

**Birthday party booking for Kids**

# **I.Record of Changes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| 14/1/2024 | M | Quan | Update Non-Requirement |
| 28/1/2024 | M | Quan | Update Use case diagram |
| 28/1/2024 | M | Phu | Update Use case Booking party |
| 28/1/2024 | A | Phu | Add Use case Rating |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

# **II. Software Requirement Specification**

## **1. Product Overview**

### **1.1 Project Information**

**• Project name:** Birthday party booking for Kids

**• Project code:** BD

**• Group name:** SWD392\_BirthDay Party

**• Software type:** Web

### **1.2 Project Team**

### **1.3 OverView**

The Birthday Party Booking for Kids application is designed to simplify the process of planning and organizing your Kid's birthday party. It serves parents, guardians, and event organizers looking for an easy way to create memorable birthday celebrations for their children.

# 

# 

# **2. User Requirements**

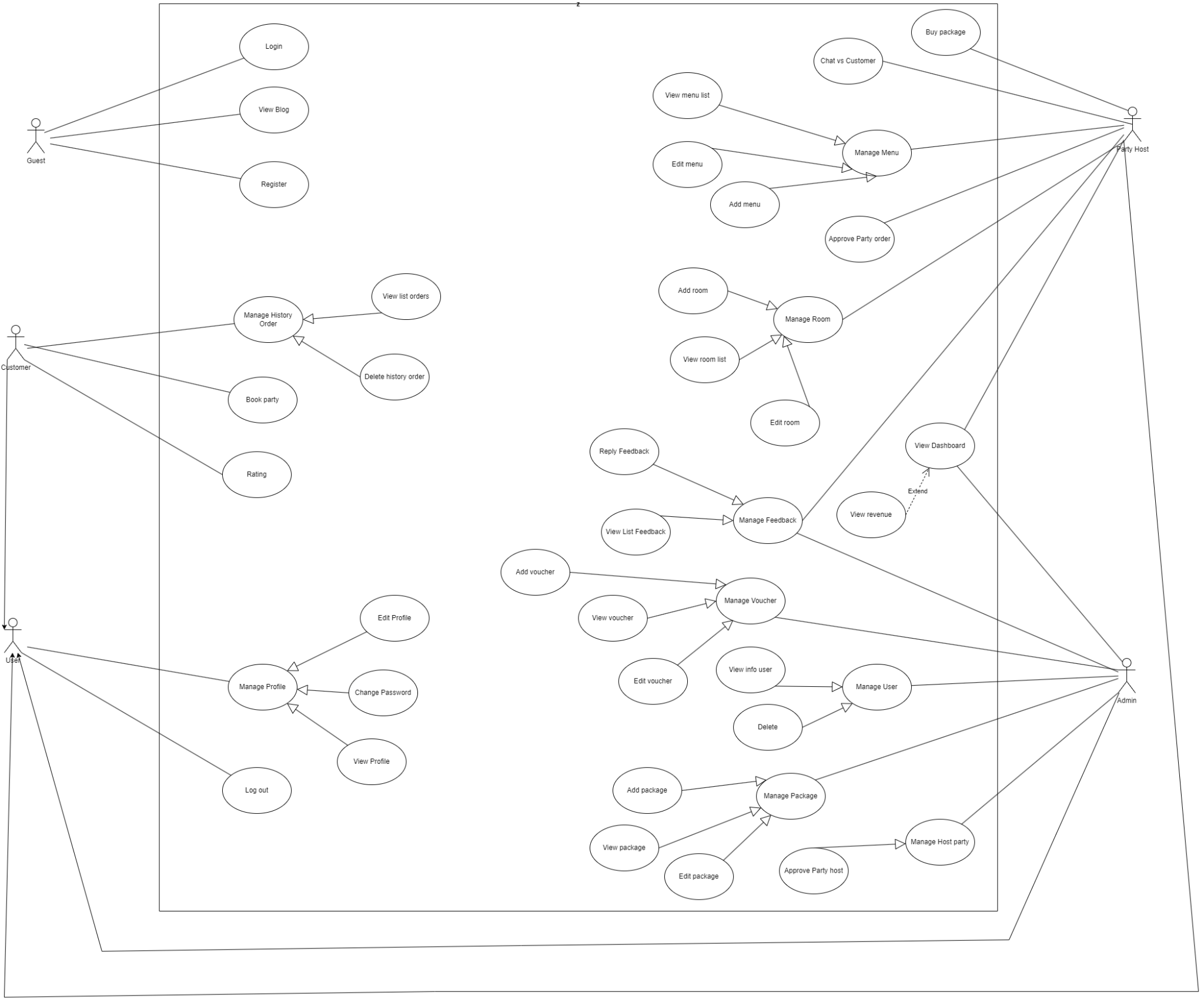
## **2.1 Actors**

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
|  | Admin | System administrator with privileged access, overseeing and managing the party management system. |
|  | Guest | Unregistered or unauthenticated user exploring limited features and public content. |
|  | Customer | User interested in booking and attending party events, capable of actions like feedback submission, booking, and payments. |
|  | Party Host | User responsible for hosting and organizing events, managing event-related actions, and ensuring a positive customer experience. |

## 

## **2.2 Use Cases**

### **2.2.1 Use case Diagram**



### **2.2.2 Use case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Use Case** | **Actors** | **Use Case Description** |
| UC-01 | Login | Guest | Allows a guest to log in to the system using valid credentials |
| UC-02 | View Blog | Guest | Permits guests to view the blog section of the platform. |
| UC-03 | Register | Guest | Enables a guest to create a new account on the platform by providing necessary registration details. |
| UC-04 | View List Orders | Customer | Allows customers to view a list of their orders. |
| UC-05 | Book Party | Customer | Enables customers to book a party, providing necessary details and preferences. |
| UC-06 | Delete History Order | Customer | Allows customers to delete their order history. |
| UC-07 | Rating | Customer | Permits customers to provide ratings for parties or services they have experienced. |
| UC-08 | Edit Profile | Party Host, Admin, Customer | Allows Party Hosts, Admins, and Customers to edit their profiles by updating personal information. |
| UC-09 | Change Password | Party Host, Admin, Customer | Enables Party Hosts, Admins, and Customers to change their account passwords. |
| UC-10 | View Profile | Party Host, Admin, Customer | Allows Party Hosts, Admins, and Customers to view their profiles. |
| UC-11 | Logout | Party Host, Admin, Customer | Allows Party Hosts, Admins, and Customers to log out of their accounts. |
| UC-12 | Chat with Customer | Party Host | Allows Party Hosts, Admins, and Customers to log out of their accounts. |
| UC-13 | View Menu | Party Host | Allows Party Hosts to view the menu for party events. |
| UC-14 | Edit Menu | Party Host | Permits Party Hosts to edit the menu for party events. |
| UC-15 | Add Menu | Party Host | Allows Party Hosts to add new menu items for party events. |
| UC-16 | Add Room | Party Host | Enables Party Hosts to add details for a new party room. |
| UC-17 | View Room List | Party Host | Allows Party Hosts to view a list of party rooms. |
| UC-18 | Edit Room | Party Host | Permits Party Hosts to edit details of a party room. |
| UC-19 | Buy package | Party Host | Allows Party Hosts to purchase party packages. |
| UC-20 | Reply FeedBack | Party Host, Admin | Enables Party Hosts and Admins to reply to customer feedback. |
| UC-21 | View List FeedBack | Party Host, Admin | Allows Party Hosts and Admins to view a list of customer feedback. |
| UC-22 | Add Package | Admin | Enables Admins to add a new package. |
| UC-23 | View Package | Admin | Allows Admins to view a list of party packages. |
| UC-24 | Edit Package | Admin | Permits Admins to edit details of a package. |
| UC-25 | View Info User | Admin | Allows Admins to view information about users. |
| UC-26 | Delete User | Admin | Enables Admins to delete user accounts. |
| UC-27 | View Voucher | Admin | Allows Admins to view a list of vouchers. |
| UC-28 | Edit Voucher | Admin | Permits Admins to edit details of a voucher. |
| UC-29 | Add Voucher | Admin | Enables Admins to add details for a new voucher. |
| UC-30 | Approve Party Host | Admin | Allows Admins to approve Party Host accounts. |
| UC-31 | View DashBoard | Admin, Party Host | Enables Admins, Party Host to view the dashboard with relevant metrics and information. |
| UC-32 | View Revenue | Admin, Party Host | Allows Admins, Party Host to view revenue-related information. |

### **2.2.3 Use case Specification**

#### **2.2.3.1 Login**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-01 - Login | | |
| **Created By:** | Hồ Tấn Bình | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Guest | **Secondary Actors:** |  |
| **Trigger:** | The Guest initiates the login process. | | |
| **Description:** | This use case enables Guests to log into the system. | | |
| **Preconditions:** | The Guest has an account.  The Guest is not currently logged into the system | | |
| **Post–conditions:** | The Guest successfully logs into the system. | | |
| **Normal Flow:** | 1. The Guest accesses the system. 2. The system prompts the Guest to enter their credentials. 3. The Guest enters their username and password. 4. The system validates the credentials. 5. If the credentials are valid, the system grants access, and the Guest is logged in. | | |
| **Alternative Flows:** | Alternative Flow : Login with Google   1. The Guest selects the option to log in with Google. 2. The system redirects the Guest to the Google login page. 3. The Guest enters their Google credentials. 4. Google validates the credentials. 5. If the credentials are valid, Google redirects the Guest back to the system, and the system grants access, logging in the Guest. | | |
| **Exceptions:** | If there are issues with the server connection during the login process:  The system reports an error and advises the Guest to try again later. | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **High, as Guests need to log in to access personalized features.** | | |
| **Business Rules:** | **BR-01:** The system enforces secure password policies.  **BR-02:** Accounts may be temporarily locked after a certain number of unsuccessful login attempts. | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### **2.2.3.2 View Blog**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-02 - View Blog | | |
| **Created By:** | Hồ Tấn Bình | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Guest | **Secondary Actors:** |  |
| **Trigger:** | The Guest initiates the request to view a blog. | | |
| **Description:** | The Guest initiates the request to view a blog. | | |
| **Preconditions:** | The Guest has access to the system.  The system has published blogs for viewing. | | |
| **Post–conditions:** | The Guest successfully views the selected blog. | | |
| **Normal Flow:** | 1. The Guest accesses the system. 2. The Guest clicks on a specific blog from the list. 3. The system presents a list of available blogs. 4. The Guest selects a specific blog from the list. 5. The system displays the content of the selected blog. | | |
| **Alternative Flows:** | N/A | | |
| **Exceptions:** | If there are no blogs available:  The system notifies the Guest that there are currently no blogs to view. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium, as Guests may want to explore the available blogs.** | | |
| **Business Rules:** | **BR-03:**.Blogs are periodically updated or added to the system. | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### 

#### 

#### **2.2.3.3 Register**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-03 - Register | | |
| **Created By:** | Hồ Tấn Bình | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Guest | **Secondary Actors:** |  |
| **Trigger:** | Guest decides to book a birthday party. | | |
| **Description:** | This use case allows Guest, to create a new account in the system | | |
| **Preconditions:** | The User has access to the system.  The User is not already registered with an account. | | |
| **Post–conditions:** | The User successfully creates a new account and gains access to the system. | | |
| **Normal Flow:** | 1. The User accesses the registration page. 2. The system presents a form requesting necessary registration information, such as username, email, and password. 3. The User fills in the required information. 4. The system validates the provided information. 5. If the information is valid, the system creates a new account for the User. | | |
| **Alternative Flows:** | N/A | | |
| **Exceptions:** | If there are issues with server connectivity or data validation during registration:  The system reports an error and advises the User to try again later. | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **Medium, as users register only when creating a new account.** | | |
| **Business Rules:** | **BR-04:** The system enforces secure password policies during registration.  **BR-05:** Each username and email must be unique. | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### 

#### **2.2.3.4 View List orders**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-04 - View list orders | | |
| **Created By:** | Dương Gia Phú | **Date Created:** | Jan-28, 2024 |
| **Primary Actor:** | Customer | **Secondary Actors:** |  |
| **Trigger:** | The customer selects the "View Orders" option from the web. | | |
| **Description:** | This use case describes the process by which a customer views the list of orders related to booking birthday parties through the system. | | |
| **Preconditions:** | The customer has previously placed orders for booking birthday parties. | | |
| **Post–conditions:** | The system displays the list of orders related to booking birthday parties. | | |
| **Normal Flow:** | 1. The customer logs into the system.  2. The customer navigates to the "View Orders" section.  3. The system retrieves and displays the list of orders associated with booking birthday parties.  4. The customer reviews the list of orders, including details such as order ID, party date, venue, and status. | | |
| **Alternative Flows:** | N/A | | |
| **Exceptions:** | If there are no orders related to booking birthday parties, display a message indicating that no orders are available. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Frequent** | | |
| **Business Rules:** | **BR-06:** The customer can only view orders that are associated with their account.  **BR-07:** The order list should be displayed in a user-friendly format with relevant details. | | |
| **Other Information:** | The system should provide sorting and filtering options for the order list to enhance the customer's experience. | | |
| **Assumptions:** | The customer has successfully placed orders for booking birthday parties in the past. | | |

#### **2.2.3.5 Book Party**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-05 - Book party | | |
| **Created By:** | Dương Gia Phú | **Date Created:** | Jan-28, 2024 |
| **Primary Actor:** | Customer | **Secondary Actors:** |  |
| **Trigger:** | Customer decides to book a birthday party. | | |
| **Description:** | This use case describes the process of a customer booking a birthday party at a designated venue or service. | | |
| **Preconditions:** | The customer has selected a specific date for the birthday party.  The customer has chosen the preferred venue or service for the party. | | |
| **Post–conditions:** | The booking is confirmed.  The customer receives a confirmation email with details of the booking. | | |
| **Normal Flow:** | 1. Customer logs into the booking system.  2. Customer selects the "Book Birthday Party" option.  3. Customer chooses the desired date for the party.  4. Customers choose their favorite party package.  5. Customers choose the menu for the party.  6. System checks the availability of the chosen date and package.  7. If available, Customer confirms the booking.  8. System generates a confirmation email and sends it to the customer.  9. The booking details are stored in the system. | | |
| **Alternative Flows:** | Customers can cancel the booking before confirming the birthday party booking. | | |
| **Exceptions:** | System encounters an error during the booking process. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **High** | | |
| **Business Rules:** | **BR-8:** A minimum advance notice of two weeks is required for booking. | | |
| **Other Information:** | The system will provide a list of available themes and services for the birthday party.  Confirmation email includes a summary of the booking, and contact information for inquiries. | | |
| **Assumptions:** | The system is connected to the database of available dates, venues, and services. | | |

#### **2.2.3.6 Delete History Order**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-06 - Delete history order | | |
| **Created By:** | Dương Gia Phú | **Date Created:** | Jan-28, 2024 |
| **Primary Actor:** | Customer | **Secondary Actors:** |  |
| **Trigger:** | Customer selects the option to delete the history order with the topic of booking a birthday party. | | |
| **Description:** | This use case describes the process of a customer deleting a history order with the specific topic of booking a birthday party. | | |
| **Preconditions:** | The customer has at least one history order related to booking a birthday party. | | |
| **Post–conditions:** | The selected history order with the birthday party booking theme will be hidden from the list. | | |
| **Normal Flow:** | 1. The customer logs into their account.  2. The customer navigates to the order history section.  3. The customer identifies the specific order with the topic of booking a birthday party that they want to delete.  4. The customer selects the option to delete the history order.  5. The system prompts the customer for confirmation.  6. The customer confirms the deletion.  7. The system hides the selected history order of the birthday party booking theme from the system.  8. The system updates the order history to reflect the deletion. | | |
| **Alternative Flows:** | If the customer decides not to proceed with the deletion after confirming, the system cancels the operation, and the order remains unchanged. | | |
| **Exceptions:** | If there are dependencies on the order (e.g., ongoing services or upcoming events related to the order), the system may prevent deletion and provide relevant information to the customer. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | **BR-9:** A history order with the topic of booking a birthday party can only be deleted by the customer who placed the order. | | |
| **Other Information:** | Deleting a history order with the topic of booking a birthday party does not affect any current or future bookings or services. | | |
| **Assumptions:** | The system provides appropriate feedback to the customer after the deletion process is completed. | | |

#### **2.2.3.7 Rating**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-07 - Rating | | |
| **Created By:** | Dương Gia Phú | **Date Created:** | Jan-28, 2024 |
| **Primary Actor:** | Customer | **Secondary Actors:** |  |
| **Trigger:** | Customers can view and check information about the event they have booked. | | |
| **Description:** | This use case involves the process of customers providing ratings for birthday party events they have booked through the system. The ratings will help the system and party organizers to improve their services and better meet customer expectations. | | |
| **Preconditions:** | The customer must have booked a birthday party event.  The booked birthday party event must be completed. | | |
| **Post–conditions:** | The rating provided by the customer is stored in the system for future reference. | | |
| **Normal Flow:** | 1. The customer receives a notification or prompt to provide a rating after the completion of the birthday party event.  2. The customer accesses the rating feature through the system interface.  3. The system displays a rating scale or options for the customer to choose from.  4. The customer selects a rating based on their satisfaction with the birthday party event.  5. The system records the provided rating and associates it with the specific birthday party booking. | | |
| **Alternative Flows:** | None. | | |
| **Exceptions:** | If the customer chooses not to provide a rating, the system does not record any rating information.  In case of a system error or technical issue, the customer may not be able to submit the rating. An error message should be displayed, and the issue should be resolved. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Frequent (after each booked birthday party event)** | | |
| **Business Rules:** | **BR-10:** Ratings should be on a predefined scale or set of options. | | |
| **Other Information:** | The system may use the collected ratings for analytical purposes, such as improving services or recognizing exceptional performances. | | |
| **Assumptions:** | The system has a mechanism for notifying customers to provide ratings after the completion of party. | | |

#### **2.2.3.8 Edit Profile**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-8 - Edit Profile | | |
| **Created By:** | Hồ Tấn Bình | **Date Created:** | Jan-14, 2024 |
| **Primary Actor:** | Party Host, Customer, Admin | **Secondary Actors:** |  |
| **Trigger:** | The Actor initiates the request to edit their own user profile. | | |
| **Description:** | This use case allows Actors to modify and update their own profiles within the system. | | |
| **Preconditions:** | The Actor has successfully logged into the system.  The Actor has navigated to the "Edit Profile" section. | | |
| **Post–conditions:** | The Actor successfully edits and saves changes to their own user profile. | | |
| **Normal Flow:** | 1. The Actor logs into the system. 2. The Actor navigates to the "Edit Profile" section. 3. The system presents a form with editable fields containing the Actor's current profile information. 4. The Actor modifies the desired fields (e.g., username, email, bio). 5. The system validates the updated information. 6. If the information is valid, the system saves the changes to the Actor's profile. | | |
| **Alternative Flows:** | If there are issues with validating the updated information:  The system prompts the Actor to correct errors before saving. | | |
| **Exceptions:** | If there are issues with server connectivity or data validation during the editing process:  The system reports an error and advises the Actor to try again later. | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **Medium, as Party Hosts, Customers, and Admins may need to update their profiles periodically.** | | |
| **Business Rules:** | **BR-11:** User profiles are regularly updated with the latest information | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Party hosts have a stable internet connection to access the system.  The system's data is accurate and reliable. | | |

#### **2.2.3.9 Change password**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-09 - Change password | | |
| **Created By:** | Hồ Tấn Bình | **Date Created:** | Jan-14, 2024 |
| **Primary Actor:** | Party Host, Customer, Admin | **Secondary Actors:** |  |
| **Trigger:** | The Actor initiates the request to change their own password. | | |
| **Description:** | This use case enables Party Hosts, Customers, and Administrators (referred to as the Actor) to change their current password within the system. | | |
| **Preconditions:** | The Actor has successfully logged into the system.  The Actor has navigated to the "Change Password" section. | | |
| **Post–conditions:** | The Actor successfully changes their password and can log in with the new credentials. | | |
| **Normal Flow:** | 1. The Actor logs into the system. 2. The Actor navigates to the "Change Password" section. 3. The system prompts the Actor to enter their current password and the new desired password. 4. The Actor enters the required information. 5. The system validates the entered information. 6. If the information is valid, the system updates the Actor's password. | | |
| **Alternative Flows:** | If the Actor's current password is entered incorrectly:  The system prompts the Actor to re-enter the correct current password. | | |
| **Exceptions:** | If there are issues with server connectivity or data validation during the password change process:  The system reports an error and advises the Actor to try again later. | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **Low to Moderate, as password changes are typically infrequent but essential for security.** | | |
| **Business Rules:** | **BR-12:** The system enforces secure password policies during the change process. | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Party hosts have a stable internet connection to access the system.  The system's data is accurate and reliable. | | |

#### **2.2.3.10 View Profile**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-10 - View Profile | | |
| **Created By:** | Hồ Tấn Bình | **Date Created:** | Jan-14, 2024 |
| **Primary Actor:** | Party Host, Customer, Admin | **Secondary Actors:** |  |
| **Trigger:** | The Actor initiates the request to view their user profile. | | |
| **Description:** | This use case allows Actors to view their own profiles within the system. | | |
| **Preconditions:** | The Actor has successfully logged into the system. | | |
| **Post–conditions:** | The Actor successfully views their own user profile. | | |
| **Normal Flow:** | 1. The Actor logs into the system. 2. The Actor navigates to the "View Profile" section. 3. The system automatically displays the profile information of the logged-in Actor. | | |
| **Alternative Flows:** | None | | |
| **Exceptions:** | If there are issues with retrieving the Actor's own profile information:  The system reports an error and advises the Actor to try again later. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium, as Party Hosts, Customers, and Admins may want to view and verify their own profile information.** | | |
| **Business Rules:** | **BR-13:** User profiles are regularly updated with the latest information. | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Actors have a stable internet connection to access the system.  The system's data is accurate and reliable. | | |

#### **2.2.3.11 Log out**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-11 - Log out | | |
| **Created By:** | Hồ Tấn Bình | **Date Created:** | Jan-14, 2024 |
| **Primary Actor:** | Party Host, Customer, Admin | **Secondary Actors:** |  |
| **Trigger:** | The Actor initiates the request to log out of the system. | | |
| **Description:** | This use case allows Actor to safely terminate their current session within the system. | | |
| **Preconditions:** | The Actor has successfully logged into the system. | | |
| **Post–conditions:** | The Actor is successfully logged out, and the session is terminated. | | |
| **Normal Flow:** | 1. The Actor, having completed their tasks, navigates to the "Log Out" section or clicks on the "LogOut" button. 2. The system clears the session data and logs the Actor out of the system. 3. The system displays a confirmation message indicating successful log out. | | |
| **Alternative Flows:** | None | | |
| **Exceptions:** | If there are issues with server connectivity or data validation during the log out process:  The system reports an error but still attempts to log out the Actor. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Frequently, actors may log out after completing their tasks or to secure their sessions.** | | |
| **Business Rules:** | **BR-14:** The system clears sensitive session data during the log out process to enhance security. | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Party hosts have a stable internet connection to access the system.  The system's data is accurate and reliable. | | |

#### 

#### **2.2.3.12 Chat with Customer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-12-Chat with Customer | | |
| **Created By:** | Trần Quốc An | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Party Host | **Secondary Actors:** | Customer |
| **Trigger:** | The Party Host initiates a chat with a customer attending the event. | | |
| **Description:** | This use case allows the party host to have a real time chat conversation with a customer to provide information | | |
| **Preconditions:** | The Party Host and Customer have access to the chat system. | | |
| **Post–conditions:** | The party host provides the necessary information or assistance to the customer. | | |
| **Normal Flow:** | 1. The Party Host logs into the system.  2. Party Host selects the "Chat" option from the main interface.  3. The system displays a list of all Customer's previous chat messages.  4. Party Host selects a specific chat to view responses messages.  5. The system displays chat messages details from Customer | | |
| **Alternative Flows:** | If the system has no previous chat message, the system notifies and Party Host can start a new chat with their Customer. | | |
| **Exceptions:** | If it cannot connect to the database, the system reports an error and asks the user to try again later. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium, especially after each party event.** | | |
| **Business Rules:** |  | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### **2.2.3.13 View Menu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-13-View Menu | | |
| **Created By:** | Trần Quốc An | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Party Host | **Secondary Actors:** |  |
| **Trigger:** | The Party Host can View all of the existing Menu List they created. | | |
| **Description:** | This use case allows the Party Host to view all of the existing Menu they had created | | |
| **Preconditions:** | The Party Host has access to view a list of Menu. | | |
| **Post–conditions:** | Party Host successfully views all of the existing Menu. | | |
| **Normal Flow:** | 1. Party Host logs into the system. 2. Party Host selects the "Party" option from the main interface. 3. Party Host selects the "List of Menu" option. 4. The system displays all of the existing Menu. | | |
| **Alternative Flows:** | If the Menu list is empty, the system will notify the Party Host to create a new one. | | |
| **Exceptions:** | If it cannot connect to the server, the system reports an error and asks the user to try again later. | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **High, especially upon login.** | | |
| **Business Rules:** |  | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### **2.2.3.14 Edit Menu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-14-Edit Menu | | |
| **Created By:** | Trần Quốc An | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Party Host | **Secondary Actors:** |  |
| **Trigger:** | The Party Host can Editall of the existing Menu List they created. | | |
| **Description:** | This use case allows the Party Host to edit all of the existing Menu they had created | | |
| **Preconditions:** | The Party Host has access to edit an existing Menu. | | |
| **Post–conditions:** | Party Host successfully edit the existing Menu. | | |
| **Normal Flow:** | 1. Party Host logs into the system. 2. Party Host selects the "Party" option from the main interface. 3. Party Host selects the "List of Menu" option. 4. The system displays all of the existing Menu. 5. The system displays the information that Party Host can edit.. 6. The Party Host fills in all the information that they want to update. 7. Click Update to finish. | | |
| **Alternative Flows:** | If the Menu list is empty, the system will notify the Party Host to create a new one. | | |
| **Exceptions:** | If it cannot connect to the server, the system reports an error and asks the user to try again later. | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **High, especially upon login.** | | |
| **Business Rules:** |  | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### **2.2.3.15 Add Menu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-15-Add Menu | | |
| **Created By:** | Trần Quốc An | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Party Host | **Secondary Actors:** |  |
| **Trigger:** | The Party Host creates a new Menu. | | |
| **Description:** | This use case allows the Party Host to create a new Menu. | | |
| **Preconditions:** | The Party Host has access to create a new Menu. | | |
| **Post–conditions:** | Party Host successfully created a new Menu. | | |
| **Normal Flow:** | 1. Party Host logs into the system. 2. Party Host selects the "Menu" option from the main interface. 3. Party Host selects the "Add New Menu" option. 4. The system displays the information to fill in. 5. The Party Host fills in all the required information. 6. Click Add to finnish | | |
| **Alternative Flows:** | If the Party Host left requires information blank, the system will notify Party Host to fill all of it to add a new Menu. | | |
| **Exceptions:** | If it cannot connect to the server, the system reports an error and asks the user to try again later. | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **High, especially upon login.** | | |
| **Business Rules:** |  | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### **2.2.3.16 Add Room**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-16-Add Room | | |
| **Created By:** | Trần Quốc An | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Party Host | **Secondary Actors:** |  |
| **Trigger:** | The Party Host creates a new Room. | | |
| **Description:** | This use case allows the Party Host to create a new Room. | | |
| **Preconditions:** | The Party Host has access to create a new Room. | | |
| **Post–conditions:** | The Party Host successfully created a new Room. | | |
| **Normal Flow:** | 1. Party Host logs into the system. 2. Party Host selects the "Room" option from the main interface. 3. Party Host selects the "Add New Room" option. 4. The system displays the information to fill in. 5. The Party Host fills in all the required information. 6. Click Add to finnish | | |
| **Alternative Flows:** | If the Party Host left requires information blank, the system will notify Party Host to fill all of it to add a new Menu. | | |
| **Exceptions:** | If it cannot connect to the server, the system reports an error and asks the user to try again later. | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **High, especially upon login.** | | |
| **Business Rules:** |  | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### **2.2.3.17 View Room**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-17-View Room | | |
| **Created By:** | Trần Quốc An | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Party Host | **Secondary Actors:** |  |
| **Trigger:** | The Party Host can View all of the existing RoomList they created. | | |
| **Description:** | This use case allows the Party Host to view all of the existing Room List they had created | | |
| **Preconditions:** | The Party Host has access to view a list of Rooms . | | |
| **Post–conditions:** | Party Host successfully views all of the existing Rooms. | | |
| **Normal Flow:** | 1. Party Host logs into the system. 2. Party Host selects the "Room" option from the main interface. 3. Party Host selects the "List of Room" option. 4. The system displays all of the existing Rooms. | | |
| **Alternative Flows:** | If the Room list is empty, the system will notify the Party Host to create a new one. | | |
| **Exceptions:** | If it cannot connect to the server, the system reports an error and asks the user to try again later. | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **High, especially upon login.** | | |
| **Business Rules:** |  | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### **2.2.3.18 Edit Room**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-18-Edit Room | | |
| **Created By:** | Trần Quốc An | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Party Host | **Secondary Actors:** |  |
| **Trigger:** | The Party Host can Edit all of the existing Rooms they created. | | |
| **Description:** | This use case allows the Party Host to edit all of the existing Rooms they had created | | |
| **Preconditions:** | The Party Host has access to edit existing Room. | | |
| **Post–conditions:** | Party Host successfully edit the existing Room. | | |
| **Normal Flow:** | 1. Party Host logs into the system. 2. Party Host selects the "Room" option from the main interface. 3. Party Host selects the "List of Room" option. 4. The system displays all of the existing Rooms. 5. The system displays the information that Party Host can edit.. 6. The Party Host fills in all the information that they want to update. 7. Click Update to finish. | | |
| **Alternative Flows:** | If the Roomlist is empty, the system will notify the Party Host to create a new one. | | |
| **Exceptions:** | If it cannot connect to the server, the system reports an error and asks the user to try again later. | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **High, especially upon login.** | | |
| **Business Rules:** |  | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### **2.2.3.19 Buy Package**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-19-Buy Package | | |
| **Created By:** | Trần Quốc An | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Party Host | **Secondary Actors:** |  |
| **Trigger:** | The Party Host can buy the existing Package. | | |
| **Description:** | This use case allows the Party Host to buy the existing Package that the Admin had created | | |
| **Preconditions:** | The Party Host can buy an existing Package. | | |
| **Post–conditions:** | Party Host successfully buys the existing Package. | | |
| **Normal Flow:** | 1. Party Host logs into the system. 2. Party Host selects the "Upgrade to Premium" option from the main interface. 3. Party Host selects the "List of Premium Packages" option. 4. The system displays all of the existing Packages. 5. Click Buy to buy. | | |
| **Alternative Flows:** | If the Package List is empty, the system will notify the Party Host to wait for the Admin to create one. | | |
| **Exceptions:** | If it cannot connect to the server, the system reports an error and asks the user to try again later. | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **High, especially upon login.** | | |
| **Business Rules:** |  | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### **2.2.3.20 Reply Feedback**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-20-Reply Feedback | | |
| **Created By:** | Trần Quốc An | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Party Host | **Secondary Actors:** | Admin |
| **Trigger:** | The Party Host/Admin can Reply the existing Feedback created by the customer. | | |
| **Description:** | This use case allows the Party Host/Admin to buy the existing Feedback that the customer had created | | |
| **Preconditions:** | The Party Host/Admin can reply/comment on an existing Feedback. | | |
| **Post–conditions:** | Party Host/Admin successfully replied to the existing Feedback. | | |
| **Normal Flow:** | 1. Party Host/Admin logs into the system. 2. Party Host/Admin selects the "Feedback" option from the main interface. 3. The system displays all of the existing Feedback. 4. Party Host/Admin selects the Feedback they want to reply. 5. Fill in all the reply messages. 6. Click Comment to reply. | | |
| **Alternative Flows:** | If the Feedback List is empty, the system will notify the Party Host/Admin there is no Feedback. | | |
| **Exceptions:** | If it cannot connect to the server, the system reports an error and asks the user to try again later. | | |
| **Priority:** | **Low** | | |
| **Frequency of Use:** | **Low, especially upon login.** | | |
| **Business Rules:** |  | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### 

#### **2.2.3.21 View list feedback**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-21 -View list feedback | | |
| **Created By:** | Pham Phuong Nam | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Admin, Party Host |  |  |
| **Trigger:** | The Admin,Party Host wanted to view list feedback about a specific party event. | | |
| **Description:** | This use case allows the Admin, Party Host to view list feedback about all specific birthday party events. | | |
| **Preconditions:** | The system automatically recorded feedback from customers after each party event.  The system has the ability to display feedback according to the selected event.  The party event for which the admin wants to view feedback has already taken place. | | |
| **Post–conditions:** | Admin can view customer feedback to evaluate services of admin. | | |
| **Normal Flow:** | 1. Admin,Party Host log into the system.  2. Admin,Party Host select the "View list Feedback" option from the main interface.  3. The system displays a list of all Customer's previous birthday party events.  4. Admin,Party Host select a specific party event to view responses.  5. The system displays feedback details including reviews, comments and other related information. | | |
| **Alternative Flows:** | If the system has no previous party events, the system notifies and returns to the main interface. | | |
| **Exceptions:** | If it cannot connect to the database, the system reports an error and asks the user to try again later. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium, especially after each party event.** | | |
| **Business Rules:** | **BR-15:** Admin, Party Host can see all responses to party events of all customer | | |
| **Other Information:** | The system needs to provide some guidance or suggestions to help users choose the right topic. | | |
| **Assumptions:** | The admin is logged into the system and has access to view all party event's feedback information. | | |

#### **2.2.3.22 Add package**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-25 - Add Package | | |
| **Created By:** | Phạm Phương Nam | **Date Created:** | Jan-14, 2024 |
| **Primary Actor:** | Admin | **Secondary Actors:** |  |
| **Trigger:** | The Admin can create a package for the party host using. | | |
| **Description:** | This use case allows the Admin to create packages, such as information about the package. | | |
| **Preconditions:** | Admin has logged into the system  Package to be created must not exist in the system. | | |
| **Post–conditions:** | Package information is created. | | |
| **Normal Flow:** | 1. Admin logs into the system. 2. Admin selects the "Create Package" option from the main interface. 3. Admin insert ìnformation of the package to be created. 4. Admin click create button. 5. The system creates the new package. | | |
| **Alternative Flows:** | N/A | | |
| **Exceptions:** | If the package to be created has existed, the system notifies the Admin”Package already exists”. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | N/A | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### 

#### **2.2.3.23 View Package**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-23 -View Package | | |
| **Created By:** | Pham Phuong Nam | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Admin |  |  |
| **Trigger:** | The Admin wanted to view the packages that they created. | | |
| **Description:** | This use case allows the Admin to view list packages and package details. | | |
| **Preconditions:** | Admin has logged into the system  Packages to view exist in the system. | | |
| **Post–conditions:** | Package detail page is displayed. | | |
| **Normal Flow:** | 1. Admin log into the system.  2. Admin selects the "View list packages " option from the main interface.  3. The system displays a list of all packages' previous.  4. Admin selects a specific package to view.  5. The system displays package details including all information about this package. | | |
| **Alternative Flows:** | If the system has no previous package, the system notifies and returns to the main interface. | | |
| **Exceptions:** | If it cannot connect to the database, the system reports an error and asks the user to try again later. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium.** | | |
| **Business Rules:** | N/A | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### 

#### **2.2.3.24 Edit package**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-24 -Edit package | | |
| **Created By:** | Pham Phuong Nam | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Admin |  |  |
| **Trigger:** | Admin wants to update package information. | | |
| **Description:** | This use case allows the Admin to update package details, such as information of the package. | | |
| **Preconditions:** | Admin has logged into the system  Packages to be updated exist in the system. | | |
| **Post–conditions:** | Package information is updated. | | |
| **Normal Flow:** | 1. Admin logs into the system. 2. Admin selects the "Update Package" option from the main interface. 3. Admin searches and selects the package to be updated. 4. Admin makes necessary updates to the package's information. 5. Admin saves the changes, and the system updates the package's information. | | |
| **Alternative Flows:** | N/A | | |
| **Exceptions:** | If the package to be updated does not exist, the system notifies the Admin. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | N/A | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### 

#### **2.2.3.25 View info user**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-25 -View Info User | | |
| **Created By:** | Pham Phuong Nam | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Admin |  |  |
| **Trigger:** | Admin wants to view user information. | | |
| **Description:** | This use case allows the Admin to view user details, such as profile information. | | |
| **Preconditions:** | Actor has logged into the system  Users to be viewed exist in the system. | | |
| **Post–conditions:** | System displays user information. | | |
| **Normal Flow:** | 1. Admin logs into the system. 2. Admin selects the View List User option from the main interface. 3. Admin searches and selects the user to be views. 4. System displays user information. | | |
| **Alternative Flows:** | N/A | | |
| **Exceptions:** | If the user to be viewed does not exist, the system notifies the Admin. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | N/A | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### **2.2.3.26 Delete user**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-26 -Delete user | | |
| **Created By:** | Pham Phuong Nam | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Admin |  |  |
| **Trigger:** | Admin wants to delete a user. | | |
| **Description:** | This use case allows the Admin to delete a user account from the system. | | |
| **Preconditions:** | Actor has logged into the system  Users to be deleted exist in the system. | | |
| **Post–conditions:** | User account is deleted. | | |
| **Normal Flow:** | 1. Admin logs into the system. 2. Admin selects the "Delete User" option from the main interface. 3. Admin searches and selects the user account to be deleted. 4. Admin confirms the deletion action. 5. The system deletes the user account. | | |
| **Alternative Flows:** | N/A | | |
| **Exceptions:** | If the user to be deleted does not exist, the system notifies the Admin. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Low** | | |
| **Business Rules:** | N/A | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### **2.2.3.27 View Voucher**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-27 -View voucher | | |
| **Created By:** | Pham Phuong Nam | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Admin |  |  |
| **Trigger:** | The Admin wanted to view the vouchers that they created. | | |
| **Description:** | This use case allows the Admin to view list vouchers and voucher details. | | |
| **Preconditions:** | Admin has logged into the system  Vouchers exist in the system. | | |
| **Post–conditions:** | Voucher detail page is displayed. | | |
| **Normal Flow:** | 1. Admin log into the system.  2. Admin selects the "View list vouchers" option from the main interface.  3. The system displays a list of all vouchers' previous.  4. Admin selects a specific voucher to view.  5. The system displays voucher details including all information about this voucher. | | |
| **Alternative Flows:** | If the system has no previous voucher, the system notifies and returns to the main interface. | | |
| **Exceptions:** | If it cannot connect to the database, the system reports an error and asks the user to try again later. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium.** | | |
| **Business Rules:** | N/A | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### 

#### **2.2.3.28 Edit voucher**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-28 -Edit voucher | | |
| **Created By:** | Pham Phuong Nam | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Admin |  |  |
| **Trigger:** | Admin wants to update voucher information. | | |
| **Description:** | This use case allows the Admin to update voucher details, such as information about the voucher. | | |
| **Preconditions:** | Actor has logged into the system  Vouchers to be updated exist in the system. | | |
| **Post–conditions:** | Voucher information is updated. | | |
| **Normal Flow:** | 1. Admin logs into the system. 2. Admin selects the "Update Voucher" option from the main interface. 3. Admin searches and selects the voucher to be updated. 4. Admin makes necessary updates to the voucher's information. 5. Admin saves the changes, and the system updates the voucher's information. | | |
| **Alternative Flows:** | N/A | | |
| **Exceptions:** | If the voucher to be updated does not exist, the system notifies the Admin. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | N/A | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### **2.2.3.29 Add voucher**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-29 - Add Voucher | | |
| **Created By:** | Phạm Phương Nam | **Date Created:** | Jan-14, 2024 |
| **Primary Actor:** | Admin | **Secondary Actors:** |  |
| **Trigger:** | The Admin can create a voucher for the party host using to buy a package. | | |
| **Description:** | This use case allows the Admin to create vouchers, such as information about the voucher. | | |
| **Preconditions:** | Actor has logged into the system  Voucherto be created must not exist in the system. | | |
| **Post–conditions:** | Voucher information is created. | | |
| **Normal Flow:** | 1. Admin logs into the system. 2. Admin selects the "Create Voucher" option from the main interface. 3. Admin insert ìnformation of the voucher to be created. 4. Admin click create button. 5. The system creates the new voucher. | | |
| **Alternative Flows:** | N/A | | |
| **Exceptions:** | If the voucher to be created has existed, the system notifies the Admin”Voucher already exists”. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | N/A | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### 

#### **2.2.3.30 Approve Party Host**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-30 -Approve Party Host | | |
| **Created By:** | Pham Phuong Nam | **Date Created:** | Jan-22, 2024 |
| **Primary Actor:** | Admin |  |  |
| **Trigger:** | The Admin need to approve user to become a party host | | |
| **Description:** | This use case allows the Admin to review and approve party host applications submitted by users who want to host parties on the website. | | |
| **Preconditions:** | The user has submitted an application to become a party host.  The admin is logged into the system. | | |
| **Post–conditions:** | The admin approves or rejects the party host application.  The user receives a notification regarding the admin's decision. | | |
| **Normal Flow:** | 1. The user applies to become a party host, providing necessary information and agreeing to terms and conditions. 2. The admin logs into the system. 3. Admin selects the "Approve Party Host" option from the main interface. 4. The system displays a list of pending party host applications, including user details and application information. 5. Admin reviews the application details, including user profile, party hosting plans, and any additional information. 6. Admin has the option to approve or reject the party host application. 7. If approved, the user is granted party host status and access to related features. 8. If rejected, the user is informed of the decision. | | |
| **Alternative Flows:** | If there are no pending party host applications to review, display a message indicating that there are no pending applications. | | |
| **Exceptions:** | If the system encounters an error while processing the approval or rejection, it reports an error and asks the admin to try again later. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Moderate, as it depends on the number of party host applications received.** | | |
| **Business Rules:** | Admin can approve or reject party host applications based on predefined criteria. | | |
| **Other Information:** | The system may provide guidance or suggestions to help admins assess party host applications effectively. | | |
| **Assumptions:** | The admin is logged into the system.  The party host application includes essential information required for review. | | |

#### **2.2.3.31 View dashboard**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-31 -View dashboard | | |
| **Created By:** | Pham Phuong Nam | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Admin |  |  |
| **Trigger:** | The Admin logs into the system. | | |
| **Description:** | This use case allows the Admin to view the system's dashboard, providing an overview of general information about the website. | | |
| **Preconditions:** | The system automatically recorded all information of the website  The Admin logins to view dashboard | | |
| **Post–conditions:** | Admin can view the dashboard with general website information. | | |
| **Normal Flow:** | 1. Admin logs into the system. 2. Admin selects the "View Dashboard" option from the main interface. 3. The system displays the dashboard with relevant website information. | | |
| **Alternative Flows:** | If the system has not enough information, display not enough information to display dashboard | | |
| **Exceptions:** | If it cannot connect to the server, the system reports an error and asks the user to try again later. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **High, especially upon login.** | | |
| **Business Rules:** | **BR-17:** Admin can see all responses to party events of all customer | | |
| **Other Information:** | The system needs to provide some guidance or suggestions to help users choose the right topic. | | |
| **Assumptions:** | The admin is logged into the system and has access to view all party event's feedback information. | | |

#### **2.2.3.32 View revenue**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC-32- View revenue | | |
| **Created By:** | Pham Phuong Nam | **Date Created:** | Jan-13, 2024 |
| **Primary Actor:** | Admin |  |  |
| **Trigger:** | Admin wants to view revenue data. | | |
| **Description:** | This use case allows the Admin to view revenue data related to party events or other sources. | | |
| **Preconditions:** | Actor has logged into the system  The information about the party event must have been before. | | |
| **Post–conditions:** | Admin can view revenue data. | | |
| **Normal Flow:** | 1. Admin logs into the system. 2. Admin selects the "View Revenue" option from the main interface. 3. The system displays revenue data, including income, expenses, and profit. | | |
| **Alternative Flows:** | N/A | | |
| **Exceptions:** | N/A | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | N/A | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | Internet connection is available. | | |

#### 

#### 

# **3. Functional Requirements**

## **3.1 Functional Requirements of Admin**

**3.1.1. View Dashboard:**

* The admin should be able to log in to the system.
* Upon login, the admin should be redirected to a dashboard displaying relevant statistics and key performance indicators (KPIs) related to the system's activities.

**3.1.2. View List Feedback:**

* The admin should have access to a feedback management system.
* The system should present a list of feedback received from users.
* Each feedback entry should include details like date, user, and the feedback content.

**3.1.3. View Revenue:**

* The admin should be able to access a revenue tracking system.
* The system should display a summary of revenue generated, including details like date, source, and amount.

**3.1.4. Approve Party Host:**

* The admin must log in to the system.
* After successful login, the admin should have access to the "Approve Party Host" functionality.
* The "Approve Party Host" function should allow the admin to review and approve party host applications.
* The admin should have access to a list of pending party host applications.
* The admin can approve or reject each application based on predefined criteria.
* The system should notify the party host of the admin's decision.

**3.1.5. Add Package:**

* The admin must log in to the system.
* After successful login, the admin should have access to the "Add Package" functionality.
* The "Add Package" function should allow the admin to create a new package for events.
* The admin should provide details such as package name, description, pricing, and available features.
* The system must validate that all required information is provided before saving the new package.

**3.1.6. View Package:**

* The admin must log in to the system.
* After successful login, the admin should have access to the "View Package" functionality.
* The "View Package" function should display a list of all available packages in the system.
* Each package should include details such as name, description, pricing, and features.

**3.1.7. Edit Package:**

* The admin must log in to the system.
* After successful login, the admin should have access to the "Edit Package" functionality.
* The "Edit Package" function should allow the admin to modify the details of an existing package.
* The admin can update the package name, description, pricing, and features.
* The system should validate that the modified information is valid and complete.

**3.1.8. View Info User:**

* The admin must log in to the system.
* After successful login, the admin should have access to the "View Info User" functionality.
* The "View Info User" function should display user information, including their profile details.
* The admin can search for users by username or other relevant criteria.
* User information displayed should include username, contact information, and account status.

**3.1.9. Delete User:**

* The admin must log in to the system.
* After successful login, the admin should have access to the "Delete User" functionality.
* The "Delete User" function should allow the admin to deactivate or permanently delete a user account.
* The admin should provide the username or unique identifier of the user to be deleted.
* The system should confirm the deletion action to prevent accidental removal.

**3.1.10. View Voucher:**

* The admin must log in to the system.
* After successful login, the admin should have access to the "View Voucher" functionality.
* The "View Voucher" function should display a list of all available vouchers in the system.
* Each voucher should include details such as code, discount percentage, and expiration date.

**3.1.11. Edit Voucher:**

* The admin must log in to the system.
* After successful login, the admin should have access to the "Edit Voucher" functionality.
* The "Edit Voucher" function should allow the admin to modify the details of an existing voucher.
* The admin can update the voucher code, discount percentage, and expiration date.
* The system should validate that the modified voucher information is valid.

**3.1.12. Add Voucher:**

* The admin must log in to the system.
* After successful login, the admin should have access to the "Add Voucher" functionality.
* The "Add Voucher" function should allow the admin to create a new voucher.
* The admin should provide details such as voucher code, discount percentage, and expiration date.
* The system must validate that all required information is provided before saving the new voucher.

## **3.2 Functional Requirements of Customer**

**3.2.1. Manage History Order**

* **View List Orders**
* User Authentication: Users must be logged in to view their order history.
* Order Details: Display relevant details such as order ID, event date, venue, and items/services ordered.
* Filtering Options: Allow users to filter orders by date, status, or type (e.g., birthday party).
* Search Functionality: Provide a search option to help users find specific orders quickly.
* Order Status: Display the current status of each order (e.g., pending, confirmed, completed).

**Delete History Order**

* + User Confirmation: Ask for confirmation before deleting an order to prevent accidental removal.
  + Permission Check: Ensure that only the user who placed the order can delete it.
  + Archiving Mechanism: Instead of permanent deletion, consider archiving orders for record-keeping purposes.
* **3.2.2. Book Party**
* User Registration/Login: Users must have an account and be logged in to book a party.
* Party Details Form: Create a form to capture essential party details, including date, time, venue preferences, and special requests.
* Service Selection: Allow users to choose from available services, catering options, decorations, etc.
* Availability Check: Implement a system to check the availability of the selected date and time.
* Confirmation: Provide a confirmation screen before finalizing the booking, displaying a summary of the party details.
* **3.2.3. Rating**
* Post-Event Feedback: Allow users to provide ratings and feedback after the event.
* Rating Scale: Implement a rating scale (e.g., 1 to 5 stars) for overall satisfaction.
* Review Comments: Allow users to add comments to provide more detailed feedback.
* Anonymous Ratings: Optionally, allow users to submit anonymous ratings if they prefer.
* Feedback Moderation: Implement a system to moderate and filter inappropriate content in feedback.

## **3.3 Functional Requirements of Party Host**

**3.3.1. Manage Party:**

* View Party:

The party host should be able to view a list of parties they are hosting.

Each party entry should display details such as party name, date, location, and guest list.

* Delete Party:

The party host should have the ability to delete a party they are hosting.

Deleting a party should remove it from the system, and associated data (e.g., guest list) should be appropriately handled.

* Create Party:

The party host should be able to create a new party event.

Input fields should include party name, date, location, and optional details like theme or dress code.

**3.3.2. Buy Package:**

* The party host must log in to the system.
* After successful login, the admin should have access to the "Buy Package" functionality.
* The "Buy Package" function should allow the party host to buy a new package or renew it.
* The Package should provide details such as package name, description, pricing, and available features.
* After pay money, Invoice will be issued and save into party host’s Transcation

**3.3.3. Use Voucher:**

* The party host must log in to the system.
* After successful login, the admin should have access to the "View Voucher" functionality.
* The "View Voucher" function should display a list of all available vouchers in the system.
* Each voucher should include details such as code, discount percentage, and expiration date.
* Party Host can apply the voucher code to the package they buy, or click on voucher to fast buy.

**3.3.4. View Feedback:**

* The party host should be able to view feedback received from attendees.
* Each feedback entry should include details like date, attendee name, and the content of the feedback.
* Reply to Feedback:
* The party host should have the capability to reply to attendee feedback.
* Replies should be recorded and visible to the respective attendees.

**3.3.5. Contact:**

* The party host should have a contact feature to communicate with attendees or potential guests.
* The contact feature may include sending messages, announcements, or updates related to the party.

## **3.4 Functional Requirements of Guest**

**3.4.1 User Registration:**

* The system should allow guests to create a new account by providing necessary information such as name, email address, and password.
* The registration process should include validation checks to ensure data accuracy.

**3.4.2 User Login:**

* Registered guests should be able to log in to the system using their credentials (username/email and password).
* The system should provide appropriate authentication mechanisms to ensure secure access.

**3.4.3 View Blog:**

* Guests should be able to browse and view blog posts related to upcoming birthday parties, event highlights, and other relevant information.
* The blog interface should be user-friendly, allowing guests to navigate through posts easily.

**3.4.4 Search Functionality:**

* Guests should have the ability to search for specific blog posts or topics within the blog section for quick access to relevant information.

**3.4.5 Password Recovery:**

* In case a guest forgets their password, the system should provide a secure mechanism for password recovery, such as sending a reset link to the registered email address..

**3.4.6 Password Recovery:**

* In case guests forget their passwords, the system should provide a secure mechanism for password recovery, such as sending a reset link to their registered email address.

# **4. Non-Functional Requirements**

1. **Usability**

* Develop the interface so that users can easily complete tasks such as booking, payment, choose package
* Provide clear instructions and online support to help users if needed.

1. **Reliability**

* The system needs to ensure at least 99.99% availability under normal operating conditions to avoid service interruptions.
* Identify and minimize the time to fix defects once they are identified, target 6-8 hours.

1. **Performance**

* Ensure average response time is less than 3 seconds for transaction parties and does not exceed 5 seconds in special cases.
* The system needs to be able to process a minimum of 1,000 transaction sets per second to meet growth needs and ensure operational flexibility.

1. **Security**

● Personal user data must be encrypted and securely protected according to industry security standards.

1. **Compatibility**

* Ensures compatibility with popular browsers such as Chrome, Firefox and Safari to support cross-platform use.

# **5. Requirement Appendix**

## **5.1 Business Rules**

|  |  |
| --- | --- |
| **ID** | **Rule Definition** |
| BR-01 | The system enforces secure password policies. |
| BR-02 | Accounts may be temporarily locked after a certain number of unsuccessful login attempts |
| BR-03 | Blogs are periodically updated or added to the system. |
| BR-04 | The system enforces secure password policies during registration. |
| BR-05 | Each username and email must be unique. |
| BR-06 | The customer can only view orders that are associated with their account. |
| BR-07 | The order list should be displayed in a user-friendly format with relevant details. |
| BR-08 | A minimum advance notice of two weeks is required for booking |
| BR-09 | A history order with the topic of booking a birthday party can only be deleted by the customer who placed the order. |
| BR-10 | Ratings should be on a predefined scale or set of options. |
| BR-11 | User profiles are regularly updated with the latest information |
| BR-12 | The system enforces secure password policies during the change process. |
| BR-13 | User profiles are regularly updated with the latest information. |
| BR-14 | The system clears sensitive session data during the log out process to enhance security. |
| BR-15 | Admin, Party Host can see all responses to party events of all customer |
| BR-16 | Admin can see all responses to party events of all customer |
| BR-17 | Admin can see all responses to party events of all customer |