**The Bug Catchers**

**Hoteru**

**Sofrware Architecture Document**

**Version <1.0>**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 28/07/2023 | <1.0> | The Architecture Document’s initialization | Dang Quoc Thai |
| 17/08/2023 | <1.1> | Updates of the Components, and addition of Deployment and Implementation View | Dang Quoc Thai |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

|  |  |
| --- | --- |
| 1. Introduction | 2 |
| 2. Architectural Goals and Constraints | 2 |
| 3. Use-Case Model | 3 |
| 4. Logical View | 13 |
| 4.1 Component: Hotel Room Management | 13 |
| 4.2 Component: User Management | 14 |
| 4.3 Component: Admin Tool | 15 |
| 5. Deployment | 16 |
| 6. Implementation View | 16 |

**Software Architecture Document**

**1. Introduction**

The Software Architecture Document aims to provide the Developers, Testers, as well as the Product’s Actors the Structure of the Hoteru System, how it works, and how each component interacts with each other, to provide the team the plan and the insights of what should be developed, and to provide the actors everything they need to know about the System, its structure and how to use the structure correctly and efficiently.

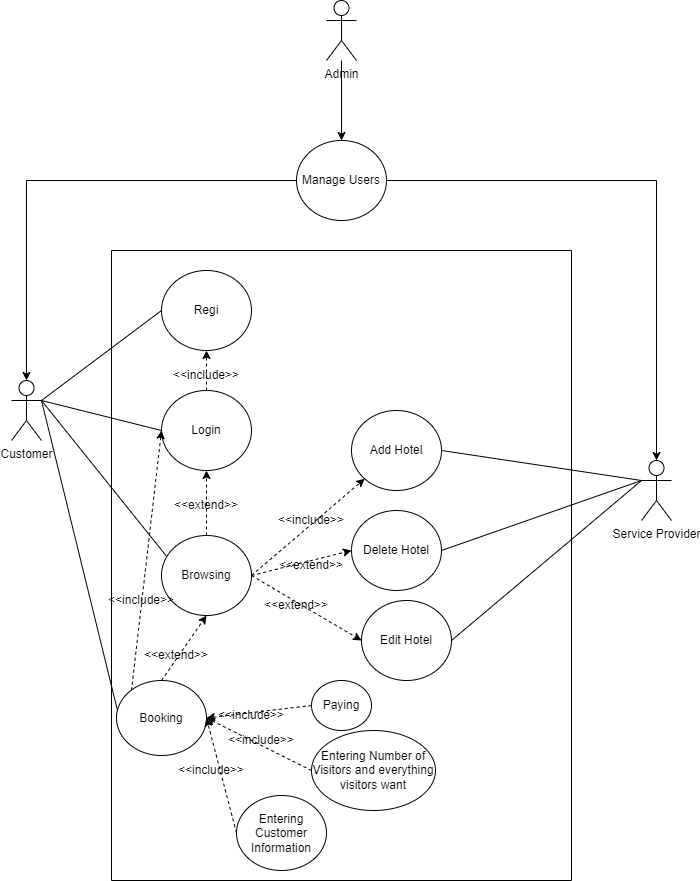
The document consists of The Goals and Constraints the System Structure meet, the Use Case Model demonstrating the relationship between actors and the system, and the components within with each other, as well as clarifying what each component does, and how it affects the rest of the system.

**2. Architecture Goals and Constraints**

The Goals of the Architecture is to construct a website system dedicated for Hotel Room Management, User base Management, and Management for all necessary booking information, so that all actors can learn to use it very quickly, with very simple and comprehensible methods, and so that all the related data flows to the right location, with the fastest speed.

The Architecture uses the Bootstrap Framework, meaning while it saves time, encourages teamwork, offers an excellent Grid System and is responsive, the Project must also take its cons, being that it has a learning curve, and that it can be heavy in terms of slower loading times, battery draining issues and its generated large files slowing the processes heavily, into account, and make sure to minimize its disadvantages as much as possible. Also, with the system developed in PHP Language, the team must deal with its lack of versatility, limited debugging tools, inability to modify core behavior, and lack security.

**3. Use-case Model**



|  |
| --- |
| **Use case: Manage Users** |
| **ID:** U01 |
| **Actors:** Admin |
| **Preconditions**:  -The Administrator is logged into the system.  -The Administrator has the necessary permissions to manage users. |
| Postconditions: The user management operations are successfully performed. |
| **Flow of events:**  1. Administrator selects "Manage Users" from the admin dashboard.  2. System displays a list of existing users with details.  3. Administrator selects a user to update.  4. System presents user details for editing.  5. Administrator modifies user information.  6. Administrator saves changes.  7. Administrator selects "Add New User."  8. System prompts to enter new user details.  9. Administrator selects a user to delete.  10. System displays user details and delete option.  11. Administrator confirms deletion. |
| **Alternative flow:**  1. Administrator cancels the update.  2. Administrator cancels adding a new user.  3. Administrator cancels user deletion. |

|  |
| --- |
| **Use case: Register** |
| **ID:** U02 |
| **Actors:** Customer |
| **Preconditions**: 1. The user must be accessing the system as a customer.  2. The Customer has not registered or logged in yet. |
| **Postconditions:** The Customer successfully registers an account in the system. |
| **Flow of events:** 1. Customer selects the "Register" or "Sign Up" option on the website.  2. System presents a registration form with required fields, such as name, email, password, etc.  3. Customer fills in the necessary information.  4. Customer submits the registration form.  5. System validates the provided information (e.g., checks for valid email format, password strength, etc.).  6. If the provided information is valid, the system creates a new customer account.  7. Customer receives a confirmation message or email about successful registration. |
| **Alternative flow:** 1. If the provided information is not valid, the system displays an error message highlighting the fields that need correction.  2. If the customer's email already exists in the system (i.e., the email is already registered), the system prompts the customer to log in instead of creating a new account. |

|  |
| --- |
| **Use case: Login** |
| **ID:** U03 |
| **Actors:** Customer |
| **Preconditions:** The user must be a registered customer with a valid account. |
| **Postconditions:** The customer successfully logs into the system and gains access to their account. |
| **Flow of events:** 1. Customer accesses the website or application.  2. System presents the login page or modal.  3. Customer enters their username/email and password.  4. Customer submits the login information.  5. System validates the provided login credentials.  6. If the provided credentials are valid, the system grants access to the customer's account and redirects them to the user dashboard or home page.  7. Customer gains access to their account and can use the system's features. |
| **Alternative flow:**  1. If the provided login credentials are incorrect or do not match any existing account, the system displays an error message and prompts the customer to re-enter their username/email and password.  2. If the customer forgot their password, the system provides a "Forgot Password" option to reset their password through a password recovery process. |

|  |
| --- |
| **Use case: Browsing** |
| **ID:** U04 |
| **Actors:** Customer |
| **Preconditions**:  1. The customer must be logged into the system or have access to the website/application.  2. The customer must have an active internet connection. |
| **Postconditions:** The customer can view available hotels and their details. |
| **Flow of events:**  1. Customer accesses the website or application.  2. System displays the homepage  3. Customer selects the "Browse Hotels" or similar option from the navigation menu.  4. System presents a list of available hotels with basic information, such as hotel names, locations, and prices.  5. Customer can apply filters (location, price range, amenities...) to refine the search results.  6. Customer selects a specific hotel from the list to view more details.  7. System displays detailed information about the selected hotel, including descriptions, photos, amenities, guest reviews, and availability.  8. Customer can click on a "Book Now" or "View Availability" button to proceed with the booking process.  9. System redirects the customer to the booking page or reservation form. |
| **Alternative flow:** 1. If there are no hotels available that match the customer's criteria, the system displays a message indicating that no results were found and suggests adjusting the search filters.  2. If the customer decides not to proceed with any of the listed hotels, they can go back to the list or perform a new search with different criteria. |

|  |
| --- |
| **Use case: Browsing** |
| **ID:** U04 |
| **Actors:** Customer |
| **Preconditions**:  1. The customer must be logged into the system or have access to the website/application.  2. The customer must have an active internet connection. |
| **Postconditions:** The customer can view available hotels and their details. |
| **Flow of events:**  1. Customer accesses the website or application.  2. System displays the homepage  3. Customer selects the "Browse Hotels" or similar option from the navigation menu.  4. System presents a list of available hotels with basic information, such as hotel names, locations, and prices.  5. Customer can apply filters (location, price range, amenities...) to refine the search results.  6. Customer selects a specific hotel from the list to view more details.  7. System displays detailed information about the selected hotel, including descriptions, photos, amenities, guest reviews, and availability.  8. Customer can click on a "Book Now" or "View Availability" button to proceed with the booking process.  9. System redirects the customer to the booking page or reservation form. |
| **Alternative flow:** 1. If there are no hotels available that match the customer's criteria, the system displays a message indicating that no results were found and suggests adjusting the search filters.  2. If the customer decides not to proceed with any of the listed hotels, they can go back to the list or perform a new search with different criteria. |

|  |
| --- |
| **Use case: Booking** |
| **ID:** U05 |
| **Actors:** Customer |
| **Preconditions**:  1. The customer must be logged into the system or have access to the website/application.  2. The customer must have selected a specific hotel and room type from the browsing process. |
| **Postconditions:** The customer successfully completes the booking process and receives a confirmation of the reservation. |
| **Flow of events:** 1. Customer selects a specific hotel and room type from the browsing results.  2. System displays detailed information and availability of the selected hotel and room type.  3. Customer clicks on the "Book Now" button next to the preferred room option.  4. System redirects the customer to the booking form.  5. Customer fills in the required booking details, such as check-in date, check-out date, number of guests, contact information, payment method, etc.  6. Customer reviews the booking details for accuracy.  7. Customer submits the booking/reservation form.  8. System validates the provided information (e.g., check-in date is not in the past, payment details are valid).  9. If the provided information is valid, the system confirms the booking and reserves the selected room.  10. System sends a confirmation email or message to the customer with booking details and reservation number. |
| **Alternative flow:** 1. If the customer decides not to proceed with the booking, they can click on the "Cancel" or "Go Back" button and return to the browsing page.  2. If there are any errors or missing information in the booking form, the system displays an error message and prompts the customer to correct the required fields.  3. If the payment method is invalid or there are issues with processing the payment, the system displays an error message and suggests using an alternative payment method. |

|  |
| --- |
| **Use case: Add Hotel** |
| **ID:** U06 |
| **Actors:** Hotel Provider |
| **Preconditions**:  1. The hotel provider must be registered and logged into the system as a verified hotel partner.  2. The hotel provider must have the necessary authorization to add new hotels to the system. |
| **Postconditions**: The new hotel is successfully added to the system's database and becomes available for customers to view and book. |
| **Flow of events:** 1. Hotel provider logs into the system with their credentials.  2. System authenticates the hotel provider's login details and grants access to the hotel provider dashboard.  3. Hotel provider navigates to the "Add Hotel” in their dashboard.  4. System presents a form for the hotel provider to input the hotel's information, such as hotel name, address, contact details, room types, amenities, photos, and other relevant details.  5. Hotel provider fills in the required information on the form.  6. Hotel provider submits the form to add the new hotel to the system.  7. System validates the provided information to ensure it meets the required format and criteria.  8. If the provided information is valid, the system adds the new hotel to the database and makes it available for customers to view and book.  9. System sends a confirmation message to the hotel provider indicating that the hotel has been successfully added. |
| **Alternative flow:** 1. If the hotel provider decides not to proceed with adding the hotel, they can click on the "Cancel" or "Go Back" button and return to their dashboard.  2. If there are any errors or missing information in the form, the system displays an error message and prompts the hotel provider to correct the required fields. |

|  |
| --- |
| **Use case: Delete Hotel** |
| **ID:** U07 |
| **Actors:** Hotel Provider |
| **Preconditions:**  1. The hotel provider must be registered and logged into the system as a verified hotel partner.  2. The hotel provider must have the necessary authorization to delete new hotels from the system. |
| **Postconditions:** The specified hotel is successfully deleted from the system's database and is no longer available for customers to view or book. |
| **Flow of events:** 1. Hotel provider logs into the system with their credentials.  2. System authenticates the hotel provider's login details and grants access to the hotel provider dashboard.  3. Hotel provider navigates to the "Manage Hotels” in their dashboard.  4. System presents a list of hotels associated with the hotel provider's account.  5. Hotel provider selects the hotel they want to delete from the list.  6. Hotel provider clicks on the "Delete" or similar option for the selected hotel.  7. System displays a confirmation message to ensure the hotel provider's intention to delete the hotel.  8. Hotel provider confirms the deletion by clicking "Yes" or a similar confirmation action.  9. System removes the selected hotel from the system's database and updates the list of hotels in the hotel provider's dashboard. |
| **Alternative flow:** 1. If the hotel provider decides not to proceed with deleting the hotel, they can click on the "Cancel" or "Go Back" button and return to their dashboard.  2. If the hotel provider mistakenly selects the wrong hotel to delete, they can click on "No" or a similar action to cancel the deletion confirmation. |

|  |
| --- |
| **Use case: Edit Hotel** |
| **ID:** U08 |
| **Actors:** Hotel Provider |
| **Preconditions:**  1. The hotel provider must be registered and logged into the system as a verified hotel partner.  2. The hotel provider must have the necessary authorization to edit hotel information in the system. |
| **Postconditions:** The changes made to the hotel information are successfully updated in the system's database. |
| **Flow of events:** 1. Hotel provider logs into the system with their credentials.  2. System authenticates the hotel provider's login details and grants access to the hotel provider dashboard.  3. Hotel provider navigates to the "Manage Hotels" or similar option in their dashboard.  4. System presents a list of hotels associated with the hotel provider's account.  5. Hotel provider selects the hotel they want to edit from the list.  6. Hotel provider clicks on the "Edit" for the selected hotel.  7. System displays a form with the existing hotel information, allowing the hotel provider to make changes to any desired fields, such as hotel name, address, contact details, room types, amenities, photos, and other relevant details.  8. Hotel provider makes the necessary changes in the form.  9. Hotel provider submits the form to update the hotel information in the system.  10. System validates the provided information to ensure it meets the required format and criteria.  11. If the provided information is valid, the system updates the hotel information in the database.  12. System sends a confirmation message to the hotel provider indicating that the changes have been successfully saved. |
| **Alternative flow:**  1. If the hotel provider decides not to proceed with editing the hotel, they can click on the "Cancel" or "Go Back" button and return to their dashboard.  2. If there are any errors or missing information in the form, the system displays an error message and prompts the hotel provider to correct the required fields. |

|  |
| --- |
| **Use case: Paying** |
| **ID:** U09 |
| **Actors:** Customer |
| **Preconditions:**  1. The customer must have selected a hotel and completed the booking process.  2. The customer must have entered valid payment information. |
| Postconditions: 1. The payment is successfully processed, and the booking is confirmed.  2. The customer receives a payment receipt and booking confirmation. |
| **Flow of events:** 1. Customer selects a hotel and enters the required details for booking, such as check-in/out dates, number of guests, and room preferences.  2. System displays the total booking cost and prompts the customer to proceed with payment.  3. Customer selects the preferred payment method (credit card, debit card, PayPal).  4. Customer enters the necessary payment details, such as card number, expiration date, CVV, or logs in to their payment account.  5. System securely processes the payment information through a payment gateway.  6. System verifies the payment details and checks for sufficient funds or valid payment credentials.  7. If the payment is successful, the system confirms the booking and generates a booking reference number.  8. Customer receives a payment receipt and booking confirmation via email or on-screen notification. |
| **Alternative flow:** 1. If the payment information provided by the customer is invalid or insufficient, the system displays an error message and prompts the customer to re-enter the payment details or choose an alternative payment method.  2. If the payment is declined by the payment gateway or financial institution, the system informs the customer of the declined transaction and provides guidance on resolving the payment issue or using an alternative payment method.  3. If the customer encounters any issues with the booking confirmation or payment receipt, they can contact customer support for assistance. |

|  |
| --- |
| **Use case: Entering Number of Visitors and everything visitors want** |
| **ID:** U10 |
| **Actors**: Customer |
| **Preconditions:** 1. The customer must have selected a hotel and proceeded to the booking process.  2. The customer must have logged into their account or provided necessary contact details for booking confirmation. |
| **Postconditions:** The customer's preferences for the number of visitors, room type, and any additional requests are successfully recorded in the booking details. |
| **Flow of events:** 1. Customer selects a hotel and enters the required details for booking, such as check-in/out dates, number of guests, and room preferences (single room, double room, suite).  2. System displays available room options based on the entered details.  3. Customer selects the desired room type and indicates the number of visitors accompanying them.  4. Customer specifies any special requirements or preferences, such as bed preferences (twin beds, king-size bed), smoking or non-smoking room, accessibility features, or any other specific requests in the provided text field.  5. System updates the booking summary with the selected room type, number of visitors, and any additional preferences.  6. Customer reviews the booking details to ensure accuracy and completeness.  7. Customer proceeds to the next step, such as entering payment information or confirming the reservation. |
| **Alternative flow:** 1. If the customer enters an invalid or unavailable number of visitors (exceeding maximum occupancy), the system displays an error message and prompts the customer to correct the entered information.  2. If the customer's preferences or special requests cannot be accommodated by the hotel or are not available, the system informs the customer and offers alternative options if applicable.  3. If the customer wants to modify the entered preferences or special requests, they can go back to the previous step and make necessary changes before proceeding with the booking confirmation. |

|  |
| --- |
| **Use case: Entering Customer Information** |
| **ID:** U11 |
| **Actors:** Customer |
| **Preconditions:** The customer must have selected a hotel and proceeded to the booking process. |
| **Postconditions:** The customer's personal information, including name, contact details, and payment information (if applicable), is successfully recorded for the booking. |
| **Flow of events:** 1. Customer selects a hotel, enters the required details for booking, and specifies any preferences (number of visitors, room type, special requests...).  2. System prompts the customer to provide their personal information for booking confirmation.  3. Customer enters their full name, email address, phone number, and any other required contact details.  4. If applicable, customer enters the payment information, such as credit card or PayPal details.  5. Customer reviews the entered information for accuracy.  6. Customer confirms the entered details and proceeds to complete the booking. |
| **Alternative flow:** 1. If the customer enters an invalid or incomplete personal information the system displays an error message and prompts the customer to correct the entered information.  2. If the customer's payment information is invalid or incomplete, the system displays an error message and prompts the customer to correct the payment details or choose an alternative payment method. |

**4. Logical View**

4.1 Component: Hotel Room Management

Hotel System is a subsystem designed for every necessary information for every hotel room the database store, receive and maintain, as well as every functions for every important operation surrounding the database, including booking and payment procedures

Room

- id

- facility

- available\_num

- room\_type

- description

- price

Hotel

- id

- title

- location

- facility

- description

+ bookRoom()

+ pay()

+ cancelBooking()

- visitors

- customer\_id

- customer\_email

- pay\_method

- check\_in

- check\_out

Room\_Booking

+ addRoom()

+ deleteRoom()

+ editRoom()

+ addHotel()

+ deleteHotel()

+ editHotel()

Key Classes:

- Hotel: Storing, maintaining and managing all hotel information

- Room: Storing, maintaining and managing the hotel rooms that are within a hotel

- Room\_Booking: allows the customer to make a Hotel Room Booking, perform payment to the system, as well as cancelling a booking if necessary

4.2 Component: User Management

User Management allows the customer to create an account of their own, as well as editing their Information and performing activities like browsing and booking.

account

- username

- password

booking

- user\_id

- name

- email

- address

- phone\_num

- profile\_pic

customer

+ register()

+ login()

+ deactivate()

+ changepassword()

+ update\_info()

+ feedback()

+ qna()

+ browsing()

- hotel\_id

- room\_id

- pay\_method

- check\_in

- check\_out

- cost

+ bookRoom()

+ pay()

+ cancelBooking()

Key Classes:

- account: Contains the information of username and password for every account. Can be used for registering a new account, logging into an existing account, disabling the account or change the account’s password in case the user forgot it, or wants a newer one.

- customer: the information of the customer related to a specific account. It includes Name, Email, Address, Phone Number and Profile Picture. Customer can edit their information, or use it to make feedback, make questions to the service providers and admins, and perform browsing on the system.

- booking: allows the customer to make a Hotel Room Booking, perform payment to the system, as well as cancelling a booking if necessary. Includes info about Hotel Rooms, Payment Method, Check-in and Check-out Date, and the Booking Cost.

4.3 Component: Admin Tool

The Component helping the actors called “admins” to monitor, configure and maintain the system and its database, its users and its team

- users

- bookings

- queries

- reviews

- page\_status

Dashboard

+ add\_member()

+ delete\_member()

- member\_name

- profile\_pic

Management team

+ edit\_info()

- address

- google map

- phone\_num

- email

- fb

- tw

-insta

Contact Info

+ edit\_info()

- site\_name

- site\_info

Site Info

- admin\_name

- password

+ register()

+ login()

+ deactivate()

+ changepassword()

admin

Key Class:

- admin: The Core class containing every admin name and password, and is also responsible for registering a new admin, allowing an admin to log in, deactivate admin and change password for admin.

- Site Info: The place containing the tool admins need to change the site’s name, the site’s about, and toggle between the shutdown mode and the normal mode of the website

- Contact Info: Has a tools necessary to edit the contact information, including social media site, address and e-mail address

- Management Team: Allows admins to add new member, including their name, their profile picture, as well as deleting them at anytime

- Dashboard: Shows admins every statistics about booking of every kind, users, queries and reviews

**5. Deployment**

**A diagram of a web server

Description automatically generated**

- Customer Client: The Web Browser from the Customer’s end of the system, allowing the users to perform User Management and Booking Management of their own account at Web Server via TCP/IP protocol

- Hotel Owner Client: Supported by the Service Provider, this Client can get access to Web Server with the TCP/IP protocol to view, validate and configure the Hotel Room Information and Booking Information

- Admin Client: Is the browser for the Administrator. Access the Webpage via TCP/IP Protocol, the admins can perform every action to maintain the website, from configuring data with the Admin Tools to Performing Booking, User and Hotel Room Management

**6. Implementation View**

