

# *Hotel Reservation System*

## Requirements Analysis Document

**Team:** Tech Travel Titans

**Team Members:** Muskanmeet Kaur (219633361)

Sahib Deep Singh (219170646)

Mithunan Sivagnanam (215770472)

Saba Aafreen (217945825)

### Table of Contents

<b>Introduction.....</b>	<b>2</b>
<b>Purpose of the system.....</b>	<b>2</b>
<b>Scope of the system.....</b>	<b>2</b>
<b>Objectives and success criteria of the project.....</b>	<b>3</b>
<b>Definitions, acronyms, and abbreviations.....</b>	<b>4</b>
<b>Current System.....</b>	<b>5</b>
<b>Proposed System.....</b>	<b>6</b>
<b>Overview.....</b>	<b>6</b>
<b>Functional requirements.....</b>	<b>6</b>
<b>Nonfunctional requirements.....</b>	<b>8</b>
Usability.....	8
Reliability.....	8
Performance.....	9
Supportability.....	9
Implementation.....	9
Interface.....	10
Packaging.....	10
Legal.....	10
<b>System models.....</b>	<b>10</b>
Scenarios.....	10
Use case model.....	13
Use case descriptions.....	14
Activity Diagram.....	24
State Machine Diagram.....	25
<b>Glossary.....</b>	<b>26</b>

## Introduction

### Purpose of the system

The hotel industry is built on providing outstanding experiences to guests, and reservations are an important part of this. The lack of a consistent and integrated infrastructure to support the booking of spaces across platforms is one of the major problems the hotel reservation industry is facing. The development and planning of individual hotels and reservation services has resulted in a fragmented, uneven, and duplicated landscape. Online travel agents (OTAs), hotel operators and hotel guests face challenges because of this situation.

Since the reservation platforms vary in infrastructure, users get a different experience with each use of a new platform. The result is inefficient hotel management, since a lot of money needs to be invested in a system built from scratch for each facility. Without a clear channel, online travel agencies must work with multiple systems independently to provide a smooth booking experience. Additionally, the fragmented nature of the system limits the possibility of collaboration and consistency between various platforms.

The purpose of the proposed system is to provide the hospitality industry with a standardized, modular hotel reservation system to address these issues. This initiative will streamline the customer experience in hotels, save on hotel management and development costs, establish a unified system for booking support and provide an integrated platform for online travel agencies, individual users, and hotel personnel.

### Scope of the system

The Hotel Reservation System project involves creating, deploying and continuously refining a customized, modular reservation system for the users. This project seeks to deliver a complete and user-friendly system that improves the efficiency of hotel reservations. The system will use state-of-the-art technology to increase efficiency, improve customer experience and provide a solid foundation for hotel reservation. A secure payment gateway, analytical vision and environmentally friendly activities are among the highlights. To ensure that the project succeeds, objectives are defined, quantifiable, attainable, relevant and time-bound.

The project deliverables are as follows:

- Fully functional hotel reservation system.
- Extensive documentation, including system architecture, digital user manual, and maintenance instructions.
- Training materials for staff and support staff.

There are some exclusions that need to be taken into consideration:

- Integration with external systems or services other than the stated PMSs and payment gateways is not included.
- The project does not include any unique modification of the system for hotels beyond the basic elements.

## Objectives and success criteria of the project

Objectives of the hotel reservation system project include, but are not limited to:

- Creating a standardized hotel reservation system that will provide consistent infrastructure and user experience across all hotels' reservation services.
- Providing an infrastructure with specific tools to efficiently manage hotel operations, including registration, booking, managing reservations, check-ins & check-outs, etc.
- Reducing the learning curve involved in navigating between different reservation platforms, this will be done by providing a smooth and seamless reservation experience for users throughout the process.
- Focusing on interoperability in developing the system, allowing better integration with related software tools used in the industry, such as PMSs.
- Providing a secure and smooth online payment transaction portal during the reservation process while maintaining the payment processing and security requirements of the industry.
- Using analytical techniques to gain insights from reservation data, such as occupancy rates, cancellation rates, booking trends, guest demographics, customer satisfaction, etc. This allows hotels to make informed decisions, monitor operations, and improve the overall efficiency of the installation process.
- Prioritizing sensitive visitor data, financial transactions, and system integrity security. Implement strong security measures to prevent data breaches, unauthorized access, and other potential risks.
- Developing a platform in the system to encourage collaboration and networking and to support successful and innovative ways to share best practice insights between hotel brands and stakeholders.
- Enhancing sustainability by incorporating features that support ecological activities, such as paperless bookings, etc.

We are going to measure the system's success using SMART criteria. It stands for Specific, Measurable, Achievable, Relevant, and Time-bound:

- SPECIFIC - To achieve a standardized reservation procedure strategy and user interface within a four-month time constraint.

- SPECIFIC - Over six months, ensure that 70% of hotel guests participate in the post-stay survey, resulting in a robust and representative sample for meaningful data collection.
- MEASURABLE - To measure a 15% reduction in average reservation time compared to previous systems.
- MEASURABLE - To successfully complete 85% of online transactions through the reservation system.
- ACHIEVABLE - Achieve an acceptance rate of 75% for environment friendly services and practices in hotels.
- ACHIEVABLE - Establish seamless integration with three PMSs to start, taking into account each system's compatibility and requirements, to facilitate data exchange.
- RELEVANT - Get feedback from a variety of users in the post-implementation phase that emphasize the importance of inclusion and accessibility.
- RELEVANT - Implement real-time reporting tools in the reservation process to give hotel admin immediate access to key performance indicators (KPIs) including occupancy, booking trends and revenue data. Ensure that at least 90% of hotels actively implement this real-time reporting service over the next six months, reducing hotels' decision-making time by 20%.
- TIME-BOUND - Critically analyze reservation data and propose the first set of actionable strategies within one month after system deployment.
- TIME-BOUND - Conduct a full security audit and fix any vulnerabilities found during the initial installation of the system within the first 30 days.

## Definitions, acronyms, and abbreviations

1. PMS (Property Management System): Software systems for basic hotel management functions such as reservation, check-in, check-out, and payment.
2. OTA (Online Travel Agency): A digital platform that allows online booking of travel services, including hotel reservations.
3. API (Application Programming Interface): Tools/platforms that allow different types of softwares, applications, and programming languages to interact with each other.
4. KPI (Key Performance Indicator): These are quantifiable metrics that assist organizations in measuring and assessing how well they have performed in each business goal.
5. UI (User Interface): This is the point at which the user communicates with the system.

## Current System

The issue with the current system is the lack of communication and collaboration between different parties involved in the booking process of a hotel stay which has resulted in lower customer satisfaction and user experience. Along with that, there isn't a system that is commonly used between all parties (OTAs, hotel guests, and hotel operators) so that everyone is in sync with the data regarding the guests and their hotel reservations.

There are different ways guests can book a hotel stay: directly calling the hotel to reserve, through an online travel agency website such as Expedia, or going directly to the hotel and reserving on the same day as their arrival. Given this complexity, it is difficult to keep track of the reservations and this results in reservation errors. There are challenges such as overbooked rooms, guests being charged twice for the same room, and even empty rooms not being assigned to other guests because of last minute cancellations. There is also a practice called "walking a guest" and it's used when a hotel is overbooked and the hotel notifies the guest about their reservation being cancelled but accommodates them with a hotel room in another hotel nearby, sometimes accommodating for transportation too. One of our aims is to avoid this situation which would benefit both the customer and the hotel by increasing customer satisfaction and reducing hotel expenses. On one side, we have overbooked hotel rooms and then on the other we have vacant rooms that were left unassigned to incoming guests because previous guests cancelled. Cancellations are bound to happen but the issue starts when accommodation isn't provided as soon as possible. This is why real-time updates are very necessary for hotel operators and OTAs to stay in sync with the guests. Sometimes guests lose their reservation because they receive many confirmation numbers because of going through third-party reservation sites. Guests are also sometimes charged double the amount for the same room because of different parties that manage the reservations, or the payment never got through the first time. Incorrect or incomplete data is passed on from OTAs. When guests arrive, the check-in process is another challenge, hence the long wait times. During the stay, many guests complain about their rooms not being cleaned prior to their arrival and unfortunately the blame goes to the housekeeping employees but the real issue is disorganized management of the system hotels follow. Thus there should be a way for housekeeping employees to keep up in real-time on rooms they must prioritize for incoming guests. There are several challenges that are mentioned and must be resolved to improve the guest experience and hotel services while increasing the collaboration between different parties during the hotel reservation process.

## Proposed System Overview

The proposed system seeks to revolutionize the hotel industry by introducing a standardized reservation system. This solution aims to solve current problems across platforms due to fragmented infrastructure and inconsistent user experience. The technical objective is to shorten reservation times, assure secure online transactions, reduce hotel-individual development costs and provide a uniform, sophisticated technology. The immediate stakeholders of this system are the hotel owners and management team, online travel agencies (OTAs), individual users or guests, system developers, payment gateway providers, and the Property Management System (PMS) suppliers. There are three user views namely, Guest, Travel Agency, Hotel Staff. Hotel operators rely on the system to simplify reservation operations, while OTAs and individual customers desire a consistent and uniform booking process. The developers play a key role in the design and maintenance of the platform. Payment gateway providers are critical to secure financial transactions, and transactions require integration with the PMS provider. The user interface of the system focuses on providing an intuitive customer experience, thereby reducing the learning curve associated with other platforms. The system components include a user-friendly interface, powerful reservation engine, seamless integration with PMSs, secure payment gateways, analytics tools for data-driven decision making, stringent security measures, a collaboration platform, and features that promote environmental stability. Considering the factors and challenges described, the average timeline expected for setting up and implementing this system is between 11 to 12 months. This timeline includes phases such as project initiation, requirements gathering, system design, development, testing, implementation, and post-implementation improvements.

### Functional requirements

The functional requirements encompass the core features and capabilities of the Hotel Reservation System, ensuring that it meets the needs of stakeholders while remaining within the scope of the project.

#### **1. User Registration and Authentication:**

- Stakeholders must be able to register and create accounts.
- Different user roles such as hotel staff, OTAs, and guests should be supported and authenticated securely.

#### **2. Hotel Listing and Booking:**

- Guests/OTAs should be allowed to search for hotels based on different criterias such as price, location and amenities.
- Guests must be able to reserve rooms for specified dates, and choose their room preferences.
- Detailed information about each hotel, including room types, availability, and pricing, should be shown.

- Users should get a confirmation email after making a successful booking.

### **3. Reservation Management:**

- Hotel employees should be given a dashboard where they can manage (view, modify and cancel) reservations.
- The system should automatically update room availability depending on reservations made.

### **4. Check-In and Check-Out Process:**

- Guests have the option to check in either online or at the hotel's front desk.
- Automated check-out processes should be available, allowing guests to settle bills and provide feedback.
- The system will generate digital key cards or provide QR codes for accessing the rooms.

### **5. Integration with Property Management Systems (PMSs):**

- The system needs to work with PMSs in order to seamlessly synchronize reservation data.
- The integration should be able to handle tasks, like managing room inventory and billing.

### **6. Secure Payment Processing:**

- The system should include payment options, such as credit or debit cards, digital wallets and bank transfers.
- Users should receive payment confirmation along with the receipt, for their transactions.

### **7. Analytics and Reporting:**

- The system should collect and analyze reservation data to generate insights such as occupancy rates, booking trends, and revenue forecasts.
- Hotel staff (managers) should have access to generated reports that can help them make decisions.

### **8. Collaboration Platform:**

- The system must enable communication and collaboration among hotel brands, stakeholders, and industry partners.
- It should have features, such as forums, chat tools and document sharing to encourage interaction.

### **9. Environmental Sustainability Features:**

- The system should promote environment friendly practices, such as paperless bookings and energy-efficient operations.
- Stakeholders should be encouraged to opt-in for eco-friendly services during the booking process.

#### **10. Accessibility and Inclusivity:**

- The system must meet accessibility standards to ensure usability for stakeholders with disabilities.
- Features like text-to-speech, keyboard navigation and high contrast modes should be included.

#### **11. Feedback and Review System:**

- Hotel Guests should be prompted to provide feedback and reviews after their stay.
- The system should aggregate and display reviews to help guests make informed decisions about their future reservation.

### **Nonfunctional requirements**

Nonfunctional requirements describe user-level requirements that are not directly related to functionality. This includes usability, reliability, performance, supportability, implementation, interface, operational, packaging, and legal requirements.

#### **Usability**

- The booking system should facilitate booking with a seamless checkout process by eliminating any distractions, such as advertisements or promos, and false affordances, ensuring that the customer has a smooth experience when booking a hotel stay.
- Accessibility features should be included to ensure that people with different abilities can use the system while complying with applicable accessibility standards.
- The user interface must work on different devices such as computers, tablets, mobile phones, etc., making it accessible to a wide range of users.
- Reduce the learning curve, improve navigation, and provide clear and simple information to increase overall user satisfaction.
- Usability testing should be conducted to ensure that the system is accessible to different users, easy to use, and easy to understand.

#### **Reliability**

- The system must be highly reliable, providing consistent and accurate performance under various conditions.
- The system installation must ensure that concurrent storage is handled reliably, data consistency is maintained, and duplicate or inconsistent clients are not generated.
- Continuous system monitoring, automated error reporting, robust backup and recovery procedures must be in place to quickly resolve potential issues and provide a capable system of reliable storage.
- The system should be designed with failover capability, such as redundant hardware and backup servers, to provide continuous service in case of an unexpected challenge.



- Regular audits and monitoring procedures should be conducted to identify and resolve potential problems such as room availability, price discrepancies, or system failures that could jeopardize the reservation process.
- Automated backup and recovery mechanisms should be developed to protect reservation data and ensure timely restoration of service in case of disaster, system faults or problems.

### Performance

- To avoid delays and assure timely confirmation of guest bookings, the system must work smoothly and automate confirmations within a timeframe of 5 minutes when managing multiple booking locations simultaneously, especially during the popular booking times with mass enrollment.
- The load testing should replicate scenarios where a large number of users are attempting to create a backup simultaneously, helping to identify and resolve any performance bottlenecks that could affect the performance of the system.
- Effective caching techniques should be used to improve data recovery and reduce latency, enabling users to quickly access real-time room availability and pricing information.
- The system must be scalable enough to manage population growth of 7 times than now, and retention, while maintaining optimal performance.

### Supportability

- The hotel reservation system will be monitored regularly with dashboards showing activity of the system.
- Real-time data updates should be available across all parties such as hotel operators, OTAs, and PMS to avoid overbooking and “walking the guest” issues.
- Implement 24/7 customer support to system users.
- Employees must be trained regularly to improve their skills and understanding of the system to better help hotel guests.
- Regular backup of reservation data and implementation of a structured recovery plan to prevent data loss and enhance customer satisfaction.
- Regularly monitor security practices, update them as necessary, and train employees to improve security and protect guest data.

### Implementation

- The hotel reservation system will be implemented using Figma and Adobe XD for design and prototyping of the software.
- Backend-development will be implemented through Java, whereas Front-end development will be done through React. Swift will be used for iOS mobile devices and Java will be used for Android users.
- Implement the API with a REST framework.
- The database management of the reservations inventory will be done by using MySQL.

- Project management will be done using Jira to ensure collaboration and synchronization.

### Interface

- Hotel reservation system must comply with the API provided.
- The system should be able to integrate with the software of the PMS.
- The interface should be compatible with all devices (i.e. mobile devices, monitors, and check-in kiosks).

### Packaging

- Must provide proper documentation for the developers to implement and maintain the software system.
- Digital user manuals are to be provided to hotel operators to successfully learn and use the system software for their daily activities.

### Legal

- Hotels registered with the system must possess up-to-date food handling certifications (includes abiding by the country's nutritional regulations, policies and guidelines outlined under the FDA, also certifying in dietary restrictions)
- Registered hotels must abide by sanitary regulations (includes that hygiene items must be properly cleaned and sanitized for every guest, rooms must be free of pests and rodents, and also includes following sanitary guidelines in the plumbing systems, water supply, trash disposal, air ventilations and heating)
- Hotels are not entirely liable for lost luggage and personal items
- Every registered hotel must ensure workplace safety and labour rights outlined in the federal and local government regulations
- Real estate licensing should be provided by the Property Management System
- Hotel reservation system will provide PCI Compliant (Payment Card Industry) secure payment system to enable safe and secure payments for reservations.

### System models

#### Scenarios

#### 1. User Registration and Authentication:

- a. Scenario 1: Alice, a new hotel manager, registers for an account using her email and creates a password.
- b. Scenario 2: Bob, a guest, logs into the system using his social media account credentials.

## **2. Hotel Listing and Booking:**

- a. Scenario 1: Charlie searches for hotels in Paris, filters the results, and books a room at Hotel XYZ.
- b. Scenario 2: David, a business traveler, explores hotel options near his conference venue and books a room at Hotel ABC.

## **3. Reservation Management:**

- a. Scenario 1: Alice, the hotel manager, manages reservations for the day, updates the status of a reservation to "checked-in," and cancels a duplicate reservation.
- b. Scenario 2: Alice adjusts room availability based on maintenance schedules and special events.

## **4. Check-In and Check-Out Process:**

- a. Scenario 1: Grace, a guest, checks in through the system's check in portal and receives a digital QR code on her smartphone to gain access to the booked room.
- b. Scenario 2: Henry, another guest, does an online check out using the system and drops the key in the check-out lock box at the hotel door.

## **5. Integration with Property Management Systems (PMSs):**

- a. Scenario 1: Alice, the hotel manager, integrates the system with the existing PMS used at her hotel chain by signing in through the system's portal linked to different PMSs.
- b. Scenario 2: Jack, a PMS provider (vendor), collaborates with the system developers to ensure compatibility.

## **6. Secure Payment Processing:**

- a. Scenario 1: Karen, a guest, pays for her reservation using her credit card through the system's secure payment gateway.
- b. Scenario 2: Liam, another guest, opts for a bank transfer payment method.

## **7. Analytics and Reporting:**

- a. Scenario 1: Alice, the hotel manager, accesses real-time occupancy rates and revenue forecasts through the system's analytics dashboard.
- b. Scenario 2: Nick, a data analyst, exports reservation data from the system for further analysis.

## **8. Collaboration Platform:**

- a. Scenario 1: Claire, a hotel brand representative, participates in an online forum hosted by the system and posts about her chain's customer service.

- b. Scenario 2: Peter, a sustainability consultant, collaborates with hotels through the system's messaging capabilities.

**9. Environmental Sustainability Features:**

- a. Scenario 1: Quinn, a guest, opts for paperless check-in and check-out through the system's mobile app.
- b. Scenario 2: Rachel, a hotel staff member, utilizes the system to monitor and track the hotel's adoption of eco-friendly practices, ensuring compliance and facilitating continuous improvement in sustainability efforts. For example: She records the usage of recycled materials, the adoption of energy-efficient appliances, and the reduction in water consumption.

**10. Accessibility and Inclusivity:**

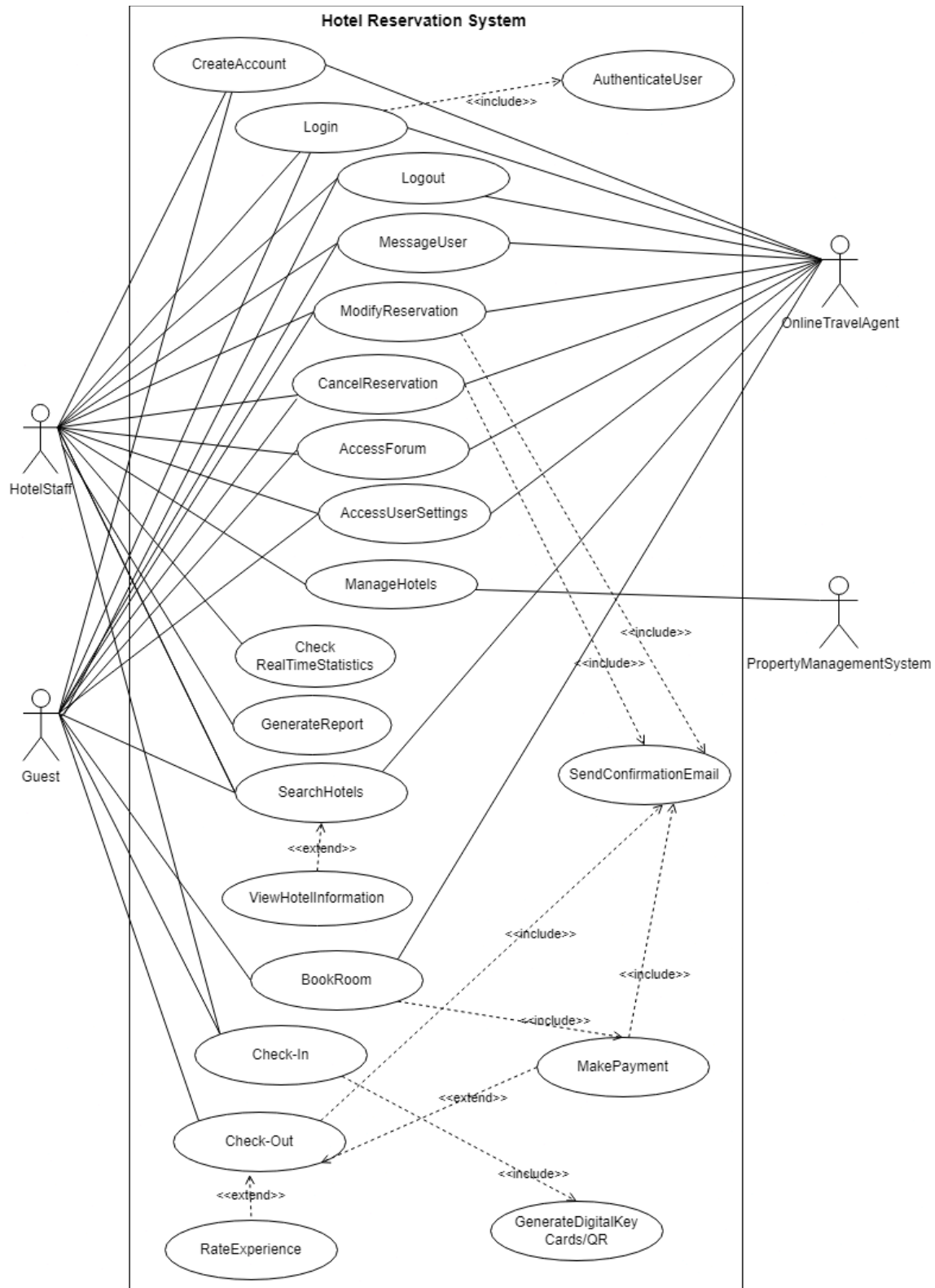
- a. Scenario: Tom, a guest with mobility impairments, books an accessible room through the system's interface.

**11. Feedback and Review System:**

- a. Scenario 1: Wendy, a recent hotel guest, receives a post-stay survey via email from the system and provides feedback.
- b. Scenario 2: Xavier, a prospective guest, reads reviews and ratings submitted by previous guests through the system's interface.

## Use case model

[Link to the Use case Model](#)



## Use case descriptions

<i>Use case name</i>	CreateAccount
<i>Participating Actors</i>	<input type="checkbox"/> Guest <input type="checkbox"/> HotelStaff <input type="checkbox"/> OnlineTravelAgent
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user starts the account creation process.</li> <li>2. The system prompts the user to provide the necessary registration information.</li> <li>3. User enters the required information.</li> <li>4. The system checks and creates a user account.</li> </ol>
<i>Entry condition</i>	User wants to create an account.
<i>Exit condition</i>	User successfully registers an account on Hotel Reservation System

<i>Use case name</i>	Login
<i>Participating Actors</i>	<input type="checkbox"/> Guest <input type="checkbox"/> HotelStaff <input type="checkbox"/> OnlineTravelAgent
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. User opens Hotel Reservation system.</li> <li>2. System displays login and logout page.</li> <li>3. User clicks on login.</li> <li>4. System ask for username and password.</li> <li>5. User enters username and password.</li> <li>6. The system initiates the "AuthenticateUser" use case to verify the entered username and corresponding password.</li> <li>7. System displays home page after login success.</li> </ol>
<i>Entry condition</i>	User wants to login into the system. User has a registered account on the system.
<i>Exit condition</i>	User successfully logged into the system.

<i>Use case name</i>	AuthenticateUser
<i>Participating Actors</i>	<input type="checkbox"/> Guest <input type="checkbox"/> HotelStaff <input type="checkbox"/> OnlineTravelAgent
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The system retrieves the user's account information from the database.</li> <li>2. The system verifies the entered credentials against the stored information.</li> <li>3. If the user is authenticated successfully: The system checks the user's role and provides a dashboard specific to the user's role (Guest, HotelStaff, OnlineTravelAgent, also includes roles within HotelStaff like administrator, manager, employee)</li> <li>4. If the authentication fails, the system notifies the user and prompts them to re-enter their credentials.</li> </ol>
<i>Entry condition</i>	User wants to log in to the system.
<i>Exit condition</i>	User successfully logs in and is directed to the appropriate dashboard based on their role.

<i>Use case name</i>	Logout
<i>Participating Actors</i>	<input type="checkbox"/> Guest <input type="checkbox"/> HotelStaff <input type="checkbox"/> OnlineTravelAgent
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. User opens Hotel Reservation system.</li> <li>2. System displays login and logout page.</li> <li>3. User clicks on logout.</li> <li>4. System will end the user session, logging the user out of the system.</li> <li>5. System redirects to home page.</li> </ol>
<i>Entry condition</i>	User is logged into the system. User wants to logout off the system. User has a registered account on the system.
<i>Exit condition</i>	User successfully logged out of the system.

<i>Use case name</i>	SearchHotels
<i>Participating Actors</i>	<input type="checkbox"/> Guest <input type="checkbox"/> HotelStaff <input type="checkbox"/> OnlineTravelAgent
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The system provides a search interface with several parameters, including price, location, and facilities.</li> <li>2. The user provides search parameters, such as preferred price range, location, and facilities.</li> <li>3. The system searches the database using the information provided and returns a list of hotels that fit the search.</li> <li>4. The user has an option to view detailed information about a specific hotel by selecting it from the search results.</li> <li>5. If the user clicks on a specific hotel, the system initiates the 'ViewHotelInformation' use case.</li> </ol>
<i>Entry condition</i>	User is logged into the system. Guest wants to search for hotels OR Hotel staff want to check hotel information.
<i>Exit condition</i>	User viewed list of hotels OR User selects a hotel and views detailed information OR User exits the search interface without selecting a hotel.

<i>Use case name</i>	BookRoom
<i>Participating Actors</i>	<input type="checkbox"/> Guest <input type="checkbox"/> OnlineTravelAgent
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. User clicks on the “Book Now” button.</li> <li>2. The system displays a booking page where the guest can select the desired room type, booking dates, and any extra preferences.</li> <li>3. User confirms the booking.</li> <li>4. System adjusts the room status to "Reserved".</li> <li>5. If selected room is available, system asks user to select a payment method.</li> <li>6. User selects a payment method, credit/debit cards/digital wallets/bank transfers or forward payment to guest.</li> </ol>
<i>Entry condition</i>	User is logged into the system. Guest wishes to reserve a room after reviewing hotel information OR OnlineTravelAgent wishes to reserve a room for their client.
<i>Exit condition</i>	User successfully booked a room.

<i>Use case name</i>	MakePayment
<i>Participating Actors</i>	<input type="checkbox"/> Guest
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. User confirms payment method (credit/debit cards/digital wallets/bank transfer).</li> <li>2. The user inputs payment information.</li> <li>3. The system verifies the payment.</li> <li>4. The system triggers the "SendConfirmationEmail " use case to send a confirmation message to the guest.</li> </ol>
<i>Entry condition</i>	User is logged into the system. The booking process has been completed, and the user has selected a payment method.
<i>Exit condition</i>	Transaction has been successfully processed. Confirmation of payment has been sent to the user.



<i>Use case name</i>	Check-In
<i>Participating Actors</i>	<input type="checkbox"/> Guest <input type="checkbox"/> HotelStaff
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. When a guest arrives at the hotel, they go to the front desk or use the system to start the online check-in process.</li> <li>2. When checking in online: <ul style="list-style-type: none"> <li>• The user adds necessary details such as name, reservation number, and any other data needed to verify their identity.</li> <li>• After confirming the information, the system initiates the "GenerateDigitalKeyCards/QR" use case to create a digital key or send a QR code so that users can enter the room.</li> </ul> </li> <li>3. When check-in at the front desk: <ul style="list-style-type: none"> <li>• The guest gives the hotel staff their identity and reservation information.</li> <li>• After confirming the details, the hotel staff initiates the "GenerateDigitalKeyCards/QR" use case to either generate a digital key or provide a physical key card.</li> </ul> </li> <li>4. System modifies the room status to "Occupied," the guest is granted entry to the reserved room.</li> </ol>
<i>Entry condition</i>	User is logged into the system. The guest arrives for check-in, whether in person or online.
<i>Exit condition</i>	Guest successfully checks in and is granted access to the room.

<i>Use case name</i>	Check-Out
<i>Participating Actors</i>	<input type="checkbox"/> Guest
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The guest begins to check-out by selecting check-out option inside the system.</li> <li>2. The system displays the guest's reservation information as well as any charges that were billed.</li> <li>3. If there are outstanding charges, the guest analyzes the charges, chooses a payment option, and initiates the "MakePayment" use case to complete the transaction.</li> <li>4. The system generates a digital receipt and triggers the "SendConfirmationEmail" use case to send a confirmation message to the guest.</li> <li>5. System adjusts the room status to "Vacant".</li> <li>6. After completing the check-out process, the guest has the option to rate their experience at the particular hotel using the 'RateExperience' use case.</li> <li>7. If guest has physical key, he/she drops the key in checkout lock box at hotel door.</li> </ol>
<i>Entry condition</i>	User is logged into the system. The guest wants to check out after finishing their stay
<i>Exit condition</i>	Guest successfully checks out, and the room becomes available to the next guest.

<i>Use case name</i>	CheckRealTimeStatistics
<i>Participating Actors</i>	<input type="checkbox"/> HotelStaff
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. User logs into the system.</li> <li>2. The system shows a dashboard with current data on revenue, occupancy rates, and other key performance indicators (KPIs) as well as hotel performance.</li> <li>3. Based on the most recent reservation data, the system dynamically refreshes the statistics.</li> <li>4. The user examines the real-time data to learn more about how the hotel is performing right now.</li> </ol>
<i>Entry condition</i>	User is logged into the system. User wants to check real-time statistics.
<i>Exit condition</i>	Real-time statistics are successfully viewed and analysed by the user.

<i>Use case name</i>	ModifyReservation
<i>Participating Actors</i>	<input type="checkbox"/> HotelStaff <input type="checkbox"/> OnlineTravelAgent <input type="checkbox"/> Guest
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. Click on reservations.</li> <li>2. The system displays a list of reservations made by guests.</li> <li>3. User selects a specific reservation to modify.</li> <li>4. The user makes any necessary changes, such as updating status, dates or type of room.</li> <li>5. The system confirms the availability of the requested modifications and updates the reservation information accordingly.</li> <li>6. If a modification causes a change in price, the system recalculates the charges and add it as outstanding charges with user account.</li> <li>7. The system triggers the "SendConfirmationEmail" use case to send a confirmation message to the guest.</li> </ol>
<i>Entry condition</i>	User is logged into the system. User wants to modify reservation.
<i>Exit condition</i>	Reservation modified successfully.

<i>Use case name</i>	CancelReservation
<i>Participating Actors</i>	<input type="checkbox"/> HotelStaff <input type="checkbox"/> OnlineTravelAgent <input type="checkbox"/> Guest
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. Click on reservations.</li> <li>2. The system displays a list of reservations.</li> <li>3. User selects a specific reservation to cancel.</li> <li>4. The user cancels the selected reservation.</li> <li>5. The system changes the reservation status to "Cancelled"</li> <li>6. If applicable, the system returns the payment to the guest's account.</li> <li>7. The system triggers the "SendConfirmationEmail" use case to send a confirmation message to the guest.</li> </ol>
<i>Entry condition</i>	User is logged into the system. The guest wants to cancel an existing reservation OR OnlineTravelAgent wants to cancel a reservation for their client OR HotelStaff find room unavailable
<i>Exit condition</i>	The reservation successfully cancelled, and the system updated the reservation details.

<i>Use case name</i>	ManageHotels
<i>Participating Actors</i>	<input type="checkbox"/> HotelStaff <input type="checkbox"/> PropertyManagementSystem
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. Clicks on "Hotels" button.</li> <li>2. The system presents a list of available hotels.</li> <li>3. To update hotel: <ul style="list-style-type: none"> <li>• The user selects a specific hotel to manage.</li> <li>• System displays existing details of hotel.</li> <li>• The user can update information, upload images, and manage facilities, rooms' availability and room rates.</li> </ul> </li> <li>4. To add new hotel: <ul style="list-style-type: none"> <li>• User clicks on "Add New".</li> <li>• The user inputs information about the new hotel, such as its name, address, phone number, and rooms inventory.</li> <li>• After verifying the data, the system saves the newly opened hotel.</li> </ul> </li> <li>5. The system dynamically updates the hotel information and settings, assuring accuracy and consistency throughout the reservation process.</li> </ol>
<i>Entry condition</i>	User is logged into the system. HotelStaff wants to change hotel information or add a new hotel OR PropertyManagementSystem wants to modify hotel information or room inventory.
<i>Exit condition</i>	Hotel details are efficiently managed. User wants to leave the hotel management interface.

<i>Use case name</i>	AccessForum
<i>Participating Actors</i>	<input type="checkbox"/> OnlineTravelAgent <input type="checkbox"/> HotelStaff <input type="checkbox"/> Guest
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. User navigates to the forum section within system</li> <li>2. System displays a list of available forum threads</li> <li>3. The user selects an option to create a new post, edit a post, or delete a post.</li> <li>4. If the user chooses to create a new post, edit or delete post, the system presents an interface to create post, edit, or delete post</li> </ol>
<i>Entry condition</i>	User is logged into the system. User wants to access and interact with online forum.
<i>Exit condition</i>	User completes interaction with forum, either by posting or choosing to exit without posting.

<i>Use case name</i>	GenerateReport
<i>Participating Actors</i>	<input type="checkbox"/> HotelStaff
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user selects the "Generate Report" button.</li> <li>2. The system verifies the user's role.</li> <li>3. If the user is identified as an hotel administrator or manager: <ul style="list-style-type: none"> <li>• The system offers choices for choosing which kind of report to produce, including estimates for revenue, booking patterns, occupancy rates, etc.</li> <li>• The user configures report parameters, such as the time frame, particular hotels, and any other pertinent filters. The system retrieves reservation data based on the provided parameters.</li> <li>• After viewing a preview, the user can export the report in a variety of formats, including PDF and Excel, for additional study or sharing with stakeholders.</li> <li>• The system downloads the report to the user's file system.</li> </ul> </li> <li>4. If the user is not identified as an administrator or manager, the system denies access to the report generation functionality and displays an error message</li> </ol>
<i>Entry condition</i>	User is logged into the system. User wants to generate a report. User has administrative or manager privileges.
<i>Exit condition</i>	Report successfully generated.

<i>Use case name</i>	MessageUser
<i>Participating Actors</i>	<input type="checkbox"/> OnlineTravelAgent <input type="checkbox"/> HotelStaff <input type="checkbox"/> Guest
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The system displays a list of messaging-ready stakeholders, such as administrators, hotel employees, and other pertinent users.</li> <li>2. The user picks a particular stakeholder to communicate with.</li> <li>3. The system displays a message composing interface, where the user can type and send messages.</li> <li>4. User types message and click “Send” button.</li> <li>5. Recipient receives the message and is notified of the incoming communication.</li> </ol>
<i>Entry condition</i>	User is logged into the system. The user wants to send a message to a certain user.
<i>Exit condition</i>	Message sent successfully.

<i>Use case name</i>	AccessUserSettings
<i>Participating Actors</i>	<input type="checkbox"/> OnlineTravelAgent <input type="checkbox"/> HotelStaff <input type="checkbox"/> Guest
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. User navigates to settings section of system</li> <li>2. System displays a list of available settings such as environmental friendly options, accessibility options, user preferences, ability to edit user information (billing, location, name etc).</li> <li>3. The user selects an option to modify their settings</li> <li>4. The system presents an interface to customize their preferences, including environmentally friendly options, accessibility options, user preferences or editing user information (billing, location, name etc).</li> </ol>
<i>Entry condition</i>	User is logged into the system. User wants to access and modify their settings OR User wants to change user information.
<i>Exit condition</i>	User completes modification of their settings and exits the settings section.

<i>Use case name</i>	RateExperience
<i>Participating Actors</i>	<input type="checkbox"/> Guest
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. Guest had clicked the option to provide feedback or ratings after the check-out process</li> <li>2. System displays rating/feedback interface</li> <li>3. Guest provides feedback and/or rating</li> <li>4. The system records the feedback or rating and updates the hotel's information database.</li> </ol>
<i>Entry condition</i>	User is logged into the system. The guest has completed the check-out process. The guest is prompted to rate their experience.
<i>Exit condition</i>	The guest successfully submits their feedback and ratings.

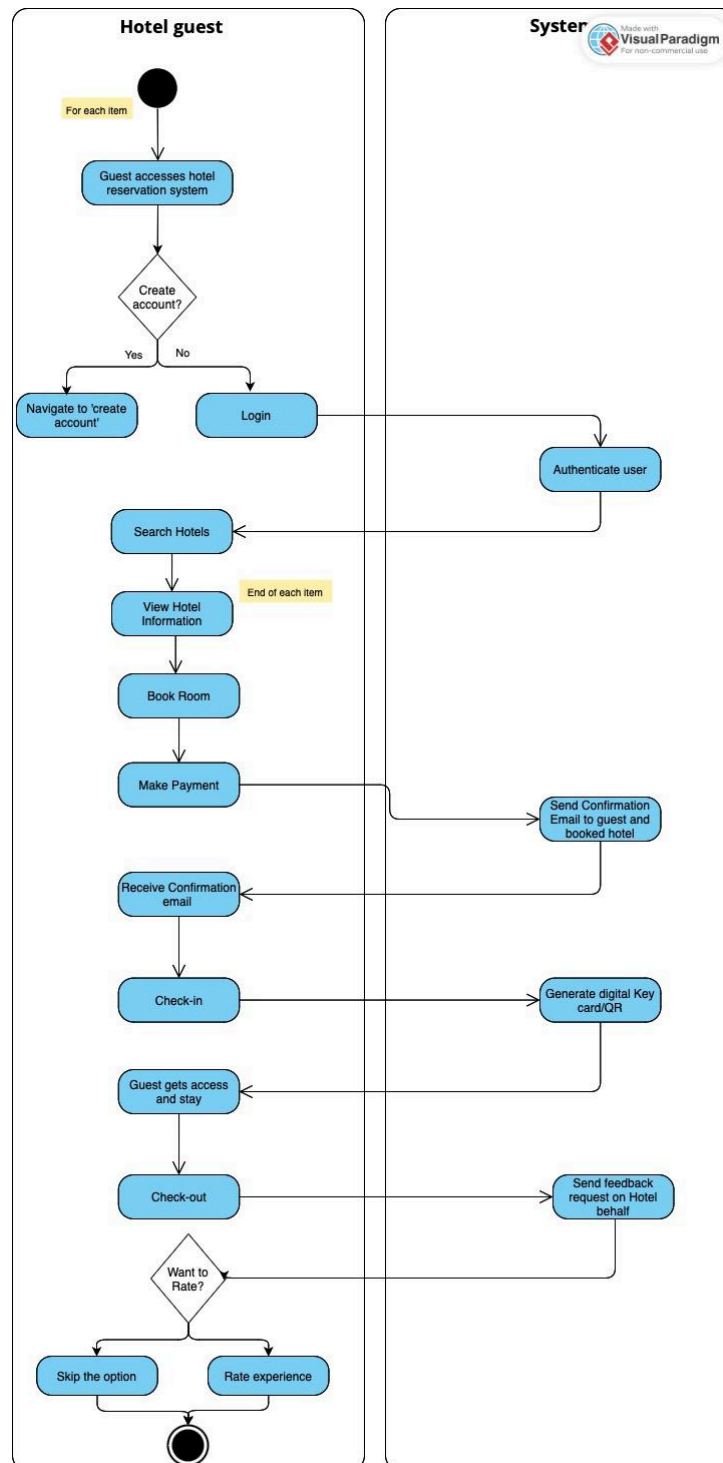
<i>Use case name</i>	GenerateDigitalKeyCards/QR
<i>Participating Actors</i>	<input type="checkbox"/> Guest <input type="checkbox"/> HotelStaff
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. System initiates the generation process upon successful verification of the guest's identity and reservation details.</li> <li>2. If checking in online, the digital key card or QR code is sent to the guest electronically.</li> <li>3. If checking in at the front desk, the digital key card or QR code is provided to the guest by the hotel staff.</li> <li>4. The guest can use it to access their reserved room by either scanning the QR code or presenting the digital key card to the door lock mechanism.</li> </ol>
<i>Entry condition</i>	User is logged into the system. The guest has completed the check-in process, either online or at the front desk.
<i>Exit condition</i>	The guest successfully receives the digital key or QR code and gains access to the reserved room.

<i>Use case name</i>	SendConfirmationEmail
<i>Participating Actors</i>	<input type="checkbox"/> Guest <input type="checkbox"/> OnlineTravelAgent
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The system retrieves the necessary details and user information to compose the confirmation email</li> <li>2. The system generates a confirmation email tailored to the specific action taken by the guest or online travel agent.</li> <li>3. Once the confirmation email is generated, the system sends it to the appropriate recipient(s): <ul style="list-style-type: none"> <li>If the action was performed by a guest: <ul style="list-style-type: none"> <li>• The confirmation email is sent to the email address associated with the guest's account.</li> </ul> </li> <li>If the action was performed by an online travel agent: <ul style="list-style-type: none"> <li>• The confirmation email is sent to the email address provided by the online travel agent during the booking process.</li> </ul> </li> </ul> </li> </ol>
<i>Entry condition</i>	User is logged into the system. The user successfully completed one of the following use cases: <ul style="list-style-type: none"> <li>• ModifyReservation</li> <li>• CancelReservation</li> <li>• MakePayment</li> <li>• Check-Out</li> </ul>
<i>Exit condition</i>	The guest successfully received the confirmation email.

<i>Use case name</i>	ViewHotelInformation
<i>Participating Actors</i>	<input type="checkbox"/> Guest <input type="checkbox"/> HotelStaff <input type="checkbox"/> OnlineTravelAgent
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. User selects hotel from the search results to view detailed information</li> <li>2. The system retrieves detailed information about the selected hotel from the database, including but not limited to: Hotel name, location, room types, pricing, guest rating and reviews</li> <li>3. The system presents the retrieved hotel information to the user</li> <li>4. The user can interact with the displayed hotel information</li> </ol>
<i>Entry condition</i>	User is logged into the system. User has selected a hotel.
<i>Exit condition</i>	User proceeds with other actions like booking User decides to return to the list of search results to explore other hotels User exits the hotel information view without taking further action.

## Activity Diagram

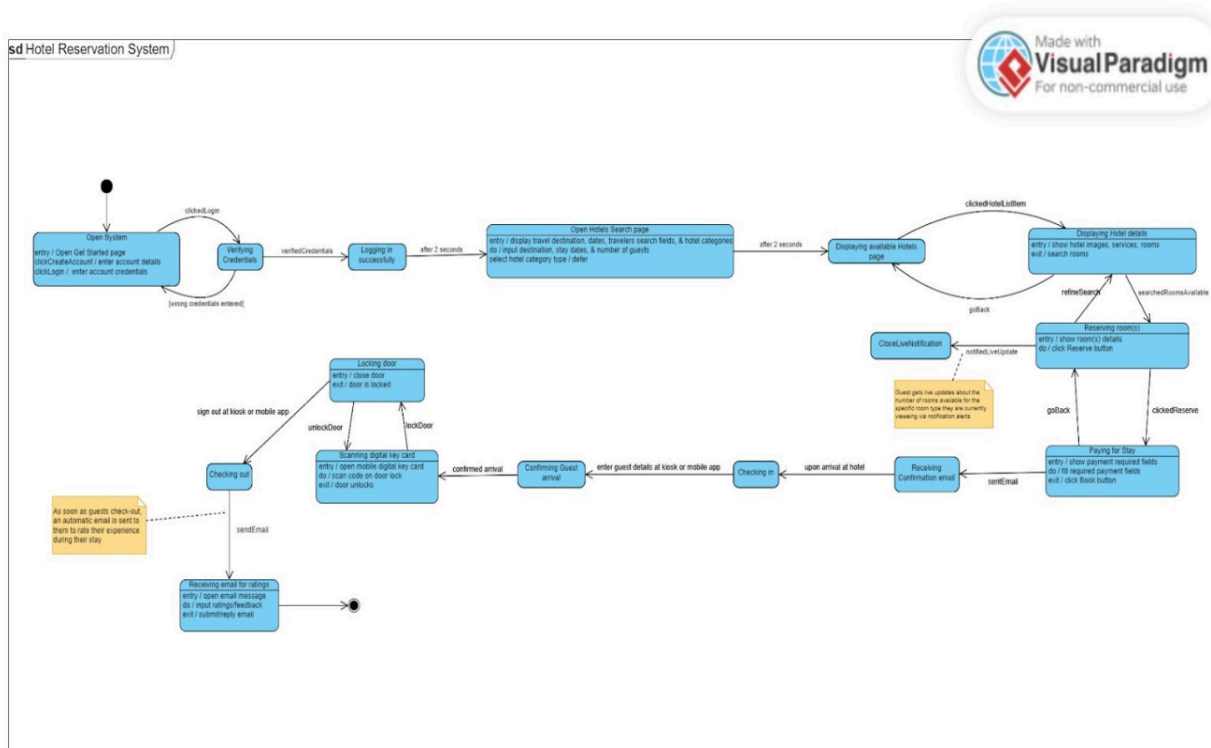
[Link to the Activity Diagram](#)





## State Machine Diagram

[Link to the StateMachine Diagram](#)



## Glossary

### *State Machine Diagram glossary:*

- verifiedCredentials: class that verifies the credentials of the guest, checks whether the username and password are a match to the ones that they created the account with.
- clickedLogin: class that initiates the event when the user clicks the login button on the login page (different from clickLogin - the button on the Get Started page).
- clickedHotelListItem: class that initiates the event when a user clicks on a hotel they like from the hotels availability page. The hotels are available in the destination where the user intends to travel to.
- goBack: class that refers to the previous page.
- refineSearch: filter search results for the rooms.
- searchedRoomsAvailable: initiated event to search the rooms available.
- clickedReserve: the reserve button was clicked and that initiated the event.
- sentEmail: confirmation email was sent to initiate the next state which is to receive the email in inbox.
- unlockDoor: when a digital key card is tapped on the door lock handle then the door is unlocked.
- lockDoor: the door, when closed, locks the door.
- sendEmail: when the email for feedback is sent.