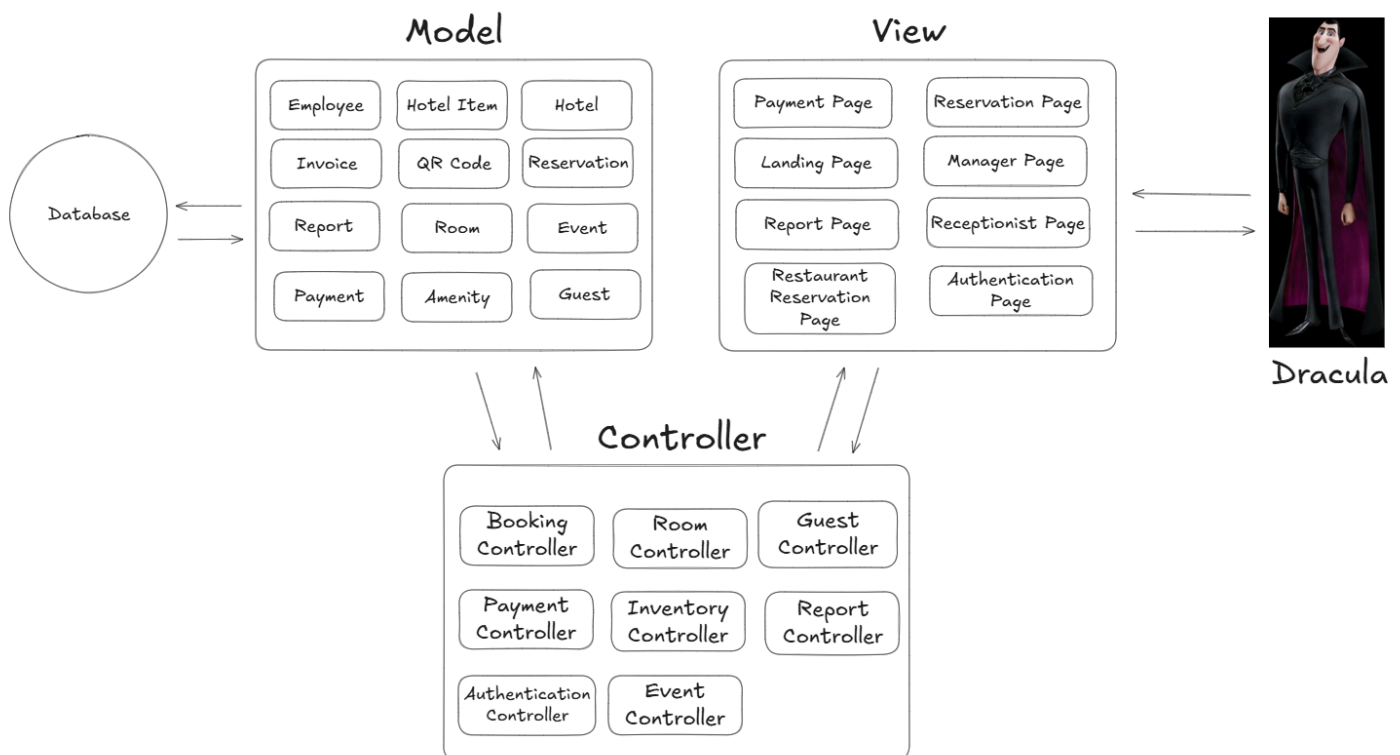
SECTION 2 : CONCEPTUAL DESIGN

Creational Patterns

The Creational Design pattern that will be used in the system is **Singleton**, using a Singleton in the **Hotel Class** ensures that certain resources or services are globally accessible and uniquely instantiated.

Architectural Patterns

The Architectural pattern that will be used in the system is **Model View Controller (MVC)**, using a MVC in this system ensures that smooth communication happens between different modules and introduces the separation of concerns principle.



SECTION 3 : SOFTWARE DECOMPOSITION

This section contains all the different types of decomposition that could be carried out with the hotel software system.

SECTION 3.1 : FUNCTIONAL DECOMPOSITION

Guest/User Functions

1. Booking
 - 1.1. Room filtering : user should be able to search and filter for rooms he prefers ex: sea view , double room
 - 1.2. Check room availability : user should be able to check if the room he wishes for is available or not before booking to prevent double booking.
 - 1.3. Payment Function: user should be prompted to enter credit card details and confirm payment amount and method
 - 1.4. Receive payment information : user should receive a pdf that includes his transaction date ,payment method,and booking details.
 - 1.5. Updating customer info : user should be able to update his personal info
 - 1.6. Cancel requests : user should be able to cancel his bookings at a specific time range and get a refund
 - 1.7. Rescheduling : system checks if the reservation is eligible for rescheduling and then checks for the availability of the new dates . The booking details are updated with the new dates and any changes in room type.

2. Services

- 2.1. Meal plan : user should be able to decide his meal plan for the stay
ex: all inclusive , half board ,....
- 2.2. Each user is sent a unique QR that allows them to enter restaurants
- 2.3. Spa resort : user should be able to view spa section and book spa sessions
- 2.4. Extra beds : user should be able to request an extra bed
- 2.5. Late check-in/check out : user should have the option to be able to check in or check out late

3. User Profile

3.1. Profile Creation & Login

- 3.1.1. User should create their own profile before being able to book their first booking
- 3.1.2. The registration form requires essential details such as name, email address, phone number, and a password.
- 3.1.3. The system should validate the provided details to ensure the uniqueness of identifiers (e.g., email, phone number)
- 3.1.4. Users can log in to the system using their registered email, phone number, or another identifier and password

3.2. Booking History

- 3.2.1. Bookings are tracked with detailed information, including check-in and check-out dates, total amount paid, room type,

and the specific branch the guest has stayed in , all of which are displayed in the user's profile.

3.2.2. Profile displays any current stays with details like in/out dates,room type,special requests /preferences (if any) and additional charges that may be added to the final bill, such as extra services

3.3. Loyalty Points

3.3.1. Loyalty points are updated after each stay ,the system should display the total points accumulated

3.3.2. The available redeemable rewards are displayed

Staff Functions

1. Keep track of rooms each staff member is responsible for: staff should be able to access the rooms that need housekeeping and additional services.
2. Update the hotel stock and issue a warning if anything is missing: staff should be able to update the stock to identify missing and low stock items, giving notice to the management to restock them.
3. Check in guests and check them out: staff should be able to help users check in and out to their assigned rooms and issue bills.
4. Update attendance of staff: staff should handle leave requests, track attendance, and manage shift changes.
5. Issue bills for guests: staff should be able to issue bills with the room charges and additional on stay charges.

Manager Functions

1. Generate feedback and financial reports : Managers should be able to interact with the system to generate reports.
2. View User booking details by Name/National id : Manager should be able to view user booking details in case of emergencies or any misunderstanding in order to fix any problem.
3. View hotel Finances such as current balance , monthly and yearly income , and expenses. This will allow managers to make decisions regarding finances.
4. Generate reports related to the number of guests per month and returning guests. This will help managers evaluate decisions regarding promotions , offers, and events.
5. View current staff details and sort staff by job title and department. This can help managers check if they are over or under staffed.

SECTION 3.2 : OBJECT ORIENTED DECOMPOSITION

Hotel Class

Attributes

- Room[] rooms
- Staff[] staff
- Manager[] managers
- String[] details

Functions

- InitializeDatabase()
- AddRoom()
- RemoveRoom()
- AddManager()
- RemoveManager()

Guest Class

Attributes

- Name
- Email
- PhoneNumber
- NumberOfVisits

- Age
- RoomNumber

Functions

- BookRoom()
- CancelBooking()
- RescheduleBooking()
- Payment()
- AddServices()
- AddRating()
- RoomFilter()
- UpdateProfileDetails()
- UpdateRoom()
- CreateProfile()

Employee Class

Attributes

- Name
- Age
- Phone Number
- ID
- Start Date
- Hours
- Salary

Functions

- GenerateContract()
- AssignHours()
- AssignSalary()

Manager Class

Attributes

- Name
- Age
- Number
- Position

Functions

- UpdateRoom()
- GenerateReport()
- ViewUserBooking()
- ViewHotelFinance()
- ViewStaffDetails()
- AssignStaffDepartment()

Staff Class

Attributes

- Contract
- Department
- String[] Task

Functions

- UpdateHotelStock()
- UpdateAttendance()
- ReadyRoom()

Receptionist Class

Attributes

- Receptionist Title

Functions

- CheckInGuest()
- CheckOutGuest()
- IssueInvoice()

SECTION 4 : DOMAIN DRIVEN DESIGN

This section covers all concepts related to domain driven design and how they are applied into the software system.

Mapping Domain Concepts to Software Concepts

1. Room Management
 - 1.1. Display room details like the type and price
 - 1.2. Tracks status of rooms and updates room availability
2. Customer Account Management
 - 2.1. Manage usernames and passwords of accounts
 - 2.2. Manage past bookings and preferences to personalize experience for customers
3. Staff Management
 - 3.1. Assign tasks like housekeeping and maintenance to staff
 - 3.2. Assign group of rooms to individual staff
 - 3.3. Notify staff of certain tasks that need to be done (e.g. room needs to be cleaned)
 - 3.4. Handle salaries of staff
4. Booking System
 - 4.1. Allows users to check available rooms and make bookings
 - 4.2. Allows users to cancel bookings
5. Payment Gateway

5.1. Handles payment processing when booking rooms

6. Feedback Management

6.1. Allows customers to add reviews and rate their experience

Entities

1. Guest Entity
2. Hotel Entity
3. Employee Entity
4. Room Entity
5. Reservation Entity
6. Payment Entity
7. Hotel Item Entity
8. Invoice Entity
9. QR Code Entity
10. Report Entity
11. Event Entity
12. Amenity Entity

Aggregates

1. Guest Aggregate

Root entity: Guest entity

- i. Contains invoice and QR code entities
- ii. Contains loyalty points and contact info value objects

2. Hotel aggregate

Root entity: Hotel entity

- i. Contains hotel items, event and amenity entities

3. Reservation aggregate

Root entity: reservation entity

- i. Contains room and payment entities
- ii. Contains discount , room type and price value objects

Domain Events

1. RoomBookedEvent()

- Triggered when a new room is booked by the user to update validity , carrying valid information like customer details, booking ID , Room booking details .

2. RoomBookingConfirmationEvent()

- Triggered when the user completes the booking and payment is processed.

3. RoomUpdatedEvent()

- Triggered when a customer updates a room that is booked , carrying valid information like customer details, booking ID , Room booking details .

4. RoomCancellationEvent()

- Triggered when booking is canceled by the user to update the validity of rooms and return user payment.

5. SpaBookedEvent()

- Triggered when a guest books spa services to update validity.

6. RoomRescheduledEvent()

- Triggered when a customer reschedules the room he booked , carrying valid information like customer details, booking ID , Room booking details.

7. RoomAvailabilityEvent()

- Triggered when a new reservation is made or canceled.

8. ProfileCreatedEvent()

- Triggered when a customer creates a new profile.

9. ProfileUpdatedEvent()

- Triggered when a customer updates his profile.

10. RoomAssignedForHousekeepingEvent()

- Triggered when a room is assigned to a housekeeper to clean.

11. StockUpdatedEvent()

- Triggered when hotel inventory is restocked.

12. GuestCheckedInEvent()

- Triggered when a guest is checked into the hotel.

13. GuestCheckedOutEvent()

- Triggered when a guest is checked out of the hotel.

14. InvoiceForGuestsEvent()

- Triggered when an invoice is created for a guest with the outstanding fees.

15. ReportIssuedEvent()

- Triggered when a manager creates a report for finance or feedback.

Domain Services

Domain Services perform actions that trigger domain events. In this software system , the domain events are listed as follows:

1. BookingService

- Responsibilities
 - i. Handle Room booking / cancellations, rescheduling or updates
 - ii. Validate room availability
- Triggers : { RoomBookedEvent, RoomUpdatedEvent, RoomAvailabilityEvent, RoomBookingConfirmationEvent, RoomRescheduledEvent, RoomCancellationEvent }

2. RoomService

- Responsibilities
 - i. Manage room assignments, housekeeping schedules, and availability.
- Triggers : { RoomAssignedForHousekeepigEvent, RoomAvailabilityEvent }

3. GuestProfileService

- Responsibilities
 - i. Manage guest profiles, including creation, updates, and access logging.
- Triggers : {ProfileCreatedEvent, ProfileUpdatedEvent, GuestDetailsAccessedEvent}

4. PaymentService

- Responsibilities
 - i. Handle payment processing for reservations, amenities, and invoices.

- ii. Validate payment information

- Triggers : {InvoiceForGuestsEvent}

5. SpaService

- Responsibilities

- i. Handle spa bookings and schedule spa services.

- Triggers : {SpaBookedEvent}

6. InventoryService

- Responsibilities

- i. Manage inventory for hotel items and amenities.

- ii. Update Stock levels and notify about low stock

- Triggers : {StockUpdatedEvent}

7. CheckInOutService

- Responsibilities

- i. Handle guest check-in and check-out processes.

- ii. Validate reservations and generate invoices.

- Triggers : {GuestCheckedInEvent, GuestCheckedOutEvent, InvoiceForGuestsEvent}

8. ReportingService

- Responsibilities

- i. Generate reports for financials, guest feedback, or other administrative tasks.

- Triggers : {ReportIssuedEvent}

9. EventManagementService

- Responsibilities
 - i. Manage hotel events and related resources
- Triggers : {EventScheduledEvent, EventCancelledEvent}

Value objects:

1. Room Type: Describes the type of room as in single, double, suite
2. Price: Holds pricing details and currency, shared across various contexts as in room rate, service cost
3. Discount: Represents a discount on a booking or service.
4. Loyalty points: Represents the number of loyalty points a customer has.
5. Contact Information: Stores customer or staff contact information in a standard format.

Architectural Patterns

The architectural pattern that will be used will be the Model-View-Controller(MVC) architectural pattern.

MVC splits the system into three components: Controller , Model , and View.

In the case of this system, view pages will be the websites related to the different functionalities of the hotel system such as booking page , login page , etc.

The controller will handle the request flow between the model and the view. In this software system, there will be different controllers such as the booking , room , guest controller.

The model corresponds to the data related logic of the system (databases , queries, etc.). In this software system there will be different models such as an employee model , a hotel model , and a guest model.