**Requirements Specifications**

**<Project Code>:<P04>**

**<team member names & ids>**

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| **25100235** | **shahrez Faisal** |
| **25100313** | **Muhammad MEHDI** |
| **25100320** | **Muhammad Usman Arshid** |
| **25100015** | **Omar Ibne Sajjad** |
| **24100199** | **Umer inayat** |

|  |  |  |
| --- | --- | --- |
| **Content** | **Totals** | **Obtained** |
| Introduction & system actors | 5 | 5 |
| Use case diagram | 10 | 8 |
| Use case descriptions | 20 | 16 |
| Class diagram | 20 | 19 |
| Sequence diagram | 20 | 16 |
| State diagram | 5 | 5 |
| Non-functional requirements | 5 | 5 |
| Who did what | 5 | 5 |
| Review checklist | 5 | 5 |
| Overall formatting/template | 5 | 5 |
| Late submission penalty | -20 | - |
| GitHub Folder structure penalty | -5 | - |
| **Grand Total** | **100** | **89** |
| **General Comments/Individual Grading:** | | |

**Table of Contents**

[1. Introduction 2](#_Toc1985637570)

[2. System Actors 2](#_Toc1975545362)

[3. Use Cases 3](#_Toc1661975531)

[3.1 Use Case Diagrams 3](#_Toc79463562)

[3.2 Description of Use Cases 3](#_Toc752218123)

[3.2.1 Create account 4](#_Toc506632931)

[3.2.2 Log in 5](#_Toc1222086035)

[3.2.3 Book hotel 5](#_Toc720689251)

[3.2.4 View hotel details 6](#_Toc193926974)

[3.2.5 Search Tourist locations 7](#_Toc2075880826)

[3.2.6 Filter search results 8](#_Toc44168805)

[3.2.7 Live weather updates 10](#_Toc1683172283)

[3.2.8 Leave Hotel Review 12](#_Toc586344773)

[3.2.9 Share trip Details 13](#_Toc779812941)

[3.2.10 Ai based trip planning 14](#_Toc538122844)

[3.2.11 Save favorite locations 16](#_Toc673400116)

[3.2.12 View rent a car 17](#_Toc1286120269)

[3.2.13 View Tourist location Reviews 18](#_Toc1104983416)

[3.2.14 View Restaurants Details 19](#_Toc1382052709)

[3.2.15 View Restaurant Reviews 20](#_Toc1699378661)

[3.2.16 Create Hotel Account 21](#_Toc1070373099)

[3.2.17 Update Hotel records 22](#_Toc1121268085)

[3.2.18 Upload hotel photos 23](#_Toc372078669)

[3.2.19 View customers reviews 24](#_Toc492742796)

[3.2.20 Connect with Customers 25](#_Toc624437338)

[4. Class Diagram 26](#_Toc1435154863)

[4.1 Diagram 27](#_Toc975207412)

[4.2 Description 27](#_Toc580357100)

[5. Sequence Diagrams 27](#_Toc515230203)

[5.1 Use case Name e.g., Withdraw cash 28](#_Toc946157595)

[5.2 Use case Name e.g., Transfer funds 28](#_Toc1455183690)

[6. State Diagrams 29](#_Toc734674479)

[6.1 Diagram details 29](#_Toc1016361175)

[6.2 Diagram 29](#_Toc728256966)

[7. Non-functional Requirements / Quality Attributes 30](#_Toc1562807387)

[8. Who Did What? 30](#_Toc1619604495)

[9. Review checklist 31](#_Toc2119709040)

# Introduction

Our project “Manzil” will be in the form of a mobile application and has two main aspects. The first one is assisting people travelling to northern destinations in several ways, helping them streamline the process. This includes letting users access hotel information/bookings, rental vehicles, travelling companies’ itinerary access, live weather conditions of the roads etc.

The second aspect will be in the form of a guide to major cities (as a starting point) such as Lahore, Karachi and Islamabad. Users will be provided with information and reviews on services such as restaurants, schools, hospitals etc. in these cities, which they might search through with integrated map location radius feature. This mobile application might use ML in a variety of ways to further enhance user experience such as search features, recommendations and chatbots. We are introducing Manzil, a one-stop solution that simplifies tourism by bringing all your travel needs into a single application.

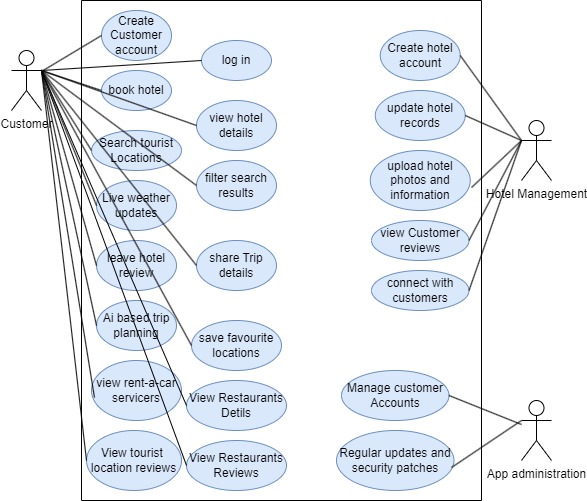
# System Actors

|  |  |
| --- | --- |
| **Actor Name** | **Description** |
| Customer | Customers are the primary users of the app. They can create accounts, log in, search for tourist locations, view details of hotels, hospitals, car rental services, and tourist attractions, book services, and leave reviews for services they’ve used |
| Hotel Management/Administration | Hotel administrators handle hotel-related operations. They can create accounts for their hotels, connect with customers, update room availability, upload hotel details and pictures, and view customer reviews, ensuring smooth customer interaction and bookings. |
| App Administration/Developers | The app administration is responsible for maintaining and managing the backend and overall functionality of the app and resolving any issues related to it. |

# Use Cases

## Use Case Diagrams

[It seems hotel management and administrator can not login? There would be some common use case for different user roles.]



## Description of Use Cases

[**Select 20 most important use cases of your project and create their comprehensive descriptions.**]

### Create account

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-001 |
| **Purpose** | | The customer creates an account to use the app’s features. |
| **Pre-conditions** | | The app must be installed on the customer’s device, and the user must have a valid email address. |
| **Post-conditions** | | The account is created successfully, and the customer is logged in. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | The customer opens the app and selects the "Sign Up" option. | |
|  | The app prompts the customer to enter their email, name, and password. | |
|  | The customer enters the required information. | |
|  | The app sends a verification code to the provided email address. | |
|  | The customer enters the verification code. | |
|  | The system verifies the code, creates the account, and logs the customer in. | |
|  | The use case ends. | |
|  |  | |
|  |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | In step 5, if the customer does not receive the verification code, they can request the system to resend the code. | |
|  | The customer can cancel the account creation at any time and exit the process. | |
| **Step#** | **Exception Paths** | |
|  | In step 4, if the email is already registered, the app displays an error message and requests the customer to use a different email. | |
|  | In step 6, if the verification code is incorrect, the app displays an error and asks the customer to re-enter the code. | |

### Log in

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-002 |
| **Purpose** | | The customer logs in to their account to access app features. |
| **Pre-conditions** | | The user must already have an account with a valid email and password. |
| **Post-conditions** | | The customer is logged in successfully. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The customer opens the app and selects the "Log In" option. | |
| 2. | The app prompts the customer to enter their email and password. | |
| 3**.** | The customer enters their credentials. | |
| 4. | The system checks the credentials and logs the customer in. | |
| 5. | The use case ends. | |
|  |  | |
|  |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | If the customer has forgotten their password, they can request a password reset link via email. | |
| **Step#** | **Exception Paths** | |
| 1. | If the email or password is incorrect, the system displays an error and asks the customer to retry. | |

### Book hotel

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-003 |
| **Purpose** | | The customer books a hotel room for their stay via the app. |
| **Pre-conditions** | | The customer must be logged into their account. |
|  | | Hotel information (availability, prices) must be up-to-date in the system. |
| **Post-conditions** | | The hotel room is successfully reserved for the selected dates |
|  | | The customer receives a booking confirmation. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The customer navigates to the "Hotels" section in the app. | |
| 2. | The customer searches for hotels based on location, check-in, and check-out dates. | |
| 3. | The app displays a list of available hotels with details like pricing and room availability. | |
| 4. | The customer selects a hotel and chooses the room type. | |
| 5. | The app shows the booking summary, including dates, price, and room details. | |
| 6. | The customer confirms the booking. | |
| 7. | The system confirms the booking and sends a confirmation to the customer’s email or within the app. | |
| 8. | The use case ends. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 4, if the selected hotel is fully booked, the app suggests alternative hotels in the same location or nearby. | |
|  | The customer can choose to cancel the booking at any time. [Formatting issue] | |
| **Step#** | **Exception Paths** | |
| 1. | In step 8, if the booking cannot be confirmed due to server issues or hotel unavailability, the system notifies the customer and asks them to try again later. | |

### View hotel details

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-004 |
| **Purpose** | | The customer views detailed information about a hotel, such as room availability, amenities, reviews, and pricing. |
| **Pre-conditions** | | The customer must be logged into their account. |
|  | | Hotel information (availability, prices) must be up-to-date in the system. |
| **Post-conditions** | | The customer views the details of the selected hotel and can proceed with further actions, such as booking a room. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The customer navigates to the "Hotels" section in the app. | |
| 2. | The customer searches for hotels by entering a destination and dates. | |
| 3. | The app displays a list of available hotels. | |
| 4. | The customer selects a hotel from the list. | |
| 5. | The app shows detailed information about the selected hotel, including: room types and availibility,hotel amenities, pricing for rooms, customers reviews, photos of hotels and rooms | |
| 6. | The customer can scroll through the details or select a specific section to view (e.g., reviews or amenities). | |
| 7. | The use case ends when the customer finishes viewing the details. | |
| 8. | The use case ends. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 4, if the hotel is not available or has no details, the app may display a message suggesting alternative hotels in the same location. | |
| **Step#** | **Exception Paths** | |
| 1. | In step 3, if no hotels are available for the selected dates or location, the app displays a "No Results Found" message and offers suggestions for nearby locations or different dates. | |

### Search Tourist locations

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-005 |
| **Purpose** | | The customer searches for tourist destinations and attractions within a specified location or radius. |
| **Pre-conditions** | | The customer must be logged into their account. |
|  | | The system should have data on tourist locations. |
| **Post-conditions** | | The customer is presented with a list of tourist locations matching their search criteria. |
|  | | The customer can select a tourist location to view more details. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The customer navigates to the "Tourist Locations" section in the app. | |
| 2. | The customer enters a destination or selects a location from a list (e.g., city, region). | |
| 3. | The app retrieves and displays a list of tourist locations that match the search criteria. | |
| 4. | The customer scrolls through the results and selects a tourist location. | |
| 5. | The app shows details about the selected tourist location (e.g., description, pictures, reviews). | |
| 6. | The use case ends when the customer finishes searching or viewing location details. | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | The customer can choose to search by categories, such as "Top-rated," "Popular," or "Hidden Gems." | |
| **Step#** | **Exception Paths** | |
| 1. | In step 3, if no tourist locations are found based on the search criteria, the app displays a "No Results Found" message and suggests expanding the radius. | |

### Filter search results

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-006 |
| **Purpose** | | The customer applies filters to narrow down the search results for hotels, restaurants, tourist locations, or services to meet specific preferences (e.g., price, rating, distance). |
| **Pre-conditions** | | The customer must be logged into their account. |
|  | | The search results must be available based on initial criteria (e.g., location, dates). |
| **Post-conditions** | | The customer is presented with filtered search results based on the applied filters. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The customer conducts a basic search (e.g., hotels, restaurants, tourist locations) within the app. | |
| 2. | The app presents an initial list of search results. | |
| 3. | The customer selects the "Filters" option to refine the results. | |
| 4. | The app displays a range of filtering options, such as:   * Price range * Customer ratings * Distance from the current location * Amenities or services (e.g., free Wi-Fi, parking) * Categories (e.g., luxury, budget-friendly) | |
| 5. | The customer selects the desired filters and applies them. | |
| 6. | The app refreshes the search results based on the applied filters and displays the updated list. | |
| 7. | The customer views the refined search results or selects a specific result to see more details. | |
| 8. | The use case ends. | |
|  |  | |
|  |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 3, the customer can choose not to apply any filters and continue browsing the original search results. | |
|  | The customer can modify or remove filters at any time and refresh the search results accordingly. | |
| **Step#** | **Exception Paths** | |
| 1. | In step 6, if no search results match the applied filters, the app displays a "No Results Found" message and suggests relaxing the filters or changing the search criteria. | |

### Live weather updates

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-007 |
| **Purpose** | | The customer receives real-time weather updates for a specific location or along a travel route to plan their journey effectively. |
| **Pre-conditions** | | The customer must be logged into their account. |
|  | | The app should have access to real-time weather data sources. |
|  | | The customer should have granted location access to the app (if required for weather updates along the travel route). |
| **Post-conditions** | | The customer receives updated weather information for the selected location or travel route. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The customer navigates to the "Weather" section in the app or selects a location for which they want to view the weather. | |
| 2. | The customer can either enter a specific location (e.g., city, tourist site) or allow the app to use their current location for weather updates. | |
| 3. | The app retrieves real-time weather data from external APIs and displays the current conditions, such as:   * Temperature * Humidity * Wind speed * Weather forecast (e.g., rain, snow, sunny) * Alerts for extreme weather conditions | |
| 4. | The customer can refresh the weather updates or enable auto-refresh to get updates every few minutes. | |
| 5. | The customer views the weather information and makes decisions accordingly (e.g., continue traveling, delay journey). | |
| 6. | The use case ends when the customer exits the weather section or navigates to another feature of the app. | |
| 7. | The use case ends when the customer finishes viewing the details. | |
|  |  | |
|  |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 2, the customer can search for weather updates along a travel route (e.g., the weather at multiple points between two cities) instead of just a single location. | |
| **Step#** | **Exception Paths** | |
| 1. | In step 3, if there is no real-time data available for the selected location, the app displays a "Weather Information Unavailable" message and suggests trying again later. [which step#?] | |

### Leave Hotel Review

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-008 |
| **Purpose** | | The customer provides feedback or a rating on a hotel based on their experience. |
| **Pre-conditions** | | The customer must be logged into their account. |
|  | | The customer must have used or visited the service/destination they are reviewing. |
| **Post-conditions** | | The review is saved and visible to other users. |
|  | | The hotel's overall rating is updated based on the new review. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The customer navigates to the service or destination (e.g., hotel site) they want to review. | |
| 2. | The customer selects the "Leave a Review" option. | |
| 3. | The app prompts the customer to provide:   * A rating (e.g., 1 to 5 stars) * A written review describing their experience * Optional: Upload images related to their visit (e.g., photos of the hotel or restaurant) | |
| 4. | The customer submits the review. | |
| 5. | The app processes and saves the review, updating the overall rating of the hotel. | |
| 6. | The review becomes visible to other users in the review section. | |
| 7. | The customer can edit or delete their review later if desired. | |
| 8. | The use case ends when the review is successfully submitted or saved. | |
|  |  | |
|  |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 4, the customer can choose to submit only a rating without writing a detailed review. | |
| **Step#** | **Exception Paths** | |
| 1. | In step 5, if there is an issue with the review submission (e.g., network issue), the app displays an error message and asks the customer to retry. | |

### Share trip Details

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-009 |
| **Purpose** | | The customer shares their planned or completed trip details (e.g., destinations, accommodations, itinerary) with friends or family through social media, email, or messaging platforms. |
| **Pre-conditions** | | The customer must be logged into their account. |
|  | | The customer must have created or completed a trip using the app's planning feature. |
| **Post-conditions** | | The trip details are successfully shared through the selected medium (e.g., email, social media). |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The customer navigates to their planned or completed trip within the app. | |
| 2. | The customer selects the "Share" option for the trip details. | |
| 3. | The app prompts the customer to choose a sharing method, such as:   * Email | |
| 4. | The customer selects their preferred sharing method and enters the recipient(s) or selects from their contacts. | |
| 5. | The app generates a shareable link or attaches the trip details (e.g., PDF, itinerary summary) depending on the chosen platform. | |
| 6. | The customer adds any optional message or note to accompany the trip details. | |
| 7. | The customer confirms and sends the trip details. | |
| 8. | The app shows a confirmation message that the trip details have been shared successfully. | |
| 9**.** | The use case ends when the trip details are sent. | |
|  |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 3, the customer can choose to share specific portions of the trip (e.g., hotel reservations or restaurant bookings) instead of the entire trip itinerary. | |
| **Step#** | **Exception Paths** | |
| 1. | In step 5, if the chosen sharing platform does not support the format (e.g., unsupported file type or link issue), the app displays an error message and suggests an alternative sharing method. | |

### Ai based trip planning

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-010 |
| **Purpose** | | The customer uses the app’s AI feature to generate a personalized trip plan based on their preferences, budget, and interests (e.g., sightseeing, adventure, relaxation). |
| **Pre-conditions** | | The customer must be logged into their account. |
|  | | The customer must provide trip-related preferences, such as destination, dates, budget, and interests. |
|  | | The app must have access to AI models and external data (e.g., hotel availability, weather conditions). |
| **Post-conditions** | | A detailed and customized trip itinerary is generated and presented to the customer. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The customer navigates to the "Trip Planning" section of the app. | |
| 2. | The customer selects the "AI Trip Planner" option. | |
| 3. | The app prompts the customer to provide trip details, such as:   * Desired destination(s) * Dates of travel * Budget (e.g., per day or total) * Preferences (e.g., type of accommodation, activities, food preferences) * Mode of transportation (e.g., flight, rental car). | |
| 4. | The customer submits the trip preferences to the AI engine. | |
| 5. | The app processes the customer’s inputs using machine learning algorithms and external data (e.g., hotel rates, weather forecasts). | |
| 6. | The AI-based system generates a personalized trip plan, including:   * Suggested accommodations * Recommended tourist spots and activities * Estimated costs * Suggested dining options * Optimized travel routes and modes of transport | |
| 7. | The app displays the trip plan for the customer to review. | |
| 8. | The customer can:   * Save the trip plan for future reference * Modify or edit the plan (e.g., change hotel, add or remove activities) * Share the trip plan with others (via email or social media) | |
| 9**.** | The use case ends when the customer finalizes or saves the trip plan. | |
|  |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | n step 6, the customer can choose to refine the trip plan by adjusting preferences, such as increasing the budget or focusing on different types of activities (e.g., more sightseeing or adventure). | |
| **Step#** | **Exception Paths** | |
| 1. | In step 5, if the AI system cannot generate a trip plan due to missing data (e.g., no available hotels or transportation for the chosen dates), the app displays an error message and suggests alternate options or dates. | |

### Save favorite locations

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-011 |
| **Purpose** | | The customer saves locations (e.g., hotels, tourist attractions, restaurants) to their favorites list for easy access later. |
| **Pre-conditions** | | The customer must be logged into their account. |
|  | | The location to be saved (e.g., a hotel or tourist spot) must exist in the system’s database. |
| **Post-conditions** | | The location is added to the customer's list of saved favorites. |
|  | | The customer can view or remove the saved location from their favorites list at any time. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The customer views details of a location (e.g., a hotel or tourist attraction) in the app. | |
| 2. | The app displays an option to "Add to Favorites" for the location. | |
| 3. | The customer selects the "Add to Favorites" option. | |
| 4. | The app confirms that the location has been added to the customer's favorites list. | |
| 5. | The customer can view their saved locations by navigating to the "Favorites" section of the app. | |
| 6. | The use case ends when the location is successfully added to the favorites list. | |
|  |  | |
|  |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | The customer can choose to remove a location from their favorites list by navigating to the "Favorites" section and selecting "Remove" next to the saved location. | |
| **Step#** | **Exception Paths** | |
| 1. | In step 3, if there is a network issue or failure, the app displays an error message indicating that the location could not be saved and allows the customer to try again. | |

### View rent a car

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-012 |
| **Purpose** | | The customer views details of available rental car services, including prices, car types, and availability. |
| **Pre-conditions** | | The customer must be logged into their account. |
|  | | The rental car service details must exist in the system’s database. |
| **Post-conditions** | | The customer is able to view rental car options, prices, and availability. |
|  | | The customer can proceed to book a rental car or save the rental car service to their favorites. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The customer navigates to the "Rent-A-Car" section of the app. | |
| 2. | The app prompts the customer to enter search criteria (e.g., location, rental dates, car type). | |
| 3. | The customer submits the search criteria. | |
| 4. | The app displays a list of available rental car services matching the search criteria, including: Car types (e.g., SUV, sedan, compact), Rental prices (per day or trip), Rental service details (e.g., contact information, pickup locations). | |
| 5. | The customer selects a specific rental car service to view more details. | |
| 6. | The app displays detailed information for the selected rental car service. | |
| 7. | The customer can get contact details and book the car if it wants. | |
| 8. | The use case ends when the customer has viewed the details or saved the service to their favorites. | |
|  |  | |
|  |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | The customer can choose to save a rental car service to their favorites for future booking. | |
| **Step#** | **Exception Paths** | |
| 1. | In step 4, if no rental cars are available for the selected criteria, the app displays a message informing the customer and suggests alternative dates or locations. | |

### View Tourist location Reviews

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-013 |
| **Purpose** | | The customer views reviews and ratings of specific tourist locations, allowing them to make informed decisions about visiting the location. |
| **Pre-conditions** | | The customer must be logged into their account. |
|  | | The tourist location must exist in the system’s database and have reviews available. |
| **Post-conditions** | | The customer is able to view a list of reviews and ratings for the selected tourist location. |
|  | | The customer can save the location to their favorites or write their own review if they've visited the location. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The customer navigates to the "Tourist Locations" section of the app. | |
| 2. | The app prompts the customer to search for or select a tourist location. | |
| 3. | The customer selects a tourist location from the search results. | |
| 4. | The app displays the tourist location’s overview, along with a "View Reviews" option. | |
| 5. | The customer selects the "View Reviews" option. | |
| 6. | The app displays a list of reviews for the selected tourist location. | |
| 7. | The customer reads the reviews and can return to the location details or save the tourist location to their favorites. | |
| 8. | The use case ends when the customer has finished viewing reviews. | |
|  |  | |
|  |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | The customer can write their own review if they have visited the location and want to provide feedback. | |
| **Step#** | **Exception Paths** | |
| 1. | In step 4, if no reviews are available for the tourist location, the app displays a message indicating there are no reviews and suggests the customer be the first to write one. | |

### View Restaurants Details

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-014 |
| **Purpose** | | The customer views detailed information about a restaurant, including its menu, location, ratings, reviews, and services offered. |
| **Pre-conditions** | | The customer must be logged into their account. |
|  | | The restaurant must exist in the system’s database. |
| **Post-conditions** | | The customer can view the restaurant's details and save it to their favorites. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The customer navigates to the "Restaurants" section of the app. | |
| 2. | The app prompts the customer to enter search criteria (e.g., cuisine type, location). | |
| 3. | The customer submits the search criteria. | |
| 4. | The app displays a list of restaurants matching the search criteria. | |
| 5. | The customer selects a restaurant from the search results. | |
| 6. | The app displays detailed information for the selected restaurant, including: Restaurant name and address, Phone number and business hours, Menu options (if available), Average cost per meal, Ratings and reviews from other customers, Photos of the restaurant and menu items. | |
| 7. | The customer can choose to save the restaurant to their favorites for future reference or share the restaurant details with others. | |
| 8. | The use case ends when the customer has viewed the restaurant details or saved the restaurant. | |
|  |  | |
|  |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | The customer can modify the search criteria to explore different restaurant options. [Is this an input or alternate path?] | |
| **Step#** | **Exception Paths** | |
| 1. | In step 4, if no restaurants match the search criteria, the app displays a message suggesting alternative options. | |

### View Restaurant Reviews

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-015 |
| **Purpose** | | The customer views reviews and ratings of a specific restaurant to help them decide whether to visit or order from it. |
| **Pre-conditions** | | The customer must be logged into their account. |
|  | | The restaurant must exist in the system’s database and have reviews available. |
| **Post-conditions** | | The customer is able to view a list of reviews and ratings for the selected restaurant. |
|  | | The customer can save the restaurant to their favorites or write their own review if they have visited the restaurant. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The customer navigates to the "Restaurants" section of the app. | |
| 2. | The app prompts the customer to search for or select a restaurant. | |
| 3. | The customer selects a restaurant from the search results. | |
| 4. | The app displays the restaurant’s overview, along with a "View Reviews" option. | |
| 5. | The customer selects the "View Reviews" option. | |
| 6. | The app displays a list of reviews for the selected restaurant. | |
| 7. | The customer reads the reviews and can return to the restaurant details or save the restaurant to their favorites. | |
| 8. | The use case ends when the customer has finished viewing reviews. | |
|  |  | |
|  |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | The customer can choose to write their own review of the restaurant if they have visited or ordered from it. | |
| **Step#** | **Exception Paths** | |
| 1. | In step 4, if no reviews are available for the restaurant, the app displays a message indicating there are no reviews and suggests the customer be the first to write one. | |

### Create Hotel Account

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-016 |
| **Purpose** | | The hotel management creates an account for their hotel to manage bookings, update room availability, and interact with customers. |
| **Pre-conditions** | | The hotel management team must have access to the app. |
|  | | The hotel must not already have an account in the system. |
| **Post-conditions** | | The hotel account is created, and the hotel management can log in to manage their hotel information and interact with customers. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The hotel management team navigates to the "Hotel Management" section of the app. | |
| 2. | The app displays an option to "Create Hotel Account." | |
| 3. | The hotel management selects "Create Hotel Account." | |
| 4. | The app prompts the hotel management to enter their hotel’s information. | |
| 5. | The app asks the hotel management to create login credentials (email and password). | |
| 6. | The hotel account is now created, and the management team is redirected to the dashboard | |
| 7. | The use case ends when the hotel account is successfully created and verified. | |
|  |  | |
|  |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | The hotel management can choose to modify the hotel details at any time before submitting the account creation. | |
| **Step#** | **Exception Paths** | |
| 1. | In step 4, if required hotel details are missing or invalid (e.g., incorrect email format), the app displays an error and asks the hotel management to correct the information before proceeding. | |

### Update Hotel records

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-017 |
| **Purpose** | | The hotel management updates their hotel’s information such as room availability, pricing, services offered, and other details to ensure accurate and up-to-date listings for customers. |
| **Pre-conditions** | | The hotel management must be logged into their hotel account. |
|  | | The hotel account must already exist in the system. |
| **Post-conditions** | | The hotel’s records are updated, and customers see the latest information when searching or booking the hotel. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The hotel management logs into the app with their credentials. | |
| 2. | The app redirects them to the "Hotel Management Dashboard." | |
| 3. | The hotel management selects the "Update Hotel Records" option. | |
| 4. | The app displays the current hotel records. | |
| 5. | The hotel management edits the desired information (e.g., updating room availability, changing pricing). | |
| 6. | The hotel management submits the updated information. | |
| 7. | The app processes the updates and confirms the changes. | |
| 8. | The updated records are now visible to customers when they search for hotels or make bookings. | |
| 9. | The use case ends when the hotel’s records are successfully updated. | |
|  |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 5, the hotel management can choose to upload new photos, change policies, or remove services offered by the hotel. | |
| **Step#** | **Exception Paths** | |
| 1. | In step 6, if any required information (e.g., room availability, pricing) is missing or invalid, the app displays an error and prompts the hotel management to correct it before submitting. | |

### Upload hotel photos

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-018 |
| **Purpose** | | The hotel management uploads images of their hotel, such as rooms, amenities, and services, to give customers a visual representation of the hotel when searching or booking. |
| **Pre-conditions** | | The hotel management must be logged into their hotel account. |
|  | | The hotel account must already exist in the system. |
| **Post-conditions** | | The photos are uploaded successfully and are displayed to customers on the hotel’s profile. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The hotel management logs into their account using valid credentials. | |
| 2. | The app redirects them to the "Hotel Management Dashboard." | |
| 3. | The hotel management selects the "Upload Photos" option. | |
| 4. | The app prompts them to choose images from their device (e.g. mobile phone) or drag and drop files. | |
| 5. | The hotel management selects the images they wish to upload (e.g., room interiors, dining areas, hotel lobby). | |
| 6. | The app displays a preview of the selected images. | |
| 7. | The hotel management confirms the selection and clicks the "Upload" button. | |
| 8. | The app processes the upload and displays a success message once the photos are added. | |
|  | The uploaded photos are now visible to customers when viewing the hotel’s profile. | |
|  | The use case ends when all photos are successfully uploaded. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | The hotel management can choose to remove or reselect images before uploading them in step 5. | |
| **Step#** | **Exception Paths** | |
| 1. | In step 5, if the selected file formats are unsupported (e.g., non-image files), the app displays an error and prompts the hotel management to upload valid image formats (e.g., JPEG, PNG). | |

### View customers reviews

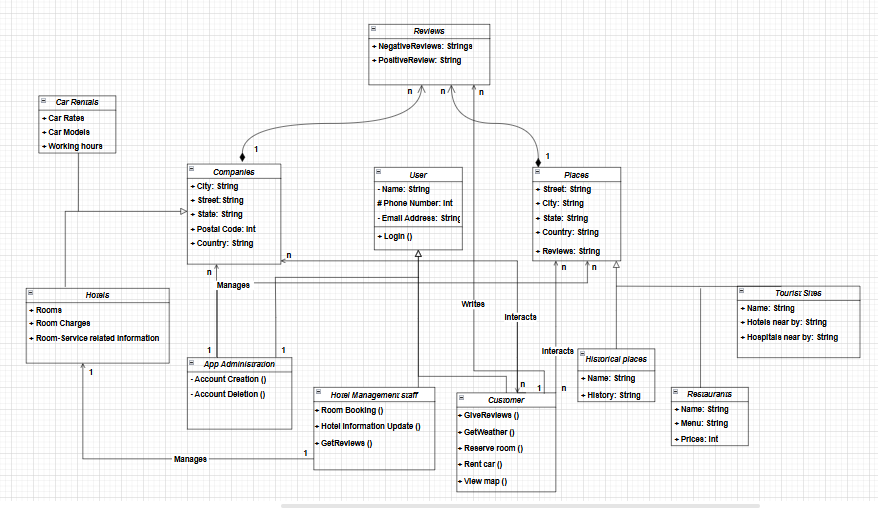
|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-019 |
| **Purpose** | | The hotel management or customer views reviews left by previous guests, helping them assess the hotel’s services and customer experiences. |
| **Pre-conditions** | | The hotel management must be logged into their hotel account. |
|  | | The hotel account must already exist in the system. |
| **Post-conditions** | | The reviews are displayed successfully to the hotel management or customer. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The hotel management logs into their account using valid credentials. | |
| 2. | The app redirects them to the "Hotel Management Dashboard." | |
| 3. | The user selects the "View Reviews" option for the hotel they are interested in. | |
| 4. | The app displays a list of customer reviews. | |
| 5. | The Hotel management can expand or collapse individual reviews to read the full text or see summarized ratings. | |
| 6. | For hotel management, the app may also display options to respond to customer reviews (e.g., address concerns or thank customers). | |
| 7. | The use case ends when the reviews are successfully viewed. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 5, the hotel management or customer may apply filters to view only positive or negative reviews, or reviews from specific date ranges. | |
| **Step#** | **Exception Paths** | |
| 1. | If there are no customer reviews for the hotel, the app displays a message stating, "No reviews available." | |

### Connect with Customers

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-020 |
| **Purpose** | | The hotel management interacts with customers by responding to inquiries, addressing concerns, or providing information about hotel services. |
| **Pre-conditions** | | The hotel management must be logged into their hotel account. |
|  | | The customer must have submitted an inquiry or request for connection through the app. |
| **Post-conditions** | | The hotel management successfully communicates with the customer, resolving inquiries or concerns. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 1. | The hotel management logs into their account using valid credentials. | |
| 2. | The app redirects them to the "Hotel Management Dashboard." | |
| 3. | The hotel management selects the "Customer Inquiries" or "Messages" option. | |
| 4. | The app displays a list of customer inquiries. | |
| 5. | The hotel management selects a specific inquiry to view details. | |
| 6. | The app displays the full message from the customer, including any attached files or images (if applicable). | |
| 7. | The hotel management drafts a response to the customer inquiry. | |
| 8. | The hotel management sends the response by clicking the "Send" button. | |
| 9. | The use case ends when the response is successfully sent. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | The hotel management can also choose to forward certain inquiries to other departments (e.g., customer support) if necessary. | |
| **Step#** | **Exception Paths** | |
| 1. | If there are no new inquiries, the app displays a message stating, "No new inquiries available." | |

# Class Diagram

## Diagram



## Description

There are three major classes:

1. Users
2. Places
3. Companies

Each Major class will have their own child classes which will then have their specific roles.

There is another class called reviews, with which the customer would interact and write reviews of the companies (hotels, car rentals) and places.

1. The class ***USER*** includes all those who will use the application, which includes the customer, the hotel management staff (responsible for managing their hotel updates) and the developers.

* The ***CUSTOMER*** is the main audience of the mobile application and will use the application for its main purpose. To make the communication smooth and facilitate the customer, the relation between customer and companies are bidirectional (n to n), i.e. the customer will be able to interact with the companies and vice versa.
* The customer will also be able to interact with the places class, but this relation is unidirectional (n to n) and only the customer will be able to view the details of the places (historical places, restaurants etc.)
* The customer will also be able to write reviews about the hotels and places.
* The class ***HOTEL MANAGEMENT STAFF*** is also a child class of the user class. This is the management team of hotel which will be responsible for managing their hotel updates on the application such as posting the latest pictures, rooms availability information, and other services information.
* It has a unidirectional 1 to 1 relation with the class ***HOTELS.***
* The class ***APP ADMINISTRSTION/DEVELOPERS*** is the child class of user responsible for managing the accounts and database of the application. They are also responsible for managing the bugs in the application and ensure true and accurate information.
* It has a unidirectional relation of 1 to n with the places and companies’ classes.

1. The class ***COMPANIES*** is another major class which includes the hotels and car rentals as its child classes.

* It has two child classes, i.e. ***hotels*** and ***car rentals.***
* It has a unidirectional relation with the customer class, such that the customer will be able to book rooms at the hotels and rent cars from the car rental companies (child classes of the companies).
* It has a whole-part composition 1 to n relation with the review class, such that the class will hold reviews about the hotels and car companies.
* The ***HOTELS*** child class of ***COMPANIES*** class will have a 1 to 1 relation with the ***HOTEL MANAGEMENT STAFF*** child class of ***USER*** class.

1. The class ***PLACES*** is the parent class of the places the customer wants to visit.

* It has three child classes, i.e. ***Historical places, restaurants, and tourist sites.***
* This class has a unidirectional relation with the user’s child classes, i.e. customer and developers.
* The customer is capable of searching for places and viewing the location and reviews of tourist sites, restaurant and historical places.
* The developers are capable of updating the class.
* This class also has a whole-part composition 1 to n relation with the review class, such that the class will hold reviews about the places.

1. The ***REVIEWS*** class, as the name suggests will be keeping the reviews put on by the customers about the places, hotels and car rentals.

* It has a whole-part composition n to 1 relation with the companies and places, i.e. each place, hotel, or car rentals can have multiple reviews.
* It also has a n to 1 relation with the customers, i.e. one customer can write multiple reviews about the places and companies.

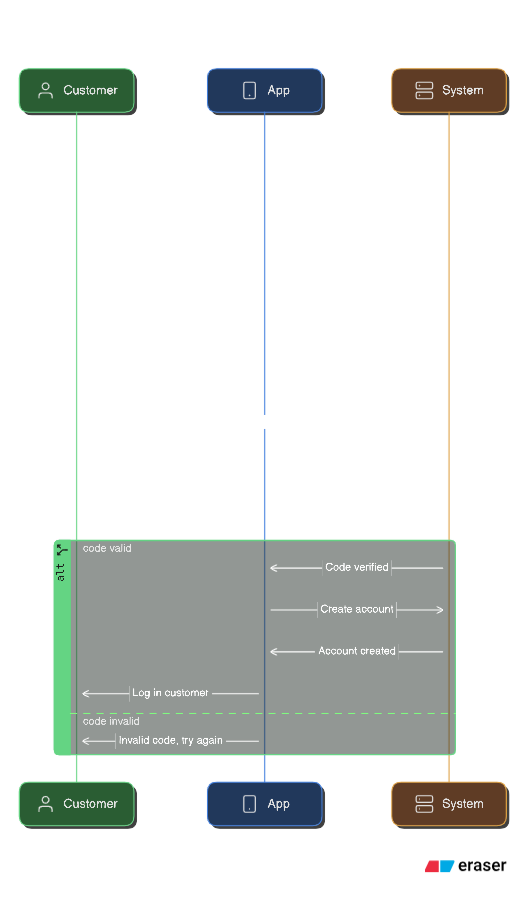
# Sequence Diagrams

<Create a sequence diagram for each of the 20 use cases selected above.>

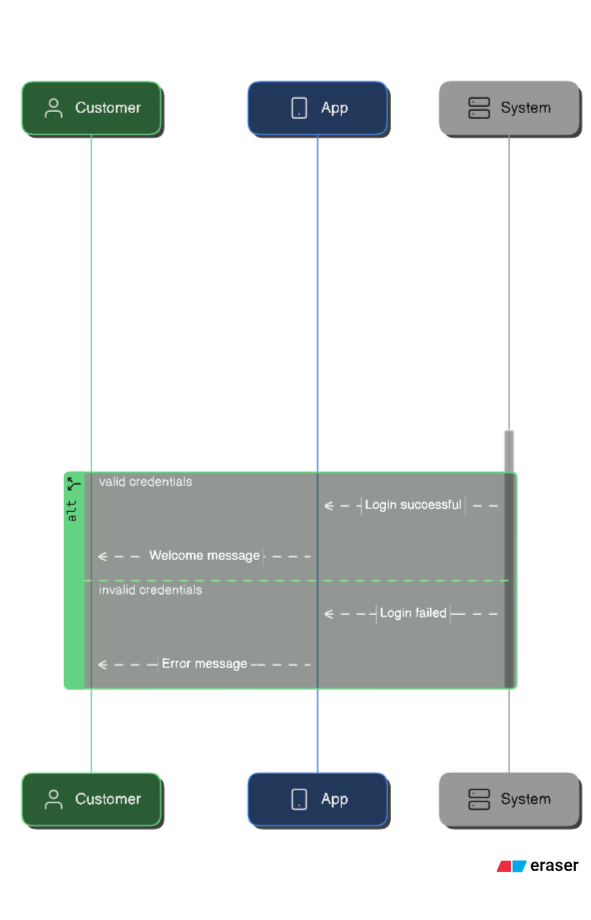
The sequence diagrams must have classes from the Class diagram you created earlier in this document. These diagrams must show the interactions among the system classes. In all of the sequence diagrams that you have created, there is hardly any classes from your class model.

[Some parts of sequence diagram are not visible due to color.]

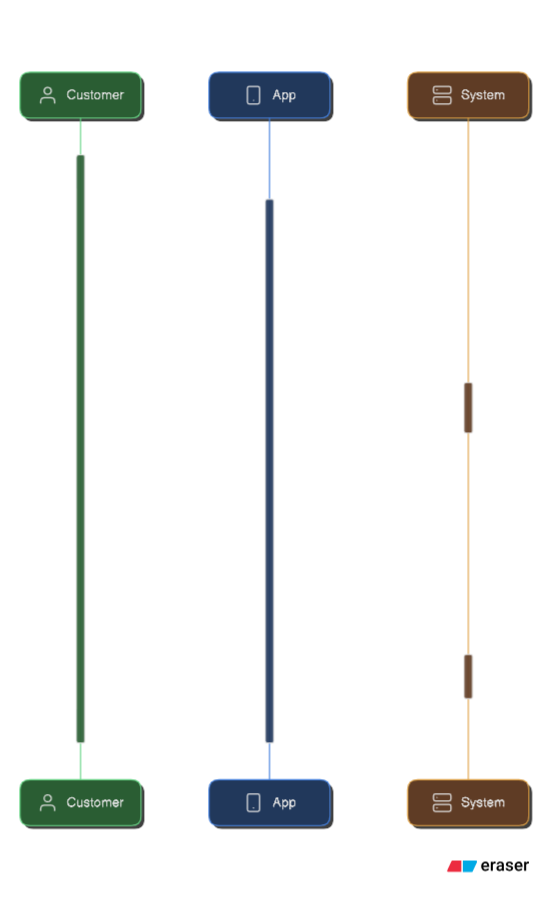
## Create Account



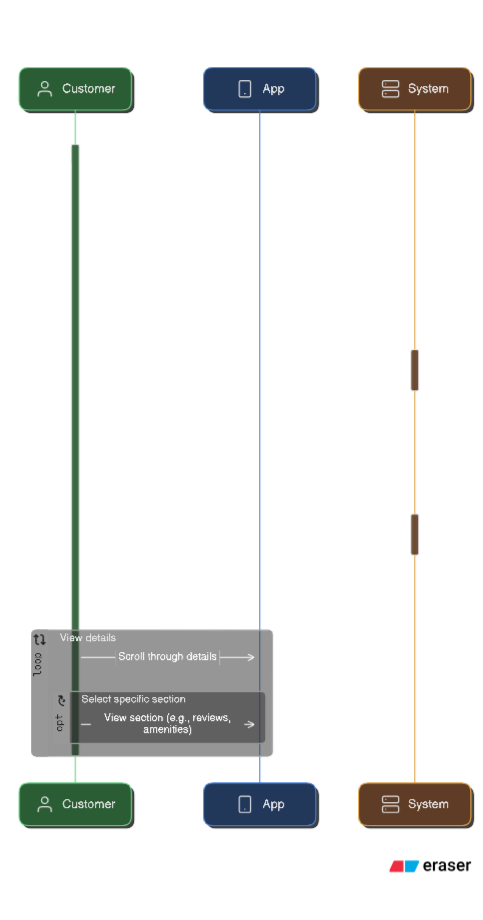
## Logging in



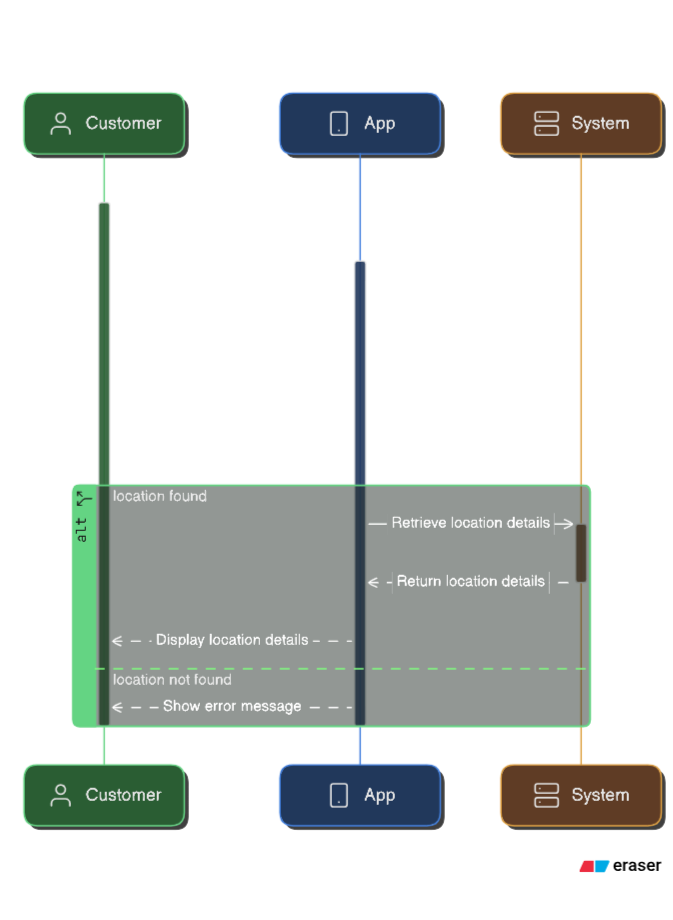
## 5.3 Book Hotel



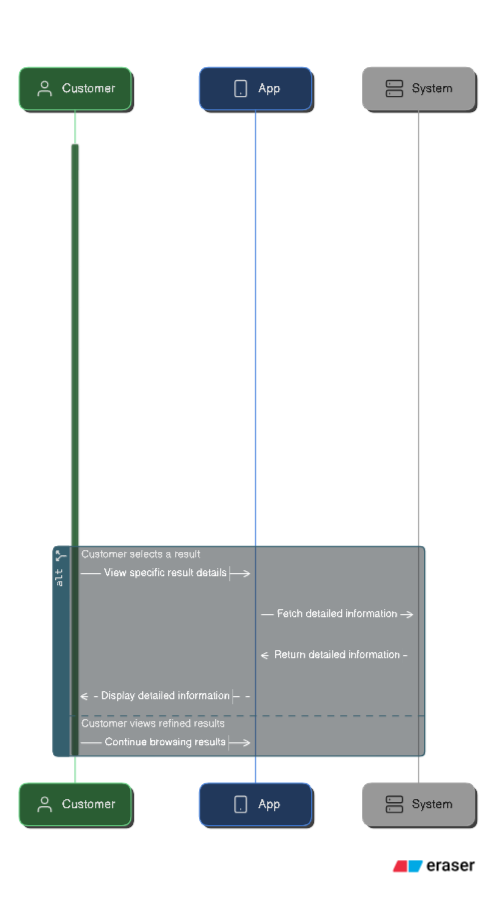
## 5.4 Hotel Details



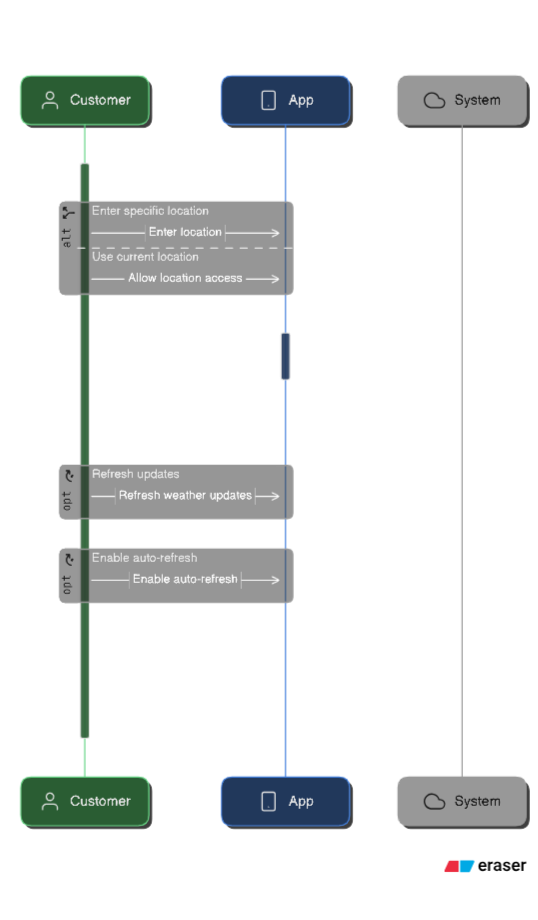
## 5.5 Search Tourist Locations



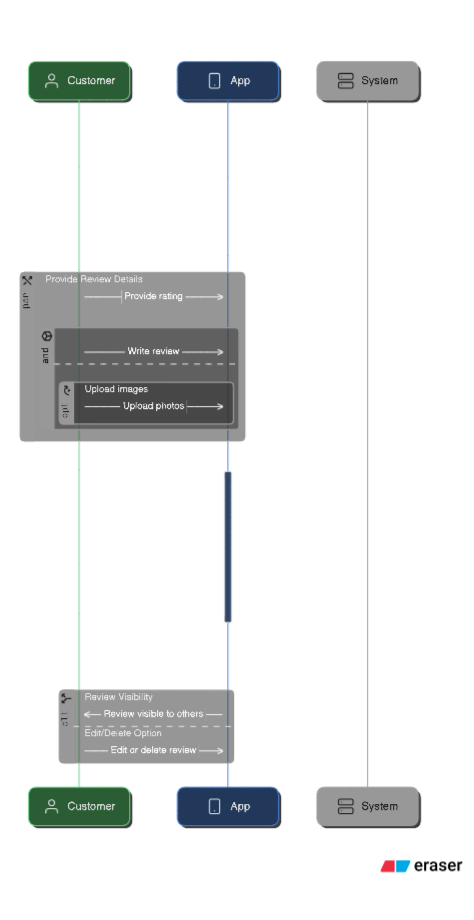
## 5.6 Filter Search Results



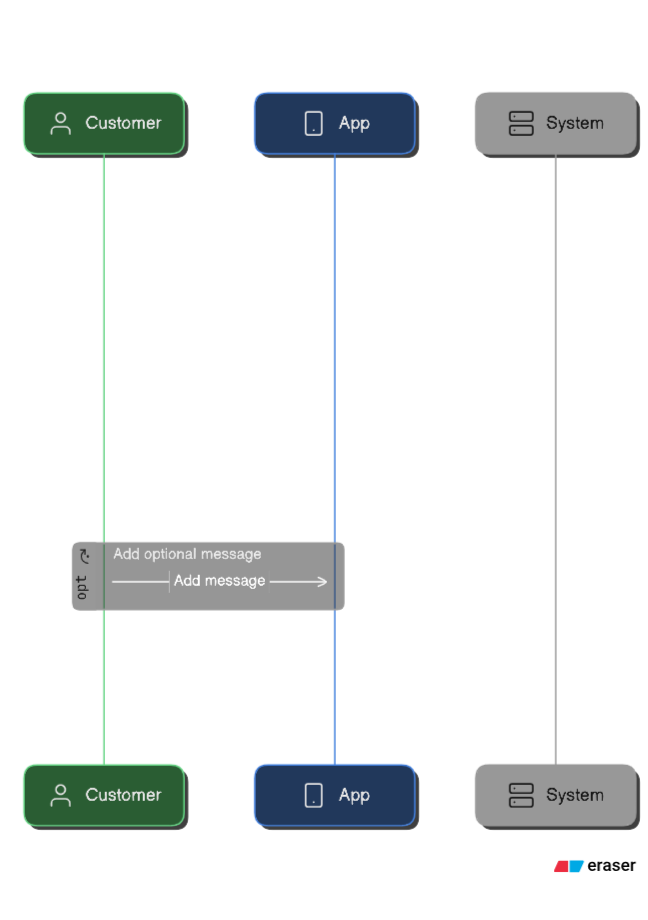
## 5.7 Live weather updates



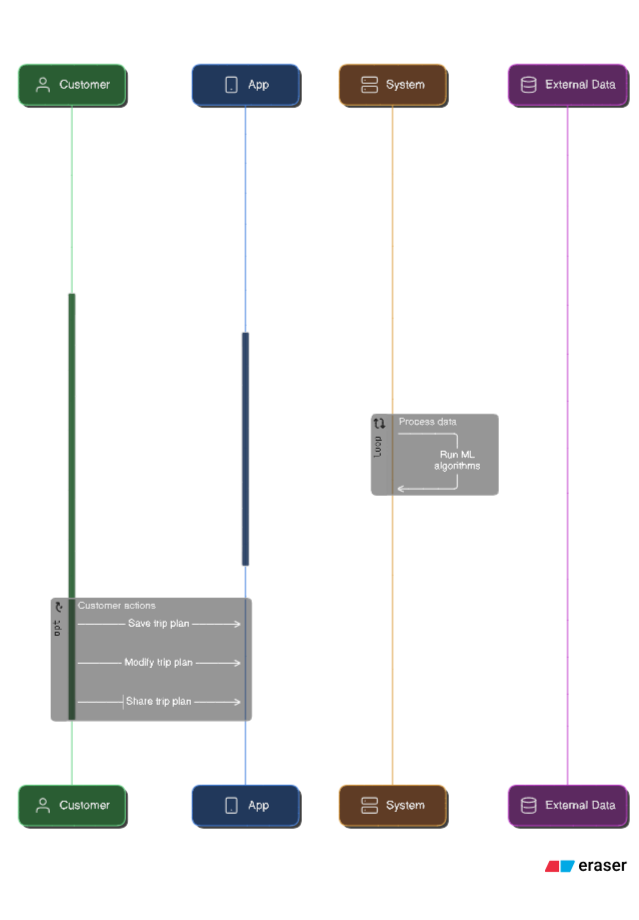
## 5.8 Leave Hotel Review



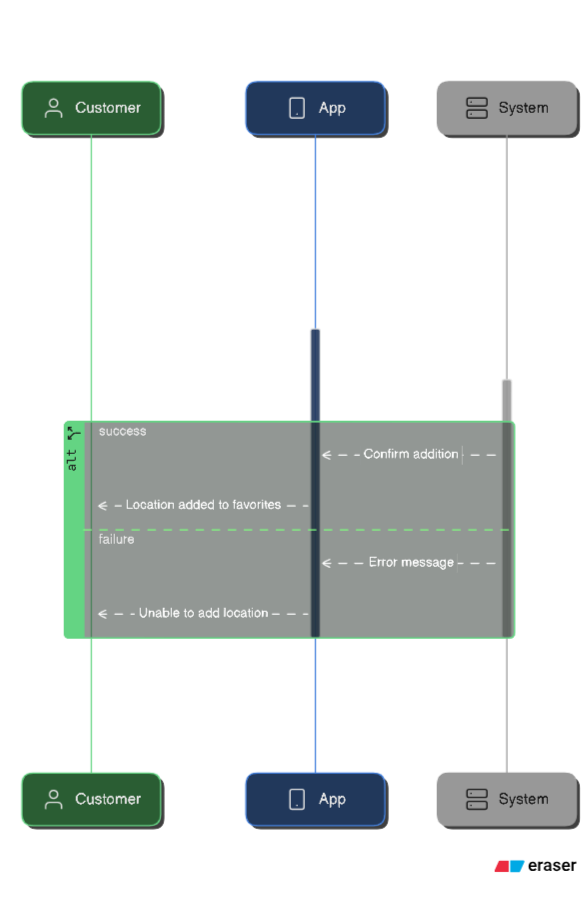
## 5.9 Share trip Details



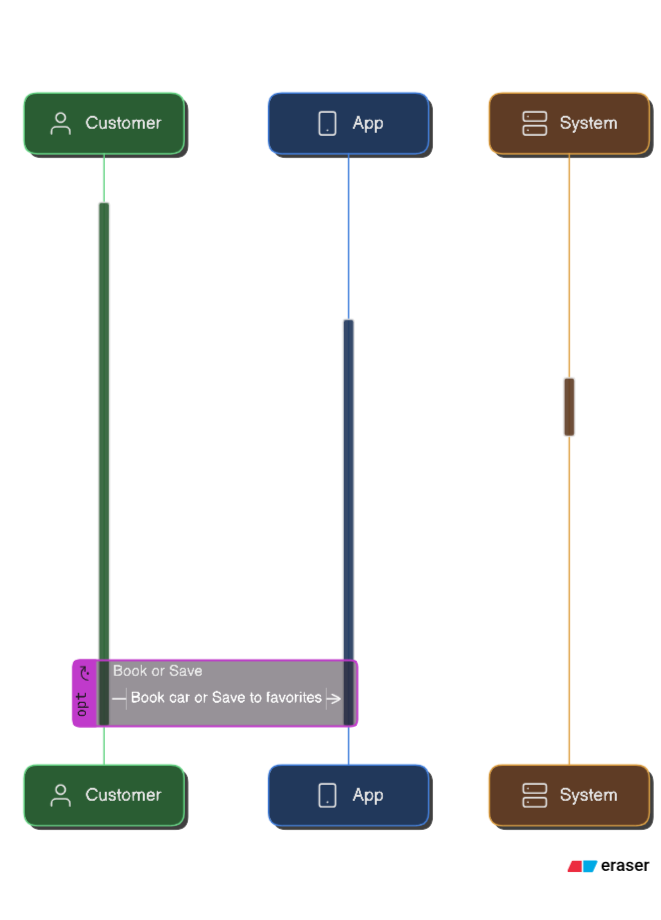
## 5.10 AI based Trip Planning



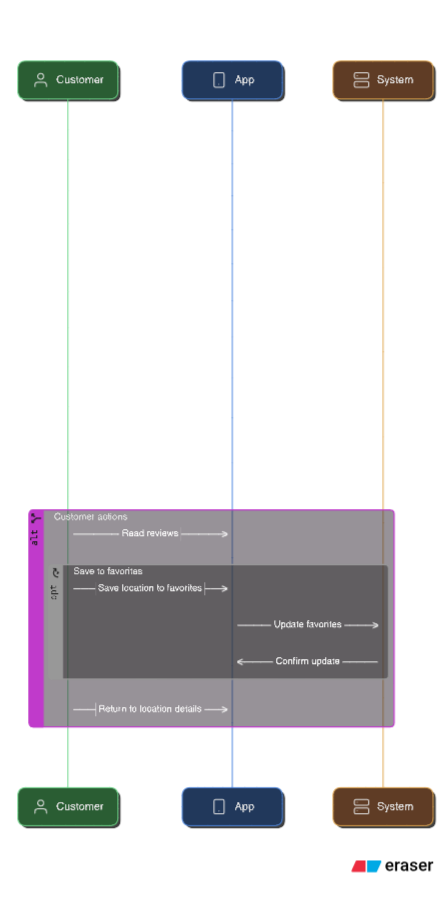
## 5.11 Save favorite locations



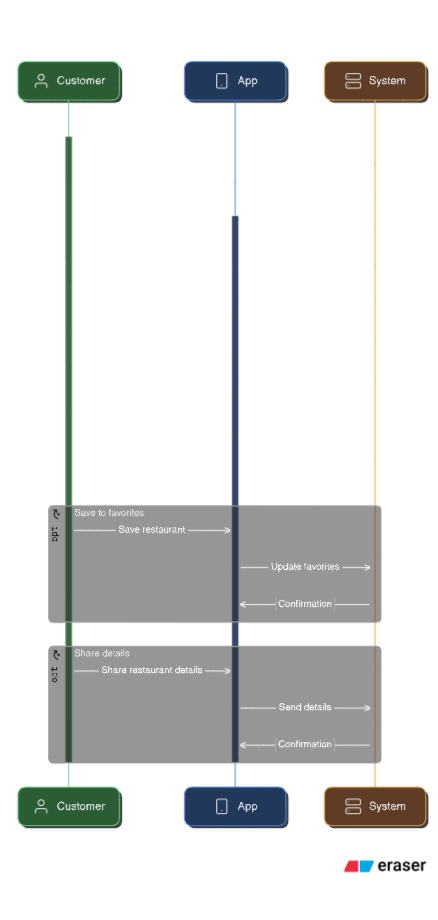
## 5.12 View rent a car



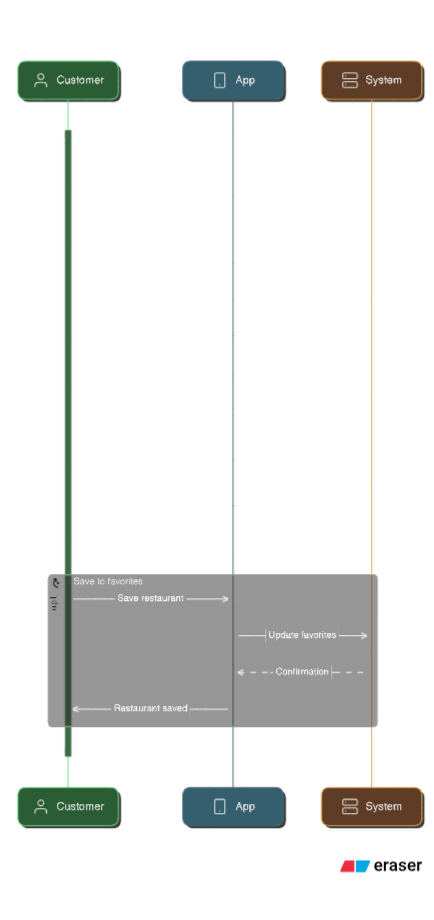
## 5.13 View Tourist location Reviews



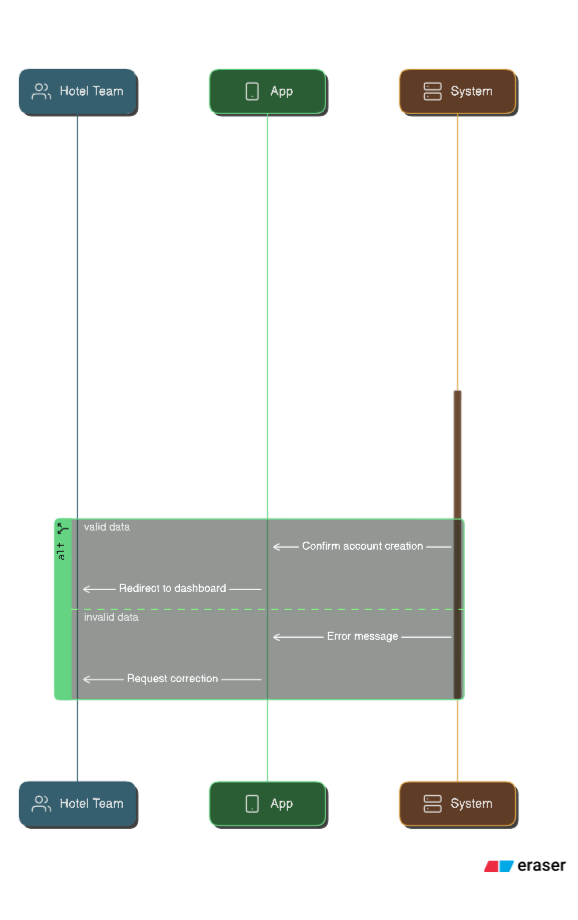
## 5.14 View Restaurant Details



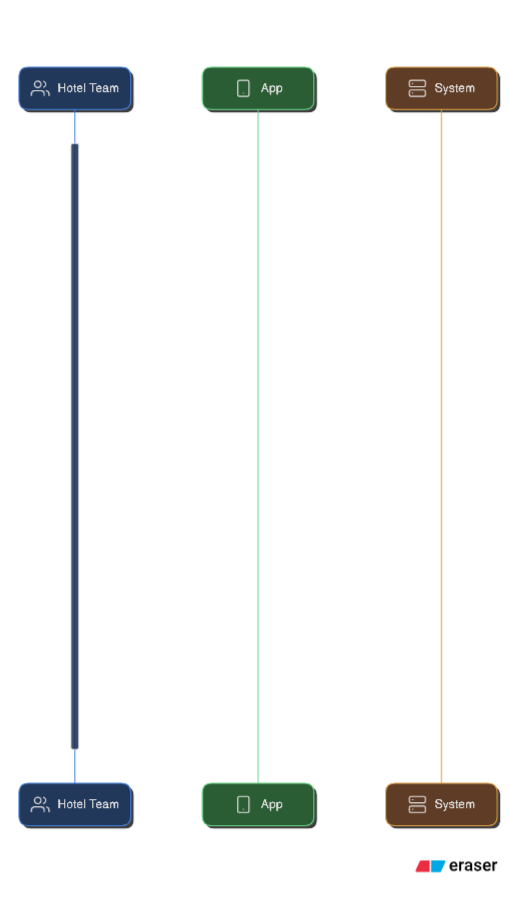
## 5.15 View Restaurant Reviews



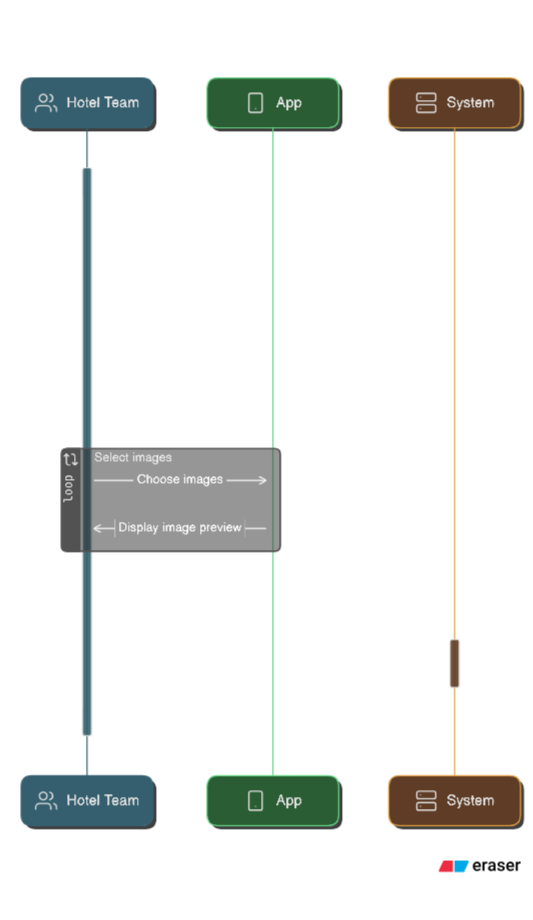
## 5.16 Create Hotel account



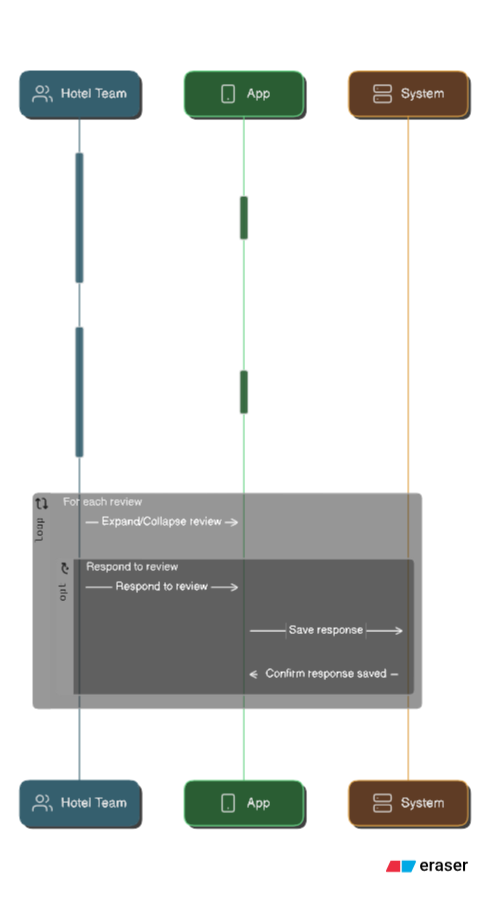
## 5.17 Update Hotel records



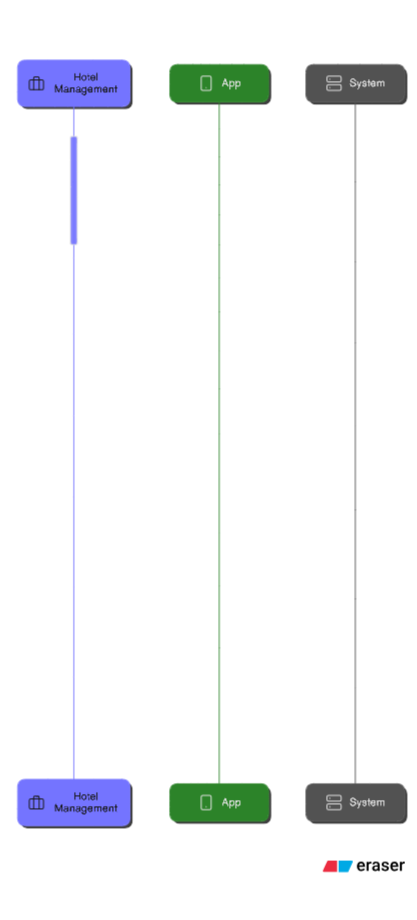
## 5.18 Upload Hotel Photos



## 5.19 View customers reviews



## 5.20 Connect with Customers



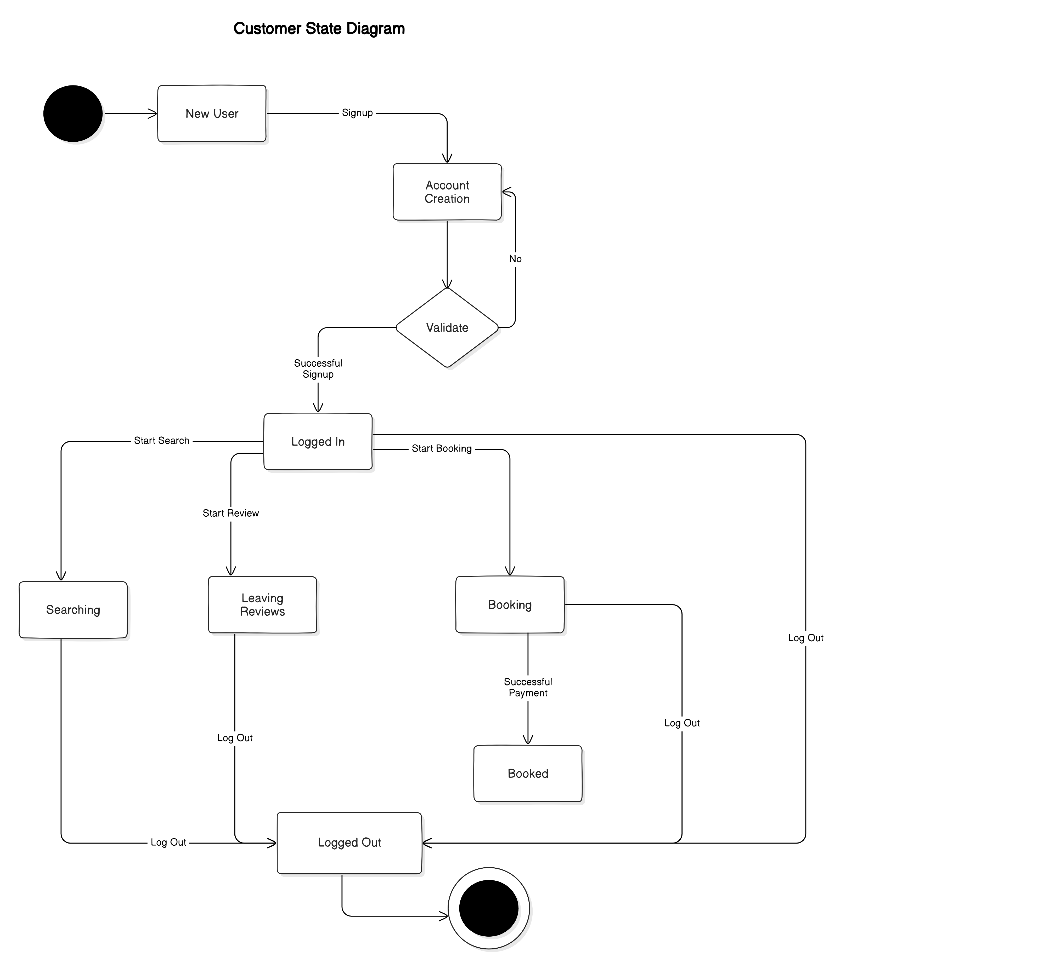
# State Diagrams

## Diagram details

The Customer State Diagram outlines the various stages a customer goes through while interacting with the system. The journey starts in an Idle state, where the customer either creates an account or logs in. Upon successful account creation, the customer transitions to the "Logged In" state, where they can search for tourist locations or services. From there, they proceed to view details about their selection and initiate the booking process, which leads to "Payment Processing." A decision diamond checks the success of the payment; if successful, the customer reaches the "Booking Confirmed" state, otherwise, they return to "Booking in Progress."

In addition to booking services, customers can also submit reviews after using a service. Once a review is written and successfully submitted, they transition to the "Review Confirmation" state. At any point, the customer can log out.

## Diagram



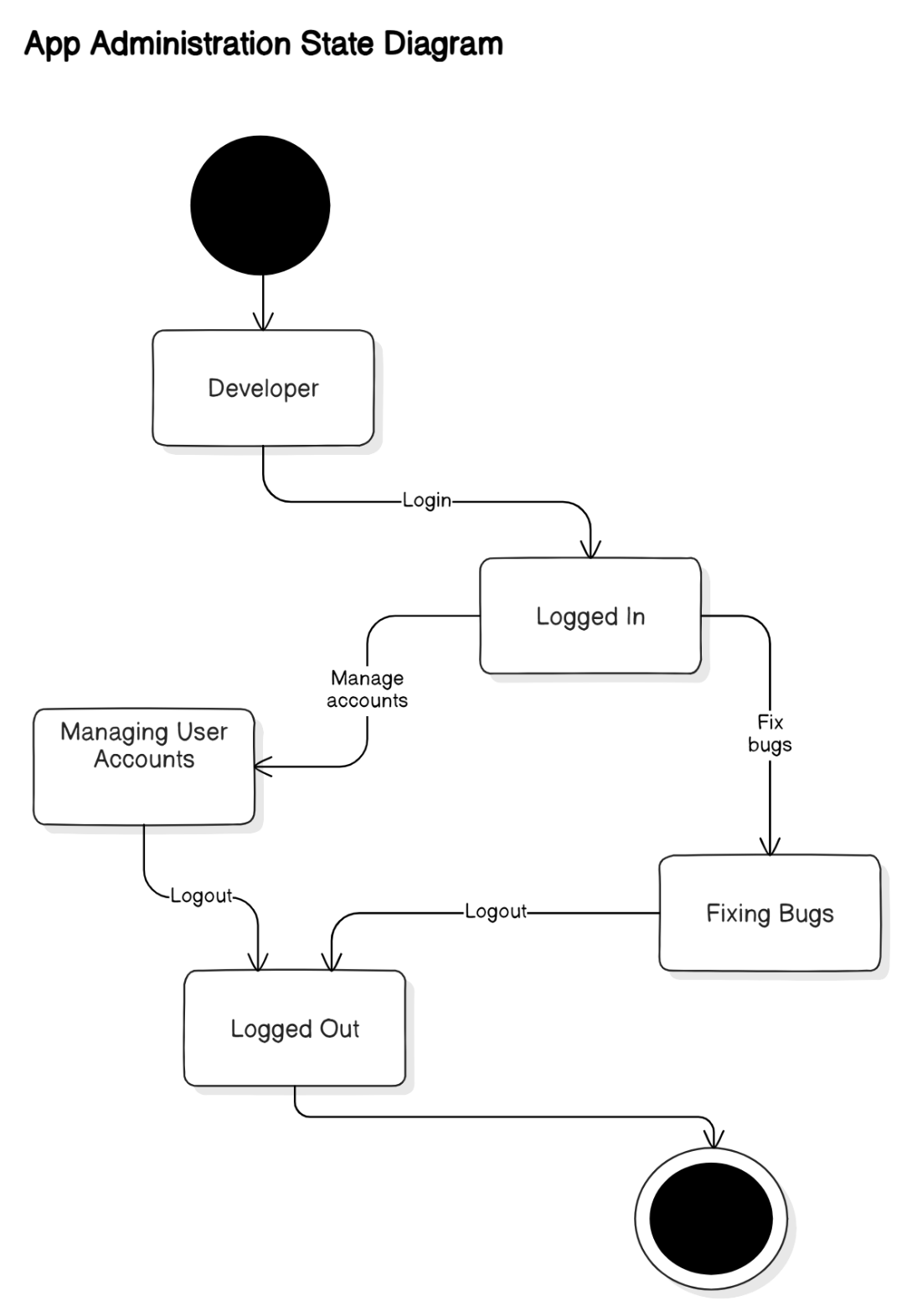
## Diagram details

This state diagram shows the Developer's actions while managing and administering the app

It begins with the developer logging into the system, transitioning to the "Logged In" state. From there, the developer can either manage user accounts or fix bugs, with transitions between these states depending on the specific tasks being handled.

Once the necessary actions are completed, the developer can log out of the system, returning to the "Logged Out" state. The diagram captures the core workflow, from logging in, performing administrative tasks, and ending with logging out.

## Diagram

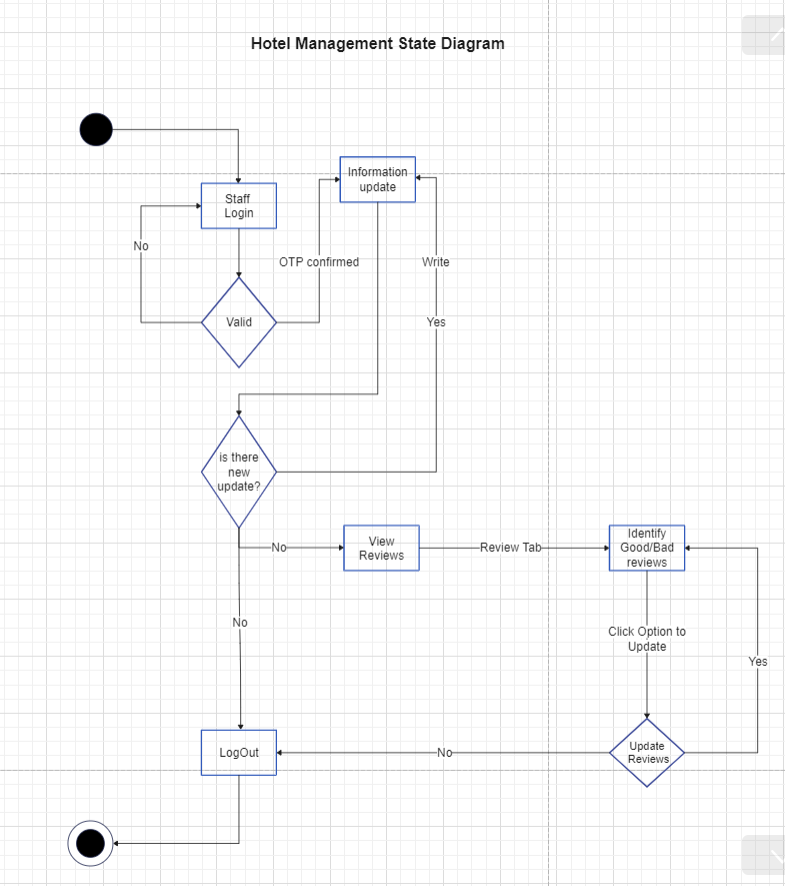


## Diagram details

The Hotel Management State Diagram outlines hotel staff managing hotel operations. Transitioning to the "Logged In" state upon successful OTP submission and login. Once logged in, staff members can either update hotel information, such as room availability, photos, and services, or they can view customer reviews and feedback from where they can update the reviews (in case if someone uses vulgar language).

After completing their tasks, staff members have the option to log out, returning to the "Logged Out" state.

## Diagram

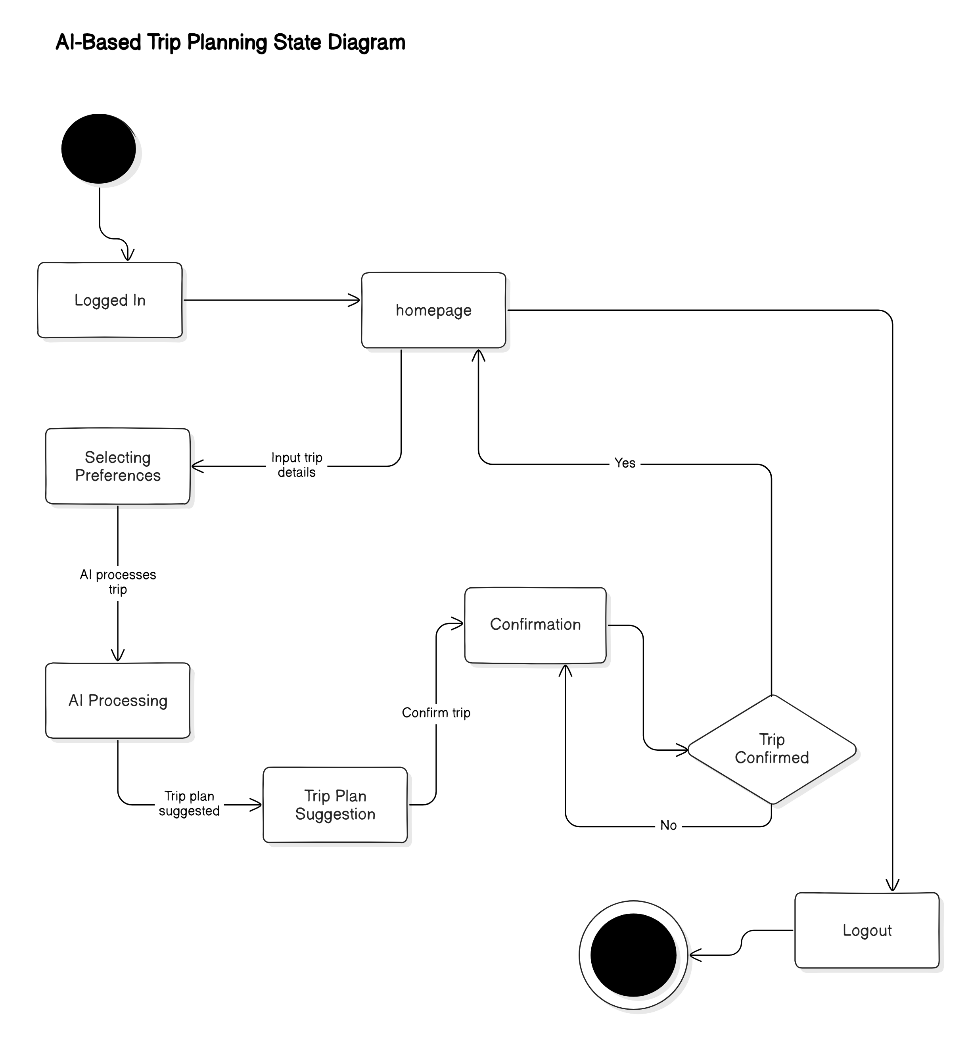


## Diagram details

The AI-Based Trip Planning State Diagram outlines the flow a user follows when planning a trip through the AI-powered system. It begins with the user logging in and navigating to the homepage. From there, the user selects their trip preferences, which are processed by the AI to generate a trip plan suggestion.

The suggested trip is presented for confirmation. If the user confirms the trip, the state moves to "Trip Confirmed." If not, the user is redirected to make changes. Upon final confirmation, the user can choose to log out, marking the end of the session. This diagram captures the user interaction, AI processing, and confirmation flow in trip planning

## Diagram

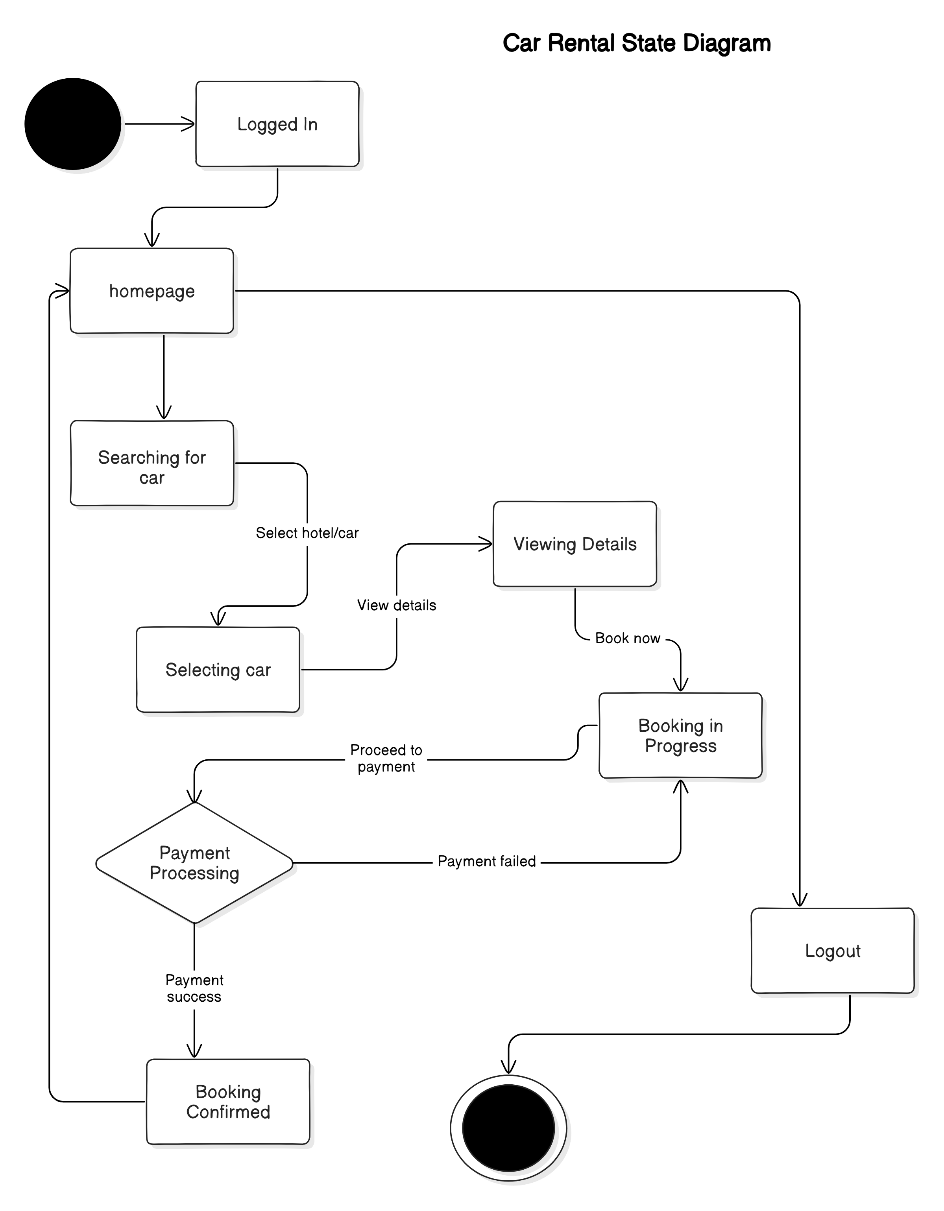


## Diagram details

The Booking state diagram represents the flow of a customer booking a car. The process begins with the user searching for a car. Once the customer makes a selection, they proceed to view details of the selected option. If satisfied, they initiate the booking process, transitioning to the "Booking in Progress" state.

In the next step, the system processes the payment. A decision point determines whether the payment is successful. If the payment is confirmed, the process concludes with "Booking Confirmed." If the payment fails, the system redirects the user back to the booking stage for corrections.

## Diagram



# Non-functional Requirements / Quality Attributes

|  |  |
| --- | --- |
| **Sr#** | **Requirements** |
| 1 | The app should not use more than 500 MB of memory when multiple features like maps, weather updates, and hotel reservations are being used simultaneously. |
| 2 | The app should load the main dashboard within 3 seconds on a standard 4G network |
| 3 | The app should be able to support up to 200 users booking hotels or navigating to destinations simultaneously without slowing or crashing |
| 4 | The app should be compatible with mobile devices running Android version 8.0 or higher and IOS version 12.0 or higher |
| 5 | The app should ensure that all screens maintain a consistent response time under 2 seconds with interacting with API calls |
| 6 | The app must be available 99.9% of the time over a period of 30 days, excluding planned maintenance |
| 7 | The system should recover and resume normal operations in less than 5 minutes in case of a failure |
| 8 | The app should display search results for tourist locations and services (e.g., hotels, restaurants) within 2 seconds after the user initiates a search. |
| 9 | The app should sync live weather and traffic updates within 2 minutes of change for any destination. |
| 10 | The app should provide feedback or error handling within 1 second of an invalid search query or failed booking attempt. |
| 11 | The app’s user interface should remain responsive (i.e., no lag) even when navigating between different modules like hotel booking, car rentals, and live weather updates. |

• **Security Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr#** | **Security Risks** | **Potential Losses** | **Controls** |
| 1 | A01:2021 - Broken Access Control | Unauthorized users gaining access to sensitive information    Loss of customer trust    Legal consequences due to privacy violations | Implement role-based access control (RBAC) to ensure that only authorized users can perform specific actions.    Ensure access controls are enforced consistently across all applications and API endpoints    Implement input validation and output encoding to prevent unauthorized data exposure    Enable audit logging to track access control violations |
| 2 | A02:2021 - Cryptographic Failures | Compromise of sensitive data (e.g., customer details, booking information)    Financial loss due to data theft attacks    Non-compliance with data protection regulations such as GDPR | Using strong, industry-standard encryption algorithms (e.g., AES-256) for data at rest and in transit    Implement secure key management practices, ensuring cryptographic keys are stored securely and rotated periodically    Use secure TLS/SSL protocols for all communications between the client and server    Avoid exposing sensitive data unnecessarily, especially in logs or error messages |
| 3 | A07:2021 - Identification and Authentication Failures | Identity theft and fraud    Potential legal repercussions | Implement multi-factor authentication (MFA) for users and administrators    Use secure password storage techniques; bcrypt or Argon2 to hash passwords.    Implement account lockout mechanisms to mitigate brute force attacks    Ensure secure session management such as secure cookies, expiring sessions after inactivity, and avoiding session identifiers in URLs |

# Who Did What?

|  |  |
| --- | --- |
| **Name of the Team Member** | **Tasks done** |
| Shahrez Faisal | 1,2,3 |
| Usman Arshid | 6 (State Diagrams) |
| Muhammad Mehdi | 4 (Class Diagram and description) |
| Umer Inayat | 7 (Nonfunctional requirements) |
| Omar Ibne Sajjad | 5 (Sequence Diagrams) |

# Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

|  |  |
| --- | --- |
| **Section** **Title** | **Reviewer Name(s)** |
| 7 (Nonfunctional requirements) | Usman Arshid |
| 1,2,3 | Muhammad Mehdi |
| 4 | Shahrez Faisal |
| 6 | Omar Ibne Sajjad |