**FPT UNIVERSITY**

Booking Birthday For Kids

Software Requirement Specification

**Project Code**: SWR302

**Document Code**: SRS-SWR302– v1.0

| **SE1702 - GR6** | |
| --- | --- |
| **Group Members** | Tạ Ngọc An - Leader - QE170035  Nguyễn Trần Anh Quốc - Member - SE161132  Nguyễn Hữu Tín - Member - SE160859  Lê Nguyễn Minh Huy - Member - SE161054 |
| **Project** | **BIRTHDAY PARTY BOOKING FOR KIDS** |

- HCM, January 2024 -

**RECORD OF CHANGE**

\*A - Added M - Modified D - Deleted

| Effective Date | Changed Items | A\*  M, D | Change Description | New Version |
| --- | --- | --- | --- | --- |
| 27/01/2024 | Use Case: Sign up | ADD | Add Sign Up Use Case Detail |  |
| 27/01/2024 | Use Case: Login | ADD | Add Login Use Case Detail |  |
| 27/01/2024 | Use Case: Logout | ADD | Add Logout Use Case Detail |  |
| 27/01/2024 | Use Case: Update personal information | ADD | Add Update personal information Use Case Detail |  |
| 27/01/2024 | Use Case: View personal information | ADD | Add View personal information Use Case Detail |  |
| 27/01/2024 | Use Case: Forgot password | ADD | Add Forgot password Use Case Detail |  |
| 27/01/2024 | Use Case: Change password | ADD | Add Change password Use Case Detail |  |
| 27/01/2024 | Use Case: View comment/rating | ADD | Add View comment/rating Use Case Detail |  |
| 27/01/2024 | Use Case: Make a booking | ADD | Add Make a booking Use Case Detail |  |
| 27/01/2024 | Use Case: Add a party to the cart | ADD | Add Add a party to the cart Use Case Detail |  |
| 27/01/2024 | Use Case: Change quality | ADD | Add Change quality Use Case Detail |  |
| 27/01/2024 | Use Case: Check out | ADD | Add Check out Use Case Detail |  |
| 27/01/2024 | Use Case: Apply voucher | ADD | Add Apply voucher Use Case Detail |  |
| 27/01/2024 | Use Case: Cancel a booking | ADD | Add Cancel a booking Use Case Detail |  |
| 27/01/2024 | Use Case: View booking details | ADD | Add View booking details Use Case Detail |  |
| 27/01/2024 | Use Case: View all booking | ADD | Add View all booking Use Case Detail |  |
| 27/01/2024 | Use Case: Send comment/rating | ADD | Add Send comment/rating Use Case Detail |  |
| 27/01/2024 | Use Case: View menu | ADD | Add View menu Use Case Detail |  |
| 27/01/2024 | Use Case: View birthday party information | ADD | Add View birthday party information Use Case Detail |  |
| 27/01/2024 | Use Case: Search a party | ADD | Add Search a party Use Case Detail |  |
| 27/01/2024 | Use Case: View host party information | ADD | Add View host party information Use Case Detail |  |
| 27/01/2024 | Use Case: View all post of host party | ADD | Add View all post of host party Use Case Detail |  |
| 27/01/2024 | Use Case: View party order information | ADD | Add View party order information Use Case Detail |  |
| 27/01/2024 | Use Case: Confirm party orders | ADD | Add Confirm party orders Use Case Detail |  |
| 27/01/2024 | Use Case: Complete party orders | ADD | Add Complete party orders Use Case Detail |  |
| 27/01/2024 | Use Case: Cancel party order | ADD | Add Complete party orders Use Case Detail |  |
| 27/01/2024 | Use Case: Create birthday party | ADD | Add Create birthday party Use Case Detail |  |
| 27/01/2024 | Use Case: View birthday party list | ADD | Add View birthday party list Use Case Detail |  |
| 27/01/2024 | Use Case: Update birthday party | ADD | Add Update birthday party Use Case Detail |  |
| 27/01/2024 | Use Case: Delete birthday party | ADD | Add Delete birthday party Use Case Detail |  |
| 27/01/2024 | Use Case: Disable package | ADD | Add Disable package Use Case Detail |  |
| 27/01/2024 | Use Case: Enable package | ADD | Add Enable package Use Case Detail |  |
| 27/01/2024 | Use Case: Create package | ADD | Add Create package Use Case Detail |  |
| 27/01/2024 | Use Case: Update package | ADD | Add Update package Use Case Detail |  |
| 27/01/2024 | Use Case: Delete package | ADD | Add Delete package Use Case Detail |  |
| 27/01/2024 | Use Case: Review package/party content | ADD | Add Review package/party content Use Case Detail |  |
| 27/01/2024 | Use Case: Enable customer account | ADD | Add Enable customer account Use Case Detail |  |
| 27/01/2024 | Use Case: Disable customer account | ADD | Add Disable customer account Use Case Detail |  |
| 27/01/2024 | Use Case: Delete comment | ADD | Add Delete comment Use Case Detail |  |
| 27/01/2024 | Use Case: Create a post | ADD | Add Create a post Use Case Detail |  |
| 27/01/2024 | Use Case: View list post | ADD | Add View list post Use Case Detail |  |
| 27/01/2024 | Use Case: View transaction | ADD | Add View transaction Use Case Detail |  |
| 27/01/2024 | Use Case: View booking | ADD | Add View booking Use Case Detail |  |
| 27/01/2024 | Use Case: View category | ADD | Add View category Use Case Detail |  |
| 27/01/2024 | Use Case: Add category | ADD | Add Add category Use Case Detail |  |
| 27/01/2024 | Use Case: Enable category | ADD | Add Enable category Use Case Detail |  |
| 27/01/2024 | Use Case: Disable category | ADD | Add Disable category Use Case Detail |  |
| 27/01/2024 | Use Case: Edit category | ADD | Add Edit category Use Case Detail |  |
| 27/01/2024 | Use Case: Update a post | ADD | Add Update a post Use Case Detail |  |
| 27/01/2024 | Use Case: Delete a post | ADD | Add Delete a post Use Case Detail |  |

**SIGNATURE PAGE**

**ORIGINATOR:** <Name> <Date>

<Position>

**REVIEWERS:** <Name> <Date>

<Position>

<Name, if it’s needed> <Date>

<Position>

**APPROVAL:** <Name> <Date>

<Position>

**TABLE OF CONTENTS**

1 Introduction.. 5

1.1 Purpose. 5

1.2 Scope. 5

1.3 Definitions, Acronyms, and Abbreviations. 5

1.4 References. 5

1.5 Overview.. 6

2 Overall Description.. 7

3 FUNCTIONAL Requirements. 8

3.1 <Use Cases Diagram>. 8

3.2 < Use Case Name 1>. 8

3.3 < Use Case Name 2>. 9

4 NON-FUNCTIONAL Requirements. 10

4.1 Usability. 10

4.2 Reliability. 10

4.3 Performance. 10

4.4 Supportability. 11

4.5 Design Constraints. 11

4.6 On-line User Documentation and Help System Requirements 11

4.7 Purchased Components. 11

4.8 Interfaces. 12

4.9 Licensing Requirements 12

4.10 Legal, Copyright, and Other Notices. 12

4.11 Applicable Standards. 12

5 Supporting Information.. 13

# **1** **Introduction**

## **1.1** **Purpose**

The purpose of this document is to present a detailed description of the birthday party booking website. It will explain the purpose and characteristics of the system, its interfaces, what the system will do, the constraints under which it must operate, and how the system will respond to external stimuli. This document is intended for both stakeholders and system developers.

## **1.2** **Scope**

This document outlines the parameters of the birthday party booking website, focusing on its features, interfaces, constraints, and associated use-cases. It covers the essential functionalities such as party package selection, booking management, payment processing, and user account administration. The scope also includes descriptions of user interfaces, constraints (technical, regulatory, and business), dependencies on external systems, and any assumptions made during the development process.

## **1.3** **Definitions, Acronyms, and Abbreviations**

The following definitions, acronyms, and abbreviations are used throughout this document:

* SRS: Software Requirements Specification
* UI: User Interface
* API: Application Programming Interface
* DBMS: Database Management System
* GUI: Graphical User Interface
* QA: Quality Assurance
* UAT: User Acceptance Testing
* HTTP: Hypertext Transfer Protocol
* HTML: Hypertext Markup Language
* CSS: Cascading Style Sheets
* UX: User Experience
* RAM: Random Access Memory
* CPU: Central Processing Unit

These definitions will aid in understanding the requirements and specifications outlined in this document.

## **1.4** **References**

[Software\_Requirment\_Specification\_Template\_2016 (2).docx](https://lms-hcmuni.fpt.edu.vn/pluginfile.php/9593/mod_assign/introattachment/0/Software_Requirment_Specification_Template_2016%20%282%29.docx?forcedownload=1) : Template

## **1.5** **Overview**

This document is written based on the standards of the Software Requirements Specification Document.

With a structure divided into three parts:

* Part 1: Provides an overview of the components of SRS.
* Part 2: Overview of factors, constraints, user characteristics, and execution environment that impact the system and its requirements.
* Part 3: Provides detailed information on functional requirements, providing software developers with information to develop software that meets those requirements.
* Part 4: The Non-functional Requirements section of this document is written to describe in detail the non-functionality of the product, defining how a system should look like.
* Part 5: The Supporting Information section of this document provides the supporting information that makes the SRS easier to use.

# **2** **Overall Description**

## **2.1** **Product Overview**

### **2.1.1 Guest Requirements**

Guests are people who have not been authenticated to the system. They only have limited access the following functions:

* View menu
* Search a party
* View host party information
* View all post of host party
* View birthday party information
* View comment/rating

### **2.1.2 Registered User Requirements**

Registered Users are people who have registered into the HappyKid website. They can use the following functions:

* Login
* Logout
* Profile management
* Forgot password
* Change password
* View comment/rating

### **2.1.3 Customer Requirements**

Customers are people who have registered into the HappyKid website with the “Customer” role. They can use the following functions:

* Make a booking
* Cancel a booking
* View booking details
* View all booking
* Send comment/rating

### **2.1.4 Party Host Requirements**

Party Hosts are people who have registered into the HappyKid website with the “Party Host” role. They can use the following functions:

* Manage birthday party booking
* Manage birthday party information
* Manage package

### **2.1.4 Administrator Requirements**

Admins are people who have registered into the HappyKid website with the “Administrator” role. They can use the following functions:

* Review package/party content
* Manage user/party host accounts
* Delete comment
* Manage post
* View transaction
* View booking
* Category

## **2.2** **User Characteristics**

* Guest : person who just view services but not choose to services to book
* Customer : person who can choose services to book
* Party Host : person who accept and organization a party
* Administrator : person who can manager system

## **2.3** **Constraints**

* Technical Constraints: These are constraints related to technology, hardware, software, or environment in which the software must operate. For example, a mobile application may have constraints regarding compatibility with older versions of the mobile operating system.
* Time Constraints: Requirements related to completion time, response time, or turnaround time that the software must adhere to. For example, a financial transaction system may have constraints on transaction processing time.
* Resource Constraints: These are constraints related to resources such as memory, network bandwidth, or storage capacity available to the software. For example, a web application may have constraints on server resources it can utilize.
* Budgetary Constraints: Constraints related to budget or finances that the project must adhere to. For example, a software project may have constraints on development budget.
* Security or Privacy Constraints: Requirements related to security or privacy that the software must adhere to. For example, a healthcare system may have constraints on personal information security.

## **2.4** **Assumptions And Dependencies**

Assumptions:

* Users have stable internet connections and access to modern web browsers for optimal website functionality.
* Users are familiar with basic website navigation and online booking procedures.
* The HappyKid website operates in a secure environment with measures in place to protect user data and transactions.
* Customer information provided during registration and booking is accurate and up-to-date.
* Party Hosts are responsible for ensuring the accuracy of the party details and availability of requested services.
* Administrators have appropriate permissions and training to manage system functions effectively.
* The website's features and functions meet the legal and regulatory requirements of the regions it operates in.

Dependencies:

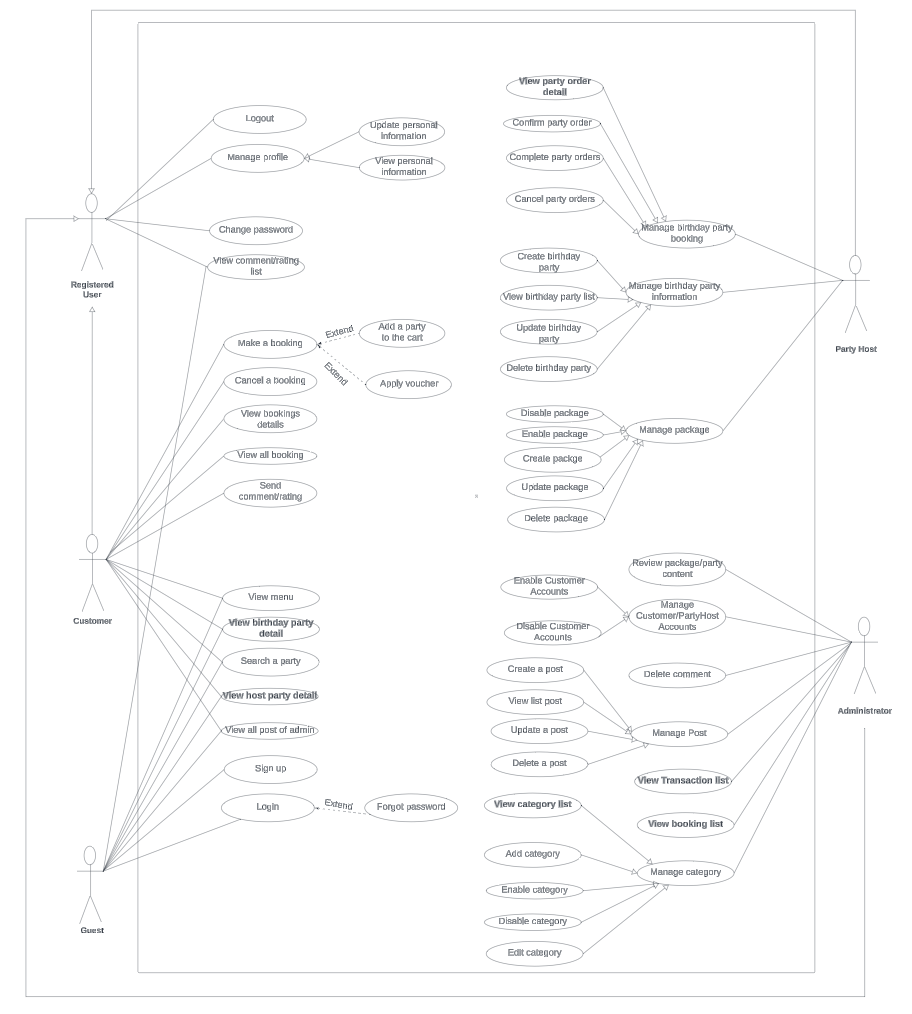
* The functionality of the website relies on stable server infrastructure and backend systems to handle user requests, data storage, and processing.
* Integration with payment gateways is necessary to facilitate secure online transactions for bookings and purchases.
* Proper communication channels need to be established for notifications regarding booking confirmations, cancellations, and updates to users.
* Regular maintenance and updates are required to ensure the website remains compatible with evolving web technologies and security standards.
* The availability of customer support resources is essential for addressing user inquiries, technical issues, and resolving disputes effectively.
* Compliance with relevant privacy laws and regulations dictates how user data is collected, stored, and managed within the system.

## **2.5** **Use Case**

| ID | UseCase | Primary Actor | Secondary Actor |
| --- | --- | --- | --- |
| UC001 | Sign up | Guest |  |
| UC002 | Login | Guest |  |
| UC003 | Logout | Registered User |  |
| UC004 | Update personal information | Registered User |  |
| UC005 | View personal information | Registered User |  |
| UC006 | Forgot password | Registered User |  |
| UC007 | Change password | Registered User |  |
| UC008 | View comment/rating | Registered User, Guest |  |
| UC009 | Make a booking | Customer |  |
| UC010 | Add a party to the cart | Customer |  |
| UC011 | Apply voucher | Customer |  |
| UC012 | Cancel a booking | Customer |  |
| UC013 | View booking details | Customer |  |
| UC014 | View all booking | Customer |  |
| UC015 | Send comment/rating | Customer |  |
| UC016 | View menu | Customer, Guest |  |
| UC017 | View birthday party information | Customer, Guest |  |
| UC018 | Search a party | Customer, Guest |  |
| UC019 | View host party information | Customer, Guest |  |
| UC020 | View all post of admin | Customer, Guest |  |
| UC021 | View party order information | Party Host |  |
| UC022 | Confirm party orders | Party Host |  |
| UC023 | Complete party orders | Party Host |  |
| UC024 | Cancel party order | Party Host |  |
| UC025 | Create birthday party | Party Host |  |
| UC026 | View birthday party list | Party Host |  |
| UC027 | Update birthday party | Party Host |  |
| UC028 | Delete birthday party | Party Host |  |
| UC029 | Disable package | Party Host |  |
| UC030 | Enable package | Party Host |  |
| UC031 | Create package | Party Host |  |
| UC032 | Update package | Party Host |  |
| UC033 | Delete package | Party Host |  |
| UC034 | Review package/party content | Administrator |  |
| UC035 | Enable customer account | Administrator |  |
| UC036 | Disable customer account | Administrator |  |
| UC037 | Delete comment | Administrator |  |
| UC038 | Create a post | Administrator |  |
| UC039 | View list post | Administrator |  |
| UC040 | View transaction | Administrator |  |
| UC041 | View booking | Administrator |  |
| UC042 | View category | Administrator |  |
| UC043 | Add category | Administrator |  |
| UC044 | Enable category | Administrator |  |
| UC045 | Disable category | Administrator |  |
| UC046 | Edit category | Administrator |  |
| UC047 | Update a post | Administrator |  |
| UC048 | Delete a post | Administrator |  |

# **3** **FUNCTIONAL Requirements**

## **3.1** **Use Cases Diagram**



## **3.2** **Use Case Description**

### **3.2.1 Sign up**

| **USE CASE-01 Sign up** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC001 | **Use-case Version** | <1.0> |
| **Use-case Name** | Sign up | | |
| **Author** | Ta Ngoc An | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Guest  **Summary:**  This "Use case" allows guests to sign up for an account.  **Goal:**  The system creates an account with a username and password is provided by the guest.  **Triggers**  The actor wants to create a new account for login  **Preconditions:**  PRE-1: Active web or mobile app.  PRE-2: The actor has reached the registration page.  PRE-3: The actor provides required information to register.  **Post Conditions:**  POST-1: The actor has successfully registered and has a new account in the system.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | In the sign-in page, the actor clicks on the “Sign up” button.  [Alternative 1] | System redirects the actor to login page and shows the login form which has:   * Complete name: A string can’t be identical. * Username: A string can’t be identical. * Password: A string contains at least 8 characters. * Register button. | | **2** | The actor fills all attributes in the form. | [Exception 1 2 3 4] | | **3** | The actor clicks the “Register” button. | System checks input information:   * Valid: system create new account and redirects to login page * Invalid: System shows error message according to wrong filled attribute. |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor chooses to sign up via Google. | System shows the list of Google accounts of actors. | | **2** | The actor chooses the account to sign up for. | System checks input information:   * Valid: system create new account and redirects to login page * Invalid: System shows error message according to wrong filled attribute. |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | Username has already existed | System requests the actor to enter the password in the right form | | **2** | Invalid password | System requests the actor to enter password in the right format | | **3** | Confirm password does not match password | System informs that the confirm password must match the password | | **4** | Phone number has been assigned to another account | System informs that the phone number has been used and requires another phone number |   **Relationships:**  N/A  **Business Rules:**  1. Username does not exist before  2. Password must not be empty and contains at least 8 characters  3. Password and confirm password must match to each other  4. Phone number has not been assigned to any account before. | | | |

### **3.2.2 Login**

| **USE CASE-02 Login** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC002 | **Use-case Version** | <1.0> |
| **Use-case Name** | Login | | |
| **Author** | Ta Ngoc An | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Guest  **Summary:**  This "Use case" allows actors to log into the system.  **Goal:**  The system checks username and password then allows the actor to login.  **Triggers**  The actor wants to log in to the system.  **Preconditions:**  PRE-1: The actor's registered account before log in.  PRE-2: The actor must provide valid username and password, or Gmail, Facebook what is registered.  PRE-3: The device must be connected to the internet.  **Post Conditions:**  POST-1: Successful Authentication: Grants user account access with valid credentials.  POST-2: Error Handling: Displays error messages for login issues.  POST-3: User State: Maintains logged-in status for account actions.  POST-4: Security Measures: Ensures ongoing data security.  POST-5: Access Control: Provides access to account features and data.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | In the sign-in page, the actor clicks on the “Login” button.  [Alternative 1] | System redirects the actor to login page and shows the login form which has:   * Username: A string can’t be identical. * Password: A string contains at least 8 characters. * Login button. | | **2** | The actor fills all attributes in the form. | [Exception 1] | | **3** | The actor clicks the “Login” button. | The system verifies the credentials and authenticates the actor:   * Valid: System logs the actor into account and redirects to the homepage. * Invalid: System shows error message “Incorrect username or password”. |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor chooses to login via Google. | System shows the list of Google accounts of the actor. | | **2** | The actor chooses the account to login. | System logs the actor into account and redirects to homepage |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | The actor enters the wrong username or password. | System responses with a message “Incorrect username or password” |   **Relationships:**  N/A  **Business Rules:**  1. If the actor enters the wrong login information more than 3 times, the account will be locked. | | | |

### **3.2.3 Logout**

| **USE CASE-03 Logout** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC003 | **Use-case Version** | <1.0> |
| **Use-case Name** | Logout | | |
| **Author** | Ta Ngoc An | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Registered User  **Summary:**  This “use case” allows actors to log out of the system.  **Goal:**  The system ends the actor's session and ensures their personal information is not accessed after logging out.  **Triggers**  The actor wants to log out of the system  **Preconditions:**  PRE-1: The actor is logged into the system.  PRE-2: The device must be connected to the internet.  **Post Conditions:**  POST-1: The actor is logged out of their current session.  POST-2: The system redirects to the home page.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | On the homepage, the actor hovers over the user name displayed on the navigation bar. | A dropdown list is is shown to the actor, which contains:   * Profile * Activities * Notification * Logout | | **2** | The actor clicks “Logout” in the dropdown list. | The system verifies the user's credentials and session. System exits the actor from the current account and redirects to the homepage.  [Exception 1] |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | The actor logs out without saving changes. | The system verifies the credentials and authenticates the actor:   * Logout: System exits the actor from the current account and redirects to the homepage. * Cancel: System cancels the logout process and returns to the previous page. |   **Relationships:**  N/A  **Business Rules:**  1. Ensure that actors no longer have access to pages that require a login after logging out. | | | |

### **3.2.4 Update personal information**

| **USE CASE-04 Update personal information** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC004 | **Use-case Version** | <1.0> |
| **Use-case Name** | Update personal information | | |
| **Author** | Ta Ngoc An | | |
| **Date** | 30/01/2024 | **Priority** | Normal |
| **Actor:**  Registered User  **Summary:**  This "Use case" allows actors to update personal information.  **Goal:**  The system updates user information, including information such as name, address, email, phone number,...  **Triggers**  The actor indicates that they want to update personal information.  **Preconditions:**  PRE-1: The actor is logged into the system.  PRE-2: The device must be connected to the internet.  **Post Conditions:**  POST-1: Update Successful: The user information is saved by system.  POST-2: The user's profile reflects the changes.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | On the homepage, the actor hovers over the user name displayed on the navigation bar. | A dropdown list is is shown to the actor, which contains:   * Profile * Activities * Notification * Logout | | **2** | The actor clicks “Profile” in the dropdown list. | The system redirects to the actor Profile page displaying the options:   * Update * View details * Change Password | | **3** | In the actor Profile page, the actor clicks on the “Update” button. | System show all fields for the actor to fill, including:   * Name: text * Phone: number text input, required, length 9 – 11 * Gender: text * Address: text * Birthday: date * Avatar: image | | **4** | The actor fills all attributes in the form. | [Exception 1] | | **5** | The actor clicks on the “Update” button. | System shows new information to the actor after update.  [Exception 2] |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | At least one input field doesn’t match the constraint. | System shows an error message corresponding to the fail constraint. | | **2** | The request took too long to process due to heavy load or system error. | Notify the error and ask the actor to try again later, or automatically try again after a period of time. |   **Relationships:**  N/A  **Business Rules:**  1. The new password must meet strength standards, such as length, upper and lower case letters, numbers and special characters.  2. New personal information must not overlap with old information. | | | |

### **3.2.5 View personal information**

| **USE CASE-05 View personal information** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC005 | **Use-case Version** | <1.0> |
| **Use-case Name** | View personal information | | |
| **Author** | Ta Ngoc An | | |
| **Date** | 30/01/2024 | **Priority** | Normal |
| **Actor:**  Registered User  **Summary:**  This "Use case" allows actors to view personal information.  **Goal:**  The system allows the actor to view information, including information such as name, address, email, phone number,...  **Triggers**  The actor indicates that they want to view personal information.  **Preconditions:**  PRE-1: The actor is logged into the system.  PRE-2: The device must be connected to the internet.  **Post Conditions:**  POST-1: The actor data does not change and is stored permanently.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | On the homepage, the actor hovers over the user name displayed on the navigation bar. | A dropdown list is is shown to the actor, which contains:   * Profile * Activities * Notification * Logout | | **2** | The actor clicks “Profile” in the dropdown list. | The system redirects to the actor Profile page displaying the options:   * Update * View details * Change Password | | **3** | The actor views information by clicking the "View details" button. | Page show information of customer selected include:   * Id: number * Name: text * Email: text * Password: text * Phone: number * Gender: text * Address: text * Birthday: date * Avatar: image |   **Alternative Scenario:**  N/A  **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  1. The actor can only view profiles during the current session and need to re-authenticate after a period of time. | | | |

### **3.2.6 Forgot password**

| **USE CASE-06 Forgot password** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC006 | **Use-case Version** | <1.0> |
| **Use-case Name** | Forgot password | | |
| **Author** | Ta Ngoc An | | |
| **Date** | 1/02/2024 | **Priority** | High |
| **Actor:**  Registered User  **Summary:**  This "use case" allows actors to reset their password when they forget their password.  **Goal:**  Allows actors to reset their password if they have forgotten it, providing a secure and actor-friendly way for actors to regain access to their accounts.  **Triggers**  When a actor indicates they have forgotten their password and wish to initiate the password reset process.  **Preconditions:**  PRE-1: The actor has registered an account before.  PRE-2: The actor must remember other security information.  PRE-3: The device must be connected to the internet.  **Post Conditions:**  POST-1: The actor’s password is successfully reset.  POST-2: The actor is provided with a new password.  POST-3: The actor is redirected to a login.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | In the sign-in page, the actor clicks on the “Forgot Password” button.  [Alternative 1] | The system redirects the actor to the forgot password page and displays a form with:   * Email: A string cannot be identical. * Search button. | | **2** | The actor fills all attributes in the form. | [Exception 1] | | **3** | The actor clicks the “Search” button. | Actor information verification system:   * Valid: The system sends a password reset link to the actor's registered email address. * Invalid: The system displays the error message "Email does not exist or is incorrect".   [Exception 2] | | **4** | The actor clicks on the password reset link. | The system redirects the actor to the Reset Password page and displays a form with:   * New password: A string cannot be identical. * Re-enter new password: A string cannot be identical.   [Exception 3] | | **5** | The actor fills all attributes in the form. | [Exception 1] | | **6** | The actor clicks the “Submit” button. | The system updates the user's password in the database. And the actor is redirected to the login page. |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor chooses to authenticate via phone number. | The system redirects the actor to the forgot password page and displays a form with:   * Phone Number * Search button. | | **2** | The actor fills all attributes in the form. | [Exception 1] | | **3** | The actor clicks the “Search” button. | Actor information verification system:   * Valid: The system sends a password reset link to the actor's registered phone number. * Invalid: The system displays the error message "Phone number does not exist or is incorrect". | | **4** | The actor clicks on the password reset link. | The system redirects the actor to the Reset Password page and displays a form with:   * New password: A string cannot be identical. * Re-enter new password: A string cannot be identical.   [Exception 3] | | **5** | The actor fills all attributes in the form. | [Exception 1] | | **6** | The actor clicks the “Submit” button. | The system updates the actor's password in the database. And the actor is redirected to the login page. |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | At least one input field doesn’t match the constraint. | System shows error message corresponding to the fail constrain | | **2** | Unable to send email. | The system will display an error message: Unable to send email. And provide instructions on other ways to resolve the problem. | | **3** | Password reset link expired. | The system notifies that the link has expired, and asks the actor to try again. |   **Relationships:**  N/A  **Business Rules:**  1. The actor can only enter the wrong email or username no more than 5 times.  2. The password reset email is only valid for 5 minutes. | | | |

### **3.2.7 Change password**

| **USE CASE-07 Change password** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC007 | **Use-case Version** | <1.0> |
| **Use-case Name** | Change password | | |
| **Author** | Ta Ngoc An | | |
| **Date** | 1/02/2024 | **Priority** | High |
| **Actor:**  Registered User  **Summary:**  This "use case" allows actors to change their password when they indicate that they want to change their password.  **Goal:**  Allow actors to change their existing passwords with new ones and ensure actors have control over their login information.  **Triggers**  The actor indicates that they want to change their password.  **Preconditions:**  PRE-1: The actor is logged into the system.  PRE-2: The device must be connected to the internet.  PRE-3: The system authenticated and presented actors with their personal information, including a form to change their password.  **Post Conditions:**  POST-1: The actor’s password is successfully changed.  POST-2: The actor is notified of the successful password change.  POST-3: Any active sessions with the old password are invalidated  POST-4: The actor is redirected to a login.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | On the homepage, the actor hovers over the actor name displayed on the navigation bar. | A dropdown list is is shown to the actor, which contains:   * Profile * Activities * Notification * Logout | | **2** | The actor clicks “Profile” in the dropdown list. | The system redirects to the actor Profile page displaying the options:   * Update * View details * Change Password | | **3** | In the actor Profile page, the actor clicks on the “Change Password” button. | The system redirects the actor to the Change Password page and displays a form with:   * New password: A string cannot be identical. * Re-enter new password: A string cannot be identical. | | **4** | The actor fills all attributes in the form. | [Exception 1] | | **5** | The actor clicks the “Submit” button. | The system updates the actor's password in the database. And the actor is redirected to the login page. |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | Old and new passwords are the same. | The system will display an error message: “The old and new passwords are the same”. And ask the actor to re-enter the new password. |   **Relationships:**  N/A  **Business Rules:**  1. New passwords need to meet strength requirements, such as minimum length, character diversity, numbers, and special characters.  2. The old password and new password must be different. | | | |

### **3.2.8 View comment/rating list**

| **USE CASE-08 View comment/rating list** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC008 | **Use-case Version** | <1.0> |
| **Use-case Name** | View comment/rating | | |
| **Author** | Ta Ngoc An | | |
| **Date** | 1/02/2024 | **Priority** | Normal |
| **Actor:**  Registered User, Guest  **Summary:**  This "use case" allows actors to view comments made by themselves or others about a party that has been booked.  **Goal:**  Allows actors to view comments and reviews list about previously booked parties. This review and rating was created by another actor who booked the party.  **Triggers**  The actor indicates that they want to see reviews and ratings of parties that have already been booked.  **Preconditions:**  PRE-1: The actor is logged into the system.  PRE-2: The device must be connected to the internet.  PRE-3: Party options must have at least 1 comment or rating to be viewable.  **Post Conditions:**  POST-1: Comments and rating of the party are displayed.  POST-2: The actor can view comments and ratings list.  POST-3: Comments and reviews hidden by administrators will not be visible.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | On the homepage, the actor clicks on the pre-created party options, the party for which they want to see comments and ratings. | System redirects the customer to the party detail page and show the information include:   * Add to cart button * Book now button * Party option buttons | | **2** | In the party details page, actors scroll down to the rating section. | The system displays a list of ratings and comments.  [Exception 1] |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | No Rating available. | The system will display a message “No rating and comments to show”. |   **Relationships:**  N/A  **Business Rules:**  1. The system must display reviews and comments in order from newest to oldest. | | | |

### **3.2.9 Make a booking**

| **USE CASE-09 Make a booking** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC009 | **Use-case Version** | <1.0> |
| **Use-case Name** | Make a booking | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Customer  **Summary:**  This use case describes the process by which The actor makes a booking for a birthday party for kids within the system  **Goal:**  Enable The actor to easily book birthday parties for their children  **Triggers**  The actor decides to make a booking for a birthday party.  **Preconditions:**  PRE-1: The actor must be logged in to the system.  PRE-2: The desired party package must be available for booking.  **Post Conditions:**  The booking is successfully made and recorded in the system..  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | On the homepage, the customer clicks on the pre-created party options, the party for which they want to book. | System redirects the customer to the party detail page and show the information include:   * Add to cart button * Book now button * Party option buttons | | **2** | The actor clicks the “Book now” button. | Redirects to the detailed information entry page and show a table to fill in personal information form including:   * Name: Text * Phone: Must be in format (+84) xxx-xxx-xxxx * Email: Must be in the format abc@gmail.com * Address: Text * Date: Must be in the format DD/MM/YYYY (e.g., 31/03/2024). * Time: Must be in a valid time format (e.g., 12:00 PM). * Venue: Must be a valid venue option available in the system. * Guest Count: Must be a positive integer indicating the number of guests attending the party. * Any Special Requests: Free-form text field allowing the party host to provide additional details or requests for the party event. * “Submit” button | | **4** | The actor fills all attributes in the form. | [Exception 1] | | **5** | The actor clicks the “Submit” button. | System checks input information:   * Valid: the system creates a new order and redirects to the view booking detail. * Invalid: The system displays an error message due to incorrectly filling in the attribute. |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | The actor enters information that does not match the constraints. | System shows an error message corresponding to the fail constraint. |   **Relationships:**  N/A  **Business Rules:**   * The actor must be login account * The actor must deposit in advance | | | |

### **3.2.10 Add a party to the cart**

| **USE CASE-10 Add a party to the cart** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC010 | **Use-case Version** | <1.0> |
| **Use-case Name** | Add a party to the cart | | |
| **Author** | Ta Ngoc An | | |
| **Date** | 1/02/2024 | **Priority** | High |
| **Actor:**  Customer  **Summary:**  This "use case" allows an actor to add a party to cart.  **Goal:**  Allow actors to add a party to cart, so they can view, manage, save parties that they want to book.  **Triggers**  The actor indicates they want to add a party to the cart.  **Preconditions:**  PRE-1: The actor is logged into the system.  PRE-2: The system is running.  PRE-3: The device must be connected to the internet.  PRE-4: The party that the actor wants to add is displayed and available for purchase.  **Post Conditions:**  POST-1: The party is added to the actor’s cart.  POST-2: The system stores party information in the actor's shopping cart.  POST-3: The actor can manage, view, add, and remove parties from the shopping cart.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | On the homepage, the actor clicks on the pre-created party options, the party for which they want to add to cart. | System redirects the actor to the party detail page and show the information include:   * Add to cart button. * Book now button. * Party option buttons | | **2** | In the party details page, the actor clicks the “Add to cart” button. | The system adds a party to the actor’s shopping cart.  [Exception 1]  [Exception 2] |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | Shopping cart is full. | The system notifies the actor and requires them to edit the shopping cart before adding a new one. | | **2** | Party temporarily disabled. | The system reports errors and provides detailed information to actors. |   **Relationships:**  N/A  **Business Rules:**  1. The actor can only add to cart if they are logged into their account.  2. The system must check and update inventory quantity before adding to cart to ensure party availability. | | | |

### **3.2.11 Apply Voucher**

| **USE CASE-011 Apply voucher** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC011 | **Use-case Version** | <1.0> |
| **Use-case Name** | Apply voucher | | |
| **Author** | Tín | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Customer  **Summary:**  The user, acting as a hoster, is granted access to the system to search for information about birthday parties, aiming to efficiently manage party-related information and interactions within the system.  **Goal:**  The goal is to enable the hoster to apply vouchers to party bookings, providing discounts or special offers to patrons.  **Triggers**  The hoster initiates the process to apply a voucher during the booking or checkout process.  **Preconditions:**   * The hoster must be logged into the system to access party management functionalities. * The application must provide access to voucher management and booking functionalities for the hoster.   **Post Conditions:**  After applying a voucher to a booking, the system updates the booking details to reflect the discounted price or special offer.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | In the product payment interface, the customer clicks the "Apply voucher" button | The system displays vouchers that customers currently have and can use. | | **2** | Customers select the voucher they want to use.  [Alternative 1] | The system checks the selected voucher, and applies the new price to the booking.  [Exception 2] |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The customer clicks the “Other” button. | The system displays a form to enter discount codes including:   * Voucher code: Text * Confirm button | | **4** | The customer fills all attributes in the form. | [Exception 1] | | **2** | The customer clicks “Confirm” button | The system checks the selected voucher, and applies the new price to the booking.  [Exception 2] |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | Invalid voucher code entered by the hoster. | The system notifies the hoster of the invalid voucher code and prompts them to enter a valid code. | | **2** | Voucher has expired. | The system displays a message: This voucher is currently sold out, please choose another voucher. |   **Relationships:**  N/A  **Business Rules:**   * The hoster must have appropriate permissions to apply vouchers to bookings. * Vouchers should be applied before the booking is finalized to ensure accurate pricing. * The system should validate voucher codes to ensure they are active and applicable to the current booking. * Applied vouchers should be recorded for auditing and tracking purposes. | | | |

### 

### **3.2.12 Cancel Booking**

| **USE CASE-012 Cancel a booking** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC012 | **Use-case Version** | <1.0> |
| **Use-case Name** | Cancel a booking | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Customer  **Summary:**  This use case describes the process by The actor cancel a booking for a birthday party for kids within the system  **Goal:**  Provide The actor with the ability to cancel their booked birthday party  **Triggers**  The actor decides to cancel a booking for a birthday party.  **Preconditions:**  PRE-1: The actorr must be logged in to the system.  PRE-2: The booking to be canceled must exist within the system.  **Post Conditions:**  The booking is successfully canceled, and any associated resources are released.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | On the homepage, the customer hovers over the user name displayed on the navigation bar. | A dropdown list is is shown to the customer, which contains:   * Profile * Activities * Notification * Logout | | **2** | The actor selects “Activities” in the dropdown list. | Redirects to the Activities page displaying the options:   * Manage booking * Manage transaction * Support | | **3** | The actor clicks into the Manage booking on the navigation bar. | The system displays a list of bookings. | | **4** | The actor checks the booking they want to cancel. |  | | **5** | The actor click “ Cancel” button | System displays a notification “you want to cancel”   * “Yes” * “No”   [Exception 1] | | **6** | The actor click “Yes” button | System will cancel booking has been booked and store it in data |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | Bookings are overdue to be allowed to cancel. | The system displays a message: the deadline to cancel the party is overdue, please contact in advance to be able to do so. |   **Relationships:**  N/A  **Business Rules:**  The actor must booked in advance | | | |

### **3.2.13 View Booking Details**

| **USE CASE-013 View booking details** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC013 | **Use-case Version** | <1.0> |
| **Use-case Name** | View booking details | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Customer  **Summary:**  This use case describes the process by which The actor views the details of a booked birthday party within the system.  **Goal:**  Provide The actor with the ability review the details of their booked birthday party  **Triggers**  The actor decides to view the details of a booked birthday party.  **Preconditions:**  PRE-1: The actor must be logged in to the system.  PRE-2: The booking to be viewed must exist within the system.  **Post Conditions:**  Customer successfully views the details of the booked birthday party.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | On the homepage, the The actor hovers over the user name displayed on the navigation bar. | A dropdown list is is shown to the customer, which contains:   * Profile * Activities * Notification * Logout | | **2** | The actor selects “Activities” in the dropdown list.  [Alternative 1] | Redirects to the Activities page displaying the options:   * Manage booking * Manage transaction * Support | | **3** | The actor clicks into the Booking on the navigation bar. | The system displays a list of bookings. | | **4** | The actor selects 1 booking. | System display booking activities information include:   * Service name: Text * Name: Text * Phone: Must be in format (+84) xxx-xxx-xxxx * Address:Text * Time:Must be in a valid time format (e.g., 12:00 PM). * Rating button: |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor selects “Notification” in the dropdown list | The system redirects to the notification page and displays a list of notifications about the booking status with the “Rating” button in notifications with "Completed" status. | | **2** | The actor clicks on the party announcement. | System display booking activities information include:   * Service name * Name * Phone * Address * Time * Rating button |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Relationships:**  N/A  **Business Rules:**  N/A | | | |

### **3.2.14 View all booking**

| **USE CASE-14 View all booking** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC014 | **Use-case Version** | <1.0> |
| **Use-case Name** | View all booking | | |
| **Author** | Ta Ngoc An | | |
| **Date** | 2/02/2024 | **Priority** | Normal |
| **Actor:**  Customer  **Summary:**  This "use case" allows actors to view a list of all bookings made in the system.  **Goal:**  Allows actors to track and manage bookings made.  **Triggers**  The actor indicates that they want to view all bookings.  **Preconditions:**  PRE-1: The actor is logged into the system.  PRE-2: The device must be connected to the internet.  PRE-3: There are 1 or more booking in "complete" status.  **Post Conditions:**  POST-1: Displays a list of booking made and related details.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | On the homepage, the actor hovers over the user name displayed on the navigation bar. | A dropdown list is is shown to the actor, which contains:   * Profile * Activities * Notification * Logout | | **2** | The actor selects “Activities” in the dropdown list. | The system redirects to the Activities page displaying the options:   * Manage booking * Manage transaction * Support | | **3** | The actor clicks into the Booking on the navigation bar. | The system displays a list of bookings with attributes.   * Id * Basic information   [Exception 1] |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | There is no booking to display. | The system displays a message or status that no booking was found. |   **Relationships:**  N/A  **Business Rules:**  1. The actor can only view all booking during the current session and need to re-authenticate after a period of time. | | | |

### **3.2.15 Send comment/rating**

| **USE CASE-15 Send comment/rating** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC017 | **Use-case Version** |  | <1.0> |
| **Use-case Name** | Send comment/rating | | | |
| **Author** | Ta Ngoc An | | | |
| **Date** | 2/02/2024 | **Priority** |  | Normal |
| **Actor:**  Customer  **Summary:**  This "use case" allows actors to send a comment/rating about a party booked by them.  **Goal:**  Allows actors to share opinions and reviews about their booking experience.  **Triggers**  The actor indicates that they want to send a comment/rating about a party booked by them.  **Preconditions:**  PRE-1: The actor is logged into the system.  PRE-2: The device must be connected to the internet.  PRE-3: The booking already exists in the system and can be evaluated.  **Post Conditions:**  POST-1: Ratings and comments have been submitted and saved in the system.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | On the homepage, the actor hovers over the user name displayed on the navigation bar. | A dropdown list is is shown to the actor, which contains:   * Profile * Activities * Notification * Logout | | **2** | The actor selects “Activities” in the dropdown list.  [Alternative 1] | The system redirects to the Activities page displaying the options:   * Manage booking * Manage transaction * Support | | **3** | The actor clicks into the Booking on the navigation bar. | The system displays a list of bookings with attributes.   * Id * Basic information | | **4** | The actor selects 1 booking. | System display booking activities information include:   * Service name * Name * Phone * Address * Time * Rating button | | **4** | The actor clicks “Rating” button | Displays form rating in that booking include:   * “Add image”Button * Rating by star (5 stars in line) * Feedback * “Cancel” button * “Submit” button | | **5** | The actor selects a star rating (1-5)  The actor enters feedback (optional)  The actor add image (optional) |  | | **6** | The actor clicks “Submit” button | System validate rating value and record rating. If correct, the system shows the message”rating success”.  [Exception 1] |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor selects “Notification” in the dropdown list | The system redirects to the notification page and displays a list of notifications about the booking status with the “Rating” button in notifications with "Completed" status. | | **2** | The actor clicks “Rating” button | Displays form rating in that booking include:  ● “Add image”Button  ● Rating by star (5 stars in line)  ● Feedback  and the “Cancel” button and “Submit” button. | | **3** | The actor selects a star rating (1-5)  The actor enters feedback (optional)  The actor add image (optional) |  | | **6** | The actor clicks “Submit” button | System validate rating value and record rating. If correct, the system shows the message”rating success”.  [Exception 1] |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | The actor did not enter a review or the review content is invalid. | The system displays a warning message. |   **Relationships:**  N/A  **Business Rules:**  1. Each actor can only comment or rate once per booking.  2. The actor can only comment or rate parties that they have booked. | | | | |

### **3.2.16 View menu**

| **USE CASE-16 View menu** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC016 | **Use-case Version** | <1.0> |
| **Use-case Name** | View menu | | |
| **Author** | Anh Quốc | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Guest, Customer  **Summary:**  The View menu function allows actors at a birthday party to view the available menu options.  **Goal:**  The goal of this functionality is to help actors view and select menu options that suit their needs.  **Triggers**  Actors want to view menu options available at the birthday party.  **Preconditions:**   1. The menu has been prepared and is available for viewing.   **Post Conditions:**   1. Actors are informed about the available menu options. 2. Actors may proceed to make selections based on their preferences.   **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor accesses the system and the system navigates to the main page. | A dropdown list is is shown to the actor, which contains:   * View menu * Search a party * Post | | **2** | The actor clicks the “View menu” button. | The system redirects a menu of available product items.  [Exception 1] |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | Actors cannot access the menu because of a system error | The system will display that this page is having problems, please try again later. |   **Relationships:**  N/A  **Business Rules:**  N/A | | | |

### 

### **3.2.17 View birthday party detail**

| **USE CASE-017 View birthday party detail** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC017 | **Use-case Version** | <1.0> |
| **Use-case Name** | View birthday party detail | | |
| **Author** | Anh Quốc | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Customer, Guest  **Summary:**  The goal of this use case is to provide actors with necessary details about a birthday party event.  **Goal:**  The goal of this use case is to provide actors know about birthday details such as theme of party, music and activity.  **Triggers**  The actor wants to obtain information about the birthday party.  **Preconditions:**   1. The actor has been invited to the birthday party. 2. The actor has access to the system or platform where the party information is hosted.   **Post Conditions:**   1. The actor has successfully viewed the birthday party information.   **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor accesses the system and the system navigates to the main page. | A dropdown list is is shown to the actor, which contains:   * View menu * Search a party * Post | | **2** | The actor clicks the “View menu” button.  [Alternative 1] | The system redirects a menu of available product items.  [Exception 1] | | **3** | The actor clicks on the pre-created party options, the party for which they want to view. | The system retrieves and displays detailed information about the birthday party, including:   * Theme of the party * Menu * Decorate * Music * Activities and games * Contact Info |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor navigates to the event section. | The system lists upcoming events, including the birthday party. | | **2** | The actor clicks on the pre-created party options, the party for which they want to view. | The system retrieves and displays detailed information about the birthday party., including:   * Theme of the party * Menu * Decorate * Music * Activities and games * Contact Info |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | In case of a technical error or system failure. | The system displays a generic error message and prompts the user to try again later. | | **2** | Server downtime or connectivity problems. | The system displays an error message informing the guest about the issue and suggests trying again later. |   **Relationships:**  N/A  **Business Rules:**  Only invited actors should have access to view the birthday party information to maintain privacy and security. | | | |

### 

### **3.2.18 Search a party**

| **USE CASE-018 Search a party** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC018 | **Use-case Version** | <1.0> |
| **Use-case Name** | Search a party | | |
| **Author** | Anh Quốc | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Guest, Customer  **Summary:**  This function allows actors to search for information about a specific party.  **Goal:**  The goal of this function is to search for and gather information about a specific party.  **Triggers**  The actor wants to search for information about a party.  **Preconditions:**  The system must have information about the party the actor wants to search for.  **Post Conditions:**  The actor receives information about the party after a successful search.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor accesses the system and navigates to the main menu. | A dropdown list is is shown to the actor, which contains:   * View menu * Search a party * Post | | **2** | The actor clicks “Search a party” on the menu. | The system redirects a search bar and requires customers to enter party information | | **3** | The actor enters the information they want to search for about the party into the system. | [Exception 1] | | **4** | The actor click “Search” button  [Alternative 1] | The system processes the request and displays the search results. |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | Actors could not find party information. | The system notifies the actor about the lack of search results. |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | The actor entered invalid information. | The actor is notified of the issue and prompted to check the internet connection. |   **Relationships:**  N/A  **Business Rules:**  Actors must enter correct party information such as date, time. | | | |

### 

### **3.2.19 View host party detail**

| **USE CASE-019 View host party detail** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC019 | **Use-case Version** | <1.0> |
| **Use-case Name** | View host party detail | | |
| **Author** | Anh Quốc | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Guest, Customer  **Summary:**  The function allows actors to view details about a party host.  **Goal:**  The purpose of the function is for actors to grasp information about the party host.  **Triggers**  The trigger for using this case is when an actor wants to find information about the party host.  **Preconditions:**   1. Actors access the product link and select the party host's avatar to view information. 2. The party information must be available in the system.   **Post Conditions:**   1. The actor has successfully viewed host party information.   **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | On the homepage, the actor clicks on the pre-created party options, the party for which they want to see comments and ratings. | System redirects the actor to the party detail page and show the information include:   * Add to cart button * Book now button * Party option buttons | | **2** | On the party details page, the actor scroll down and select the host party‘s avatar. | The system redirects party host page, and buttons:   * Profile of the party host * Contact information for the party host. * View reviews from other customers for the party host. | | **3** | The actor clicks the “Profile of the party host” button of the party.  [Alternative 1] | The system redirects the host party information:   * Name: Must be a non-empty string. * Date of birth: Must be in the format DD/MM/YYYY (e.g., 31/03/2024). * Venue: Must be a valid venue option available in the system. * Phone number: Must follow a country-specific format (e.g. +84 to represent Vietnam, +1 for the United States).   [Exception 1] |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | If the actor has hidden personal information. | The system notifies the actor that: The actor has hidden personal information. |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | Technical Error | In case of a technical error or system failure, the system displays an error message and advises the actor to try again later. |   **Relationships:**  N/A  **Business Rules:**  The actor can only view party information for parties they have been invited to or have access to through other means. | | | |

### 

### **3.2.20 View all post of admin**

| **USE CASE-020 View all post of admin** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC020 | **Use-case Version** | <1.0> |
| **Use-case Name** | View all post of admin | | |
| **Author** | Anh Quốc | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Guest  **Summary:**  The "View All Posts of Admin" functionality allows actors to access and view all posts shared by the admin of a party event.  **Goal:**  The goal of this use case is to provide actors with a comprehensive view of all posts shared by the host of a party event, including announcements, updates, photos, and other relevant information.  **Triggers**  The actor wants to stay informed about the latest updates and announcements regarding a party event by the admin.  **Preconditions:**   1. The actor is logged into the party event's platform or social media page. 2. The party event hosted by the specific host has posts shared by the host.   **Post Conditions:**   1. The actor has successfully viewed all posts shared by the host of the party event. 2. No changes are made to the posts during the viewing process.   **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor accesses the system and navigates to the main menu. | A dropdown list is is shown to the actor, which contains:   * View menu * Search a party * Post | | **2** | The actor clicks “Post” on the menu. | The system redirects a list of admin posts. | | **2** | The actor selects the title of the post they want to view.  [Alternative 1] | The system presents post shared by the admin, including:   * Birthday party information * Photo: Photos: Must be in an acceptable format (e.g., JPEG, PNG) and adhere to the platform's size and resolution restrictions * Video: Must be in an acceptable format (e.g., MP4, AVI). * Other multimedia content: Any additional multimedia content (e.g., audio clips, GIFs) should be in acceptable formats and adhere to the platform's guidelines.   [Exception 1] |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor notices that some posts are missing or not visible. | The system prompts the actor to refresh the page or check back later if there are any issues with loading the posts. |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | Information for this post is not available | The system will display that this posts information has been deleted or permissions changed so it cannot be displayed. |   **Relationships:**  N/A  **Business Rules:**   1. The actor should have access to view all posts shared by the admin to stay informed about the party event. | | | |

### 

### **3.2.21 View party order detail**

| **USE CASE-021 View party order detail** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC021 | **Use-case Version** | <1.0> |
| **Use-case Name** | View party order detail | | |
| **Author** | Anh Quốc | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Party host  **Summary:**  The "View Party Order Information" functionality allows actors to access detailed information about a specific party ordering.  **Goal:**  The goal of this use case is to provide the actor with comprehensive details regarding the party booking, including date, time, venue, guest count, selected package, and any special requests.  **Triggers**  The actor requires detailed information about a particular party booking for planning or coordination purposes.  **Preconditions:**   1. The actor is logged into the system. 2. At least one party booking exists in the system.   **Post Conditions:**   1. The actor has successfully viewed the detailed information of the party booking. 2. No changes are made to the party booking during the viewing process.   **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor logs into the system and navigates to the main menu. | A menu on the left is displayed for the Party host , which contains:   * Manage birthday party booking * Manage birthday party information * Manage package | | **2** | The actor clicks the “Manage birthday party booking” button. | The system redirects a list of booked parties, and buttons:   * View party orders information * Confirm party orders * Complete party orders * Cancel party orders | | **3** | The actor chose “View party birthday information”. | The system retrieves the list of booked parties and displays them to the actor.   * Information about booking customer’s birthday party * “Exit” button   [Exception 1] | | **4** | The actor selects “Information about booking an actor's birthday party” from the list.  [Alternative 1] | The system retrieves and displays detailed information about the selected party booking, including:   * Date: Must be in the format DD/MM/YYYY (e.g., 31/03/2024). * Time: Must be in a valid time format (e.g., 12:00 PM). * Venue: Must be a valid venue option available in the system. * Guest Count: Must be a positive integer indicating the number of guests attending the party. * Selected Package: Must be a valid package option available in the system. * Any Special Requests: Free-form text field allowing the party host to provide additional details or requests for the party event. * “Back” button | | **5** | The actor clicks the “Back” button to return to the previous page. | The system redirects the list again party order information. |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor enters search criteria to filter the list of party bookings. | The system redirects filtered results based on the search criteria provided by the actor. |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | The actor selects a birthday party from the list, but the system returns to the home page. | The system notifies that the information no longer exists and returns to the home page. |   **Relationships:**  This use case relates to the "Manage Party Bookings" feature, as it is part of the overall party booking management process.  **Business Rules:**   1. The actor can only view party order information for bookings associated with their account. 2. Detailed party order information should be presented accurately and clearly to the actor for effective planning and coordination. | | | |

### **3.2.22 Confirm party orders**

| **USE CASE-022 Confirm party orders** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC022 | **Use-case Version** | <1.0> |
| **Use-case Name** | Confirm booking orders | | |
| **Author** | Anh Quốc | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Party host  **Summary:**  Confirm party birthday booking is a function that allows the host or party planner to finalize the arrangements and details for a birthday party booking.  **Goal:**  The actor confirms all necessary details and arrangements for the birthday party booking, ensuring a successful event.  **Triggers**  This use case is triggered when an actor receives a request for a birthday party booking.  **Preconditions:**  PRE-1: The actor has received a request for a birthday party booking.  PRE-2: Relevant venue or location availability has been confirmed.  **Post Conditions:**  POST-1: The birthday party booking is confirmed with all necessary details finalized.  POST-2: Any required deposits or payments are made.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor accesses the booking system or event management platform. | A menu on the left is displayed for the Party host , which contains:   * Manage birthday party booking * Manage birthday party information * Manage package | | **2** | The actor clicks the “Manage birthday party booking” button. | The system will redirects the following functions:   * View party orders information * Confirm party orders * Complete party orders * Cancel party orders | | **3** | The actor clicks the “Confirm party orders” button of the party which they want to confirm.  [Alternative 1] | The system prompts the party host to review the selected party booking details once more to ensure accuracy.   * “Yes” button * “No” button   [Exception 1] | | **4** | The actor clicks “Yes” to confirm the party.  [Alternative 2] | The system finalizes the booking process and updates the status of the party order to "confirmed." |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | The system does not display information that needs confirmation | There is a system error, and the confirmation process cannot proceed. The system displays an error message, informing the party host to try again later or to contact technical support for assistance. |   **Relationships:**  N/A  **Business Rules:**  1. The actor should review all details of the booking for accuracy and completeness.  2. The selected date and time for the party must be available in the system and not conflicting with any existing bookings.  3.Payment for the party booking must be successfully processed before confirming the order. | | | |

### **3.2.23 Complete party orders**

| **USE CASE-023 Complete party orders** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC023 | **Use-case Version** | <1.0> |
| **Use-case Name** | Complete party orders | | |
| **Author** | Anh Quốc | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Party host  **Summary:**  The "Complete Party Orders" functionality allows the actor to finalize and complete the booking process for a party event.  **Goal:**  The goal of this use case is to ensure that all necessary arrangements for the party event are confirmed, payments are processed, and relevant notifications are sent out.  **Triggers**  The actor has confirmed all details and is ready to finalize the party booking.  **Preconditions:**   1. The actor has confirmed the party details, including date, time, venue, and package. 2. Payment for the party booking has been successfully processed.   **Post Conditions:**   1. The party booking is marked as "completed" in the system. 2. Confirmation notifications are sent out to the actor and any other relevant parties involved.   **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor accesses the booking system or event management platform. | A menu on the left is displayed for the Party host , which contains:   * Manage birthday party booking * Manage birthday party information * Manage package | | **2** | The actor clicks the “Manage birthday party booking” button. | The system redirects a list of booked parties, and buttons:   * View party orders information * Confirm party orders * Complete party orders * Cancel party orders | | **3** | The actor clicks the “Complete party orders” button of the party.  [Alternative 1] | The system prompts the party host to review the booking details for one final time.   * “Yes” button * “No” button | | **4** | The actor clicks the “Yes” to complete the party. | The system updates the status of the party booking to "completed" and sends out confirmation notifications to the party host and other involved parties.  [Exception 1] |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor notices an error or discrepancy in the details. | The system allows the party host to make necessary corrections before proceeding with completion. |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | System error or technical glitch. | The system displays an error message and advises the party host to try again later or contact technical support for assistance. |   **Relationships:**  This use case relates to the "Confirm Party Orders" functionality, as completing party orders is the next step after confirmation.  **Business Rules:**   1. Completion of party orders must be initiated by the party host and confirmed before finalization. 2. Confirmation notifications must be sent out promptly after the completion process is successful. | | | |

### 

### **3.2.24 Cancel party orders**

| **USE CASE-024 Cancel party booking** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC024 | **Use-case Version** | <1.0> |
| **Use-case Name** | Cancel party orders | | |
| **Author** | Anh Quốc | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Party host  **Summary:**  This use case describes the process of canceling a party ordering.  **Goal:**  Allow the actor to cancel a previous party order.  **Triggers**  The actor decides to cancel party orders.  **Preconditions:**  Provide accurate information about the order to be canceled, including an order code or order identification information sufficient to accurately identify the order to be canceled.  **Post Conditions:**   1. The actor order is successfully canceled. 2. Any associated payments are refunded according to the cancellation policy.   **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The party host accesses the booking system or event management platform. | A menu on the left is displayed for the Party host , which contains:   * Manage birthday party booking * Manage birthday party information * Manage package | | **2** | The party host clicks “Manage birthday party booking”. | The system will redirects the following functions:   * View party orders information * Confirm party orders * Complete party orders * Cancel party orders | | **4** | The party host clicks the “Cancel party orders” button. | The system will redirect a list of party orders that can be canceled. | | **5** | The party host selects “Accept” to cancel party orders. | The system displays whether you are sure you want to cancel party orders.   * “Accept” button * “Quit” button | | **5** | Party host click “Accept”  [Alternative 1] | The system displays a notification of successful cancellation and deletion of the birthday party order.  [Exception 1] |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | If the cancellation request is made outside the specified cancellation window | Notification system: The time allowed to cancel the reservation has expired. |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  Party cancellation must be within the specified time. | | | |

### **3.2.25 Create birthday party**

| **USE CASE-025 Create birthday party** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC 027 | **Use-case Version** | <1.0> |
| **Use-case Name** | Create birthday party | | |
| **Author** | Anh Quốc | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Party host  **Summary:**  The "Create Birthday Party" functionality allows the party host to initiate the process of booking a birthday party event.  **Goal:**  The goal of this use case is to enable the party host to specify the details of the birthday party event and begin the booking process.  **Triggers**  The party host intends to organize a birthday party event and needs to initiate the booking process.  **Preconditions:**   1. The party host is logged into the system. 2. The party host has access to the birthday party booking feature.   **Post Conditions:**   1. The party details are successfully recorded in the system for further processing. 2. The party host receives confirmation of the initiation of the booking process.   **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The party host accesses the birthday party management platform. | A menu on the left is displayed for the Party host , which contains:   * Manage birthday party booking * Manage birthday party information * Manage package | | **2** | The party host clicks the “Manage birthday party information” button. | The system redirects a list of manage parties, and buttons:   * Create a birthday party. * View birthday party list. * Update birthday party. * Delete birthday party. | | **3** | The party host clicks the “Create a birthday party.” button of the party.  [Alternative 1] | The system presents a form for the party host to fill in the details of the birthday party event.   * Name: Must be a non-empty string, representing the name of the person for whom the birthday party is being organized. * Date Birth: Must be in the format DD/MM/YYYY, indicating the birth date of the person for whom the birthday party is being organized. * Date Party Orders: Must be in the format DD/MM/YYYY, indicating the date when the party booking is being made. * Time: Must be in a valid time format (e.g., 12:00 PM), indicating the time at which the party will start. * Venue: Must be a valid venue option available in the system where the birthday party can be hosted. * Guest Count: Must be a positive integer indicating the number of guests expected to attend the birthday party.   [Exception 1] | | **4** | The party host fills in all of the information above. | The system displays the information filled in. | | **4** | The party host reviewed and confirmed the entered details. | The system records the party details and initiates the booking process. |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | At least one input field doesn’t match the constraint. | System shows an error message corresponding to the fail constraint. |   **Relationships:**  N/A  **Business Rules:**   1. Date and time must be in the correct format 2. The party host must provide all required details accurately to initiate the booking process. | | | |

### 

### **3.2.26 View birthday party list**

| **USE CASE-026 View birthday party list** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC026 | **Use-case Version** | <1.0> |
| **Use-case Name** | View birthday party list | | |
| **Author** | Anh Quốc | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Party host  **Summary:**  The "View birthday party list" functionality enables the party host to access and review details of an existing birthday party event.  **Goal:**  The goal of this use case is to provide the party host with comprehensive information about a specific birthday party event, including date, time, venue, guest count, and any special requests.  **Triggers**  The party host needs to review the details of a previously created birthday party event.  **Preconditions:**   1. The party host is logged into the system. 2. At least one birthday party event has been created and exists in the system.   **Post Conditions:**   1. The party host has successfully viewed the detailed information of the birthday party event. 2. No changes are made to the birthday party event during the viewing process.   **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The party host accesses the birthday party management platform. | A menu on the left is displayed for the Party host , which contains:   * Manage birthday party booking * Manage birthday party information * Manage package | | **2** | The party host clicks the “Manage birthday party information” button. | The system redirects a list of manage parties, and buttons:   * Create a birthday party. * View birthday party list. * Update birthday party. * Delete birthday party. | | **3** | The party host clicks the “View birthday party list” button. | The system retrieves the list of created birthday party events and displays them to the party host. | | **4** | TThe party host selects a specific birthday party event from the list.  [Alternative 1] | The system retrieves and displays detailed information about the selected birthday party event, including:   * Name: Must be a non-empty string. * Date: Must be in the format DD/MM/YYYY (e.g., 31/03/2024). * Time: Must be in a valid time format (e.g., 12:00 PM). * Venue: Must be a valid venue option available in the system. * Guest Count: Must be a positive integer indicating the number of guests attending the party. * Selected Package: Must be a valid package option available in the system.   [Exception 1] |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The party host enters search criteria to filter the list of birthday party events. | The system displays filtered results based on the search criteria provided by the party host. |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | Birthday party information is not available due to a system error that has not been updated. | In case of a technical error or system failure, the system displays information of a birthday party that no longer exists and asks the party host to come back later. |   **Relationships:**  N/A  **Business Rules:**  The party host can only view birthday party details for events associated with their account. | | | |

### 

### **3.2.27 Update Birthday party**

| **USE CASE-027 Update birthday party** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC 027 | **Use-case Version** | <1.0> |
| **Use-case Name** | Update birthday party | | |
| **Author** | Tín | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Party host  **Summary:**  The "Update Birthday Party" functionality allows the actor to modify and update the party.  **Goal:**  The goal is to enable the actor to edit for specific items from the party birthday.  **Triggers**  Actors can utilize filters or select special party options to access party information.  **Preconditions:**  Actor accesses the system and edits the party.  **Post Conditions:**  After selecting a party from the list, the actor updates its details, including the order number, customer information, party details, date, time, and status.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The party host accesses the birthday party management platform. | A menu on the left is displayed for the Party host , which contains:   * Manage birthday party booking * Manage birthday party information * Manage package | | **2** | The actor clicks”“Manage birthday party information” | The system displays a list of manage parties, and buttons:   * Create a birthday party. * View birthday party list. * Update birthday party. * Delete birthday party. | | **3** | The actor chose “Update birthday party”. | The system displays existing birthday parties. | | **4** | The actor chose a birthday party that needs to be updated. | The system presents a form for the actor to fill in the details of the birthday party event.   * Name: Must be a non-empty string, representing the name of the person for whom the birthday party is being organized. * Date Birth: Must be in the format DD/MM/YYYY, indicating the birth date of the person for whom the birthday party is being organized. * Date Party Orders: Must be in the format DD/MM/YYYY, indicating the date when the party booking is being made. * Time: Must be in a valid time format (e.g., 12:00 PM), indicating the time at which the party will start. * Venue: Must be a valid venue option available in the system where the birthday party can be hosted. * Guest Count: Must be a positive integer indicating the number of guests expected to attend the birthday party. | | **5** | The actor will edit the information that needs to be updated. | The system displays the newly changed information and asks if you are sure you want to update the above information:   * “Yes” button * “No” button   [Exception 1] | | **6** | The actor clicks the “Yes” button to update. | The system displays the birthday party's information and has been successfully updated. |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | The actor edited the information about the birthday incorrectly. | The system displays that the actor entered incorrect information and asks to re-enter it |   **Relationships:**  N/A  **Business Rules:**   * The system must verify and update inventory quantity before adding items to the cart to ensure party availability. | | | |

### **3.2.28 Delete Birthday party**

| **USE CASE-028 Delete birthday party** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC028 | **Use-case Version** | <1.0> |
| **Use-case Name** | Delete birthday party | | |
| **Author** | Tín | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Party host  **Summary:**  This actor case describes the process of deleting a birthday party from the system.  **Goal:**  The goal is to enable the actor to delete items from the party birthday.  **Triggers**  Actor can select special party options to delete party information.  **Preconditions:**  The actor must have access to the system.  The actor chooses the function “delete party birthday” to delete party birthday.  **Post Conditions:**  Actor delete successful birthday parties.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor accesses the birthday party management platform. | A menu on the left is displayed for the Actor, which contains:   * Manage birthday party booking * Manage birthday party information * Manage package | | **2** | The actor clicks”“Manage birthday party information” | The system displays a list of manage parties, and buttons:   * Create a birthday party. * View birthday party list. * Update birthday party. * Delete birthday party. | | **3** | The actor chose “Delete birthday party” to delete birthday party. | The system will display birthday parties that need to be deleted. | | **4** | The actor chose a birthday party list to delete. | The system displays whether you are sure you want to delete this birthday party.   * “Yes” button * “No” button | | **5** | The actor clicks the “Yes” button. | The system confirms that you have successfully deleted the birthday party. |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | * Party temporarily disabled. | * The system reports errors and provides detailed information to the actor. |   **Relationships:**  N/A  **Business Rules:**  The hoster must be logged into their account to delete a party.  The system must verify the deletion action to ensure data integrity and compliance with business rules | | | |

### **3.2.29 Disable package**

### 

| **USE CASE-029 Disable package** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC029 | **Use-case Version** | <1.0> |
| **Use-case Name** | Disable package | | |
| **Author** | Tín | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Party host  **Summary:**  This use case describes the process of deleting a party package from the system.  **Goal:**  To allow the actor to manage party packages by deleting them when necessary.  **Triggers**  The actor decides to delete a specific party package.  **Preconditions:**   * The actor must have access to the system. * The application must display all relevant information regarding birthday parties and party packages.   **Post Conditions:**  The deleted party package is removed from the system and is no longer available for selection or booking..  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor navigates to the package management section of the system. | A menu on the left is displayed for the Actor , which contains:   * Manage birthday party booking * Manage birthday party information * Manage package | | **2** | The actor chose “Manage package”. | The system displays the following functions:   * Disable package * Enable package * Create package * Update package * Delete package   [Exception 1] | | **3** | The actor clicks “ Disable package”. | The system will display the packages that need to be enabled. | | **4** | The actor chose a package to disable | The system displays the package you are selecting and confirms you want to disable the package.   * “Yes” button * “No” button | | **5** | The actor clicks the “Yes” button. | The system displays the disabled package successfully. |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | The actor decides to delete a specific party package. | * The system prompts the actor to confirm the deletion of the party package. |   **Relationships:**  N/A  **Business Rules:**   * The actor must be logged into their account to update a party package. * The system must verify the update action to ensure data integrity and compliance with business rules. | | | |

### **3.2.30 Enable package**

### 

| **USE CASE-030 Enable package** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC030 | **Use-case Version** | <1.0> |
| **Use-case Name** | Enable package | | |
| **Author** | Tín | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Party host  **Summary:**  This use case actor describes the process of enabling a party package from the system.  **Goal:**  To allow the actor to manage party packages by enabling them when necessary.  **Triggers**  The actor requested activation permission successfully.  **Preconditions:**   * The actor must have access to the system. * The application must display information that needs to enable the package.   **Post Conditions:**   * The enabled party package becomes available for selection and booking in the system.   **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor navigates to the package management section of the system. | A menu on the left is displayed for the Actor , which contains:   * Manage birthday party booking * Manage birthday party information * Manage package | | **2** | The actor chose “Manage package”. | The system displays the following functions:   * Disable package * Enable package * Create package * Update package * Delete package   [Exception 1] | | **3** | The actor clicks “Enable package”. | The system will display the packages that need to be enabled. | | **4** | The actorchose a package to enable. | The system displays the package you are selecting and confirms you want to enable the package.   * “Yes” button * “No” button | | **5** | The actor clicks the “Yes” button. | The system displays the enabled package successfully. |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | The Actor decides to delete a specific party package. | The system prompts the actor to confirm the deletion of the party package. |   **Relationships:**  N/A  **Business Rules:**   * N/A | | | |

### 

### **3.2.31 Create package**

### 

| **USE CASE-031 Create package** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC031 | **Use-case Version** | <1.0> |
| **Use-case Name** | Create package | | |
| **Author** | Tín | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Party host  **Summary:**  This use case describes the process of creating a party package in the system.  **Goal:**  To allow the actor to create and manage party packages.  **Triggers**  The actor decides to create a new party package.  **Preconditions:**   * The actor must have access to the system. * The application must display all relevant information regarding birthday parties and party packages.   **Post Conditions:**  The newly created party package is added to the system and becomes available for selection and booking.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor navigates to the package management section of the system. | A menu on the left is displayed for the actor , which contains:   * Manage birthday party booking * Manage birthday party information * Manage package | | **2** | The actor chose “Manage package”. | The system displays the following functions:   * Disable package * Enable package * Create package * Update package * Delete package   [Exception 1] | | **3** | The actor clicks “Create package”. | The system presents a form for the actor to fill in the details of the package for a birthday party event:   * Name of package: Must be a non-empty string. * Topic of package * Price: Must be a number. * Date: Must be in the format DD/MM/YYYY (e.g., 31/03/2024). * Time: Must be in a valid time format (e.g., 12:00 PM). * Count of guests: Must be a number. | | **4** | The actor fills all details with the package. | [Exception 1] | | **5** | The actor clicks the “Confirm” button. | The system displays the completed and confirmed information:   * “Yes” button * “No” button | | **6** | The actor clicks the “Yes” button. | The system displays that the package has been created successfully. |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | At least one input field doesn’t match the constraint. | System shows an error message corresponding to the fail constraint. |   **Relationships:**  N/A  **Business Rules:**   * The package created must be in accordance with the criteria and policies of the relevant parties. | | | |

### 

### **3.2.32 Update package**

### 

| **USE CASE-032 Update package** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC032 | **Use-case Version** | <1.0> |
| **Use-case Name** | Update package | | |
| **Author** | Tín | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Party host  **Summary:**  This use case describes the process of updating a party package in the system.  **Goal:**  To allow the actor to modify and manage party packages.  **Triggers**  The actor decides to update a specific party package.  **Preconditions:**   * The actor must have access to the system. * The application must display all relevant information regarding birthday parties and party packages.   **Post Conditions:**  The updated party package reflects the changes made by the hoster and remains available for selection and booking in the system.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor accesses the birthday party management platform. | A menu on the left is displayed for the Actor , which contains:   * Manage birthday party booking * Manage birthday party information * Manage package | | **2** | The actor clicks the “Manage package” button. | The system displays the following functions:   * Disable package * Enable package * Create package * Update package * Delete package | | **3** | The actor clicks “Update package”. | The system will display the list of packages that need to be updated. | | **4** | The actor chose a package of lists to update. | The system presents a form for the actor to fill in the details of the package:   * Name of package: Must be a non-empty string. * Topic of package * Price: Must be a number. * Date: Must be in the format DD/MM/YYYY (e.g., 31/03/2024). * Time: Must be in a valid time format (e.g., 12:00 PM). * Count of guests: Must be a number. | | **5** | The actor edits details in the package and confirms. | The system displays form has been filled and confirmed all information:   * “yes” button * “No” button   [Exception 1] | | **6** | The actor clicks the “Yes” button. | The system notices that all information has been successfully updated. |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | At least one input field doesn’t match the constraint. | System shows an error message corresponding to the fail constraint. |   **Relationships:**  N/A  **Business Rules:**   * The system must verify the update action to ensure data integrity and compliance. | | | |

### 

### **3.2.33 Delete package**

### 

| **USE CASE-033 Delete package** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC033 | **Use-case Version** | <1.0> |
| **Use-case Name** | Delete package | | |
| **Author** | Tín | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Party host  **Summary:**  This use case describes the process of deleting a party package from the system.  **Goal:**  To allow the pactor to manage party packages by deleting them when necessary.  **Triggers**  The actor decides to delete a specific party package.  **Preconditions:**   * The actor must have access to the system. * The application must display all relevant information regarding birthday parties and party packages.   **Post Conditions:**  The deleted party package is removed from the system and is no longer available for selection or booking..  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor navigates to the package management section of the system. | A menu on the left is displayed for the Party host , which contains:   * Manage birthday party booking * Manage birthday party information * Manage package | | **2** | The actor clicks the “Manage package” button. | The system displays the following functions:   * Disable package * Enable package * Create package * Update package * Delete package | | **3** | The actor chose “Delete package” . | The system displays a list of packages that can be deleted. | | **4** | The actor chose a package of lists. | The system notices that the party host wants to delete this package.   * “Yes” button * “No” button   [Exception 1] | | **5** | The actor clicks the “Yes” button. | The system shows status that the package has been deleted. |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | Cannot be deleted due to technical error | The system prompts the party host that the package can not be deleted and requires the party host to correct it later. |   **Relationships:**  N/A  **Business Rules:**   * The system must verify the delete action to ensure data integrity and compliance. | | | |

### 

### **3.2.34 Review package / party content**

### 

| **USE CASE-034 Review package / party content** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC034 | **Use-case Version** | <1.0> |
| **Use-case Name** | Review package / party content | | |
| **Author** | Tín | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Administrator  **Summary:**  This use case describes the process of deleting a party package from the system.  **Goal:**  To allow the actor to manage party packages by deleting them when necessary.  **Triggers**  The actor decides to delete a specific party package.  **Preconditions:**   * The actor must have access to the system. * The application must display all relevant information regarding birthday parties and party packages.   **Post Conditions:**   * The enabled party package becomes available for selection and booking in the system.   **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor selects the “Review” tab. | The system displays a list of available packages or party content for review. | | **2** | The actor choose a package or party which they want to review | The system presents detailed information about the selected package or party content, including:   * Name * Description * Price * Duration * Services/features | | **3** | The actor makes changes if necessary to the selected package and party. |  | | **4** | The actor clicks the “Confirm” button. | The system validates the modified information and provides feedback on any errors or inconsistencies. |   **Alternative Scenario:**  N/A  **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**   * The actor must be logged into their account to review and delete a party package. * The system must verify the deletion action to ensure data integrity and compliance with business rules. | | | |

### 

### **3.2.35 Enable customer accounts**

| **USE CASE-035 Enable patro accounts** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC035 | **Use-case Version** | <1.0> |
| **Use-case Name** | Enable patro accounts | | |
| **Author** | Tín | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Administrator  **Summary**:  The actor is granted access to the system to search for information about birthday parties, enabling them to view relevant party details and make necessary modifications.  **Goal**:  The goal is to facilitate the actor in searching for specific items from the party menu and accessing relevant party information.  **Triggers:**  Actor can initiate searches by applying filters or selecting special party options to access party information.  **Preconditions:**   * The actor must be authenticated and have access to the system to initiate a search for the desired party. * The application must present all relevant information regarding birthday parties.   **Post Conditions:**  After selecting a party from the list, the user (hoster) can view its details, including the order number, customer information, party details, date, time, and status..  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor selects the “Account” tab. | Show a drop down list of options related to Post   * Enable customer account * Disable customer account | | **2** | The actor click “Enable customer account” button | Redirect to enable customer account page that shows a list of customer accounts. | | **3** | The actor selects the account they want to enable. | The system prompts the administrator to confirm the action with a message like "Are you sure you want to enable this customer account?"   * “Yes” * “No” | | **4** | The actor click “Yes” button | The system enables the selected customer account and updates its status to "enabled."  [Exception 1] |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | Technical issue preventing the disabling of the patron/party host account. | The system displays an error message and advises the administrator to try again later or contact support for assistance. |   **Relationships:**  N/A  **Business Rules:**  N/A | | | |

### **3.2.36 Disable customer accounts**

| **USE CASE-036 Disable customer accounts** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC036 | **Use-case Version** | <1.0> |
| **Use-case Name** | Disable customer accounts | | |
| **Author** | Tín | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Administrator  **Summary:**  The actor , acting as a actor , is granted access to the system to search for information about birthday parties. The goal is to allow the hoster to search for specific items from the party menu and view relevant party information.  **Goal:**  The goal is to enable the hoster to efficiently manage party-related information and interactions within the system.  **Triggers**  Actor can utilize filters or select special party options to access party information.  **Preconditions:**   * The actor must be logged into the system to initiate actions related to party management. * The application must display all relevant information regarding birthday parties to the actor .   **Post Conditions:**   * After the actor account is disabled, the user will no longer have access to the system and cannot perform any actions related to party management.   **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor selects the “Account” tab. | Show a drop down list of options related to Post   * Enable customer account * Disable customer account | | **2** | The actor click “Disable customer account” button | Redirect to disable customer account page that shows a list of customer accounts. | | **3** | The actor selects the account they want to disable. | The system prompts the administrator to confirm the action with a message like "Are you sure you want to disable this customer account?"   * “Yes” * “No” | | **4** | The actor click “Yes” button | The system enables the selected customer account and updates its status to "disable."  [Exception 1] |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | Technical issue preventing the disabling of the actor account. | The system displays an error message and advises the hoster to try again later or contact support for assistance. |   **Relationships:**  N/A  **Business Rules:**  N/A | | | |

### 

### **3.2.37 Delete Comment**

| **USE CASE-37 Delete Comment** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC037 | **Use-case Version** | <1.0> |
| **Use-case Name** | Delete Comment | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Admin  **Summary:**  This use case describes the process of The actor deleting a comment  **Goal:**  Enable The actor to remove inappropriate or irrelevant comments from the system  **Triggers**  The actor decides to delete a comment.  **Preconditions:**  PRE-1: The actor must be logged in to the system.  PRE-2: The comment to be deleted must exist in the system.  **Post Conditions:**  POST-1: The comment is successfully removed from the system.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor selects the “Booking” tab. | Show a drop down list of options related to Booking   * Booking * Transaction * Comment | | **2** | The actor click the “Comment” | Redirect to comment page | | **3** | The actor clicks the filter comment. | System will display negative comments. | | **4** | The actor chooses negative comments. | System displays a notification you want to delete this comment   * “Yes” * “No” | | **5** | The actor click “Yes” button | System will delete the comment has been deleted and store it in data |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Relationships:**  N/A  **Business Rules:**  Only The actor can delete comment | | | |

### **3.2.38 View List Post**

| **USE CASE-38 Delete Comment** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC038 | **Use-case Version** | <1.0> |
| **Use-case Name** | View Post | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Admin  **Summary:**  This use case describes the process by The actor views posts  **Goal:**  Enable The actor to review posts related to booking birthday parties for kids.  **Triggers**  The actor decides to view posts within the system.  **Preconditions:**  PRE-1: The actor must be logged in to the system  PRE-2: The post to be accepted must be submitted and awaiting approval.  **Post Conditions:**  The actor successfully views the posts within the system.  Main Success Scenario:  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor selects the “Post” tab. | Show a drop down list of options related to Post   * Create Post * View List post * Update a post * Delete a post | | **2** | The actor clicks the “View List Post” button. | Redirect a list of posts page |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Relationships:**  N/A  **Business Rules:**  Only The actor can View List Post | | | |

### **3.2.39 Create A Post**

| **USE CASE-039 Create A Post** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC039 | **Use-case Version** | <1.0> |
| **Use-case Name** | Create a post | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Admin  **Summary:**  This use case describes the process by The actor accepts a post  **Goal:**  Enable The actor to review and approve posts related to booking birthday parties for kids.  **Triggers**  The actor decides to accept a post for publication.  **Preconditions:**  PRE-1: The actor must be logged in to the system  PRE-2: The post to be accepted must be submitted and awaiting approval.  **Post Conditions:**  The selected post is successfully accepted and published within the system.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor selects the “Post” tab. | Show a drop down list of options related to Post   * Create Post * View List post * Update a post * Delete a post | | **2** | The actor click “Create post” | Redirect to view post page that show a list of waiting accept form including:   * Title: Text * Describe: Text * Image: PNG,...   Each post has a response button to accept the post | | **3** | The actor fills all attributes in the form. | [Exception 1] | | **4** | The actor click “Create” button | System will create the post and store it in data |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | At least one input field doesn’t match the constraint. | System shows error message corresponding to the fail constrain |   **Relationships:**  N/A  **Business Rules:**  Only The actor can Create Post | | | |

### **3.2.40 View Transaction**

| **USE CASE-040 View Transaction** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC040 | **Use-case Version** | <1.0> |
| **Use-case Name** | View Transaction | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Admin  **Summary:**  This use case describes the process by The actor views transactions  **Goal:**  Enable The actor to monitor and review transactions related to booking birthday parties for kids..  **Triggers**  The actor decides to view transactions within the system..  **Preconditions:**  PRE-1: The actor must be logged in to the system  PRE-2: There must be transaction data available within the system.  **Post Conditions:**  The actor successfully views the transactions within the system.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor selects the “Booking” tab. | Show a drop down list of options related to Booking   * Booking * Transaction * Comment | | **2** | The actor click on “Transaction” button | Redirect transaction page | | **3** | The actor choose 1 transaction | System displays the details of the selected transaction form including   * ID:text * Date: Must be in the format DD/MM/YYYY (e.g., 31/03/2024). * Price:Must be in the format xVND(e.g., 200.000VND) * Status:Must be in the format success/fail(e.g.,transaction successful) |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | N/A | .N/A |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Relationships:**  N/A  **Business Rules:**  Only The actor can View Transaction | | | |

### **3.2.41 View Booking**

| **USE CASE-041 View Booking** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC041 | **Use-case Version** | <1.0> |
| **Use-case Name** | View Booking | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Admin  **Summary:**  This use case describes the process by The actor views bookings  **Goal:**  Enable The actor to monitor and review bookings related to birthday party for kids  **Triggers**  The actor decides to view bookings within the system  **Preconditions:**  PRE-1: The actor must be logged in to the system  PRE-2: There must be booking data available within the system.  **Post Conditions:**  The actor successfully views the booking within the system.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor selects the “Booking” tab. | Show a drop down list of options related to Booking   * Booking * Transaction * Comment | | **2** | The actor selects”Booking” | Redirect booking page |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Relationships:**  N/A  **Business Rules:**  Only The actor can View Booking | | | |

### **3.2.42 View Category**

| **USE CASE-042 View Category** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC042 | **Use-case Version** | <1.0> |
| **Use-case Name** | View Category | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Admin  **Summary:**  This use case describes the process by The actor views categories  **Goal:**  Enable The actor to review and manage categories related to birthday party for kids  **Triggers**  The actor decides to view categories within the system  **Preconditions:**  PRE-1: The actor must be logged in to the system  PRE-2: There must be category data available within the system.  **Post Conditions:**  Admin successfully views the category within the system.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor select “ Category “ tab | Show a drop down list of options related to Post   * View category * Add category * Enable/Disable category * Edit category | | **2** | The actor click “View” button | Redirect to view category page that show a list of including   * Events * Special set * Normal set * Booking birthday party at home |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Relationships:**  N/A  **Business Rules:**  Only The actor can View Category | | | |

### **3.2.43 Add Category**

| **USE CASE-043 Add Category** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC043 | **Use-case Version** | <1.0> |
| **Use-case Name** | Add Category | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Admin  **Summary:**  This use case describes the process by The actor adds a new category  **Goal:**  Enable The actor to expand the category options available  **Triggers**  Admin decides to view categories within the system  **Preconditions:**  PRE-1: The actor must be logged in to the system  PRE-2: The actor must have appropriate permissions to add categories  **Post Conditions:**  POST-1: The new category is successfully added to the system.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor select “ Category “ tab | Show a drop down list of options related to Post   * View category * Add category * Enable/Disable category * Edit category | | **2** | The actor click “Add Category” | Redirect to Add Category page | | **3** | The actor clicks the “Add” button at the bottom | System displays a notification you want to edit category:   * Category name: Text * Description: Text * Representative image: PNG,... | | **3** | The actor fills all attributes in the form. | [Exception 1] | | **4** | The actor fill information in the blank and click “Done” button | System displays a notification “you want to add”   * “Yes” * “No” | | **5** | The actor click “Yes” button | System will add and store it in data. |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | At least one input field doesn’t match the constraint. | System shows error message corresponding to the fail constrain |   **Relationships:**  N/A  **Business Rules:**  Only The actor can Add Category | | | |

### **3.2.44 Enable Category**

| **USE CASE-044 Enable Category** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC044 | **Use-case Version** | <1.0> |
| **Use-case Name** | Enable Category | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Admin  **Summary:**  This use case describes the process by The actor enables a category  **Goal:**  Enable The actor to activate previously disabled categories  **Triggers**  The actor decides to enable a category within the system  **Preconditions:**  PRE-1: The actor must be logged in to the system  PRE-2: The category to be enabled must exist within the system and be currently disabled  **Post Conditions:**  POST-1: The selected category is successfully enabled within the system.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor select “ Category “ tab | Show a drop down list of options related to Post   * View category * Add category * Enable/Disable category * Edit category | | **2** | The actor click “Enable/Disable” | Redirect to the Enable/Disable Category page and show a list of categories. | | **3** | The actor clicks the “Enable” button next to the category that they want to enable. | System displays a notification you want to Enable Category   * “Yes” * “No” | | **4** | The actor click “Yes” button | System will disable and store it in data |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Relationships:**  N/A  **Business Rules:**  Only The actor can Enable Category | | | |

### **3.2.45 Disable Category**

| **USE CASE-045 Disable Category** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC045 | **Use-case Version** | <1.0> |
| **Use-case Name** | Disable Category | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Admin  **Summary:**  This use case describes the process by The actor disables a category  **Goal:**  Enable The actorn to deactivate categories that are no longer needed for booking birthday parties for kids.  **Triggers**  The actor decides to disable a category within the system  **Preconditions:**  PRE-1: The actor must be logged in to the system  PRE-2: The category to be disabled must exist within the system and be currently disabled  **Post Conditions:**  The selected category is successfully disabled within the system.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor select “ Category “ tab | Show a drop down list of options related to Post   * View category * Add category * Enable/Disable category * Edit category | | **2** | The actor click “Enable/Disable category” | Redirect to the Enable/Disable category page and show a list of categories. | | **3** | The actor clicks the “Disable” button next to the category they want to enable. | System displays a notification you want to disable category   * “Yes” * “No” | | **4** | The actor click “Yes” button | System will enable and store it in data |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Relationships:**  N/A  **Business Rules:**  Only The actor can Disable Category | | | |

### **3.2.46 Edit Category**

| **USE CASE-046 Edit Category** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC046 | **Use-case Version** | <1.0> |
| **Use-case Name** | Edit Category | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Admin  **Summary:**  This use case describes the process by which The actor edits a category  **Goal:**  Enable The actor to modify the details of existing categories.  **Triggers**  The actor decides to edit a category  **Preconditions:**  PRE-1: The actor must be logged in to the system  PRE-2: The category to be edited must exist within the system.  **Post Conditions:**  The selected category is successfully edited within the system.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor select “ Category“ tab | Show a drop down list of options related to Post   * View category * Add category * Enable/Disable category * Edit category | | **2** | The actor click “Edit category” | Redirect to edit Category page and show a list of categories. | | **3** | The actor clicks the “Edit” button next to the category they want to enable. | System displays a notification you want to edit category:   * Category name: Text * Description: Text * Representative image: PNG | | **3** | The actor fills all attributes in the form. | [Exception 1] | | **5** | The actor clicks the “Done” button. | System displays a notification “you want to edit”   * “Yes” * “No” | | **6** | The actor click “Yes” button | System will edit and store it in data |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor attempts to edit a category but encounters an error | System displays an error message. | | **2** | The actor retries the category editing process | System successfully disables the category. |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | At least one input field doesn’t match the constraint. | System shows error message corresponding to the fail constrain |   **Relationships:**  N/A  **Business Rules:**  Only The actor can Edit Category | | | |

### **3.2.47 Update A Post**

| **USE CASE-047 Update A Post** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC047 | **Use-case Version** | <1.0> |
| **Use-case Name** | Update a post | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Admin  **Summary:**  This use case describes the process of The actor update a post  **Goal:**  Enable The actor to update a post from the system  **Triggers**  The actor decides to update a comment.  **Preconditions:**  PRE-1: The actor must be logged in to the system.  PRE-2: A post must exist in the system.  **Post Conditions:**  POST-1: A post is successfully updated from the system.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor selects the “Post” tab. | Show a drop down list of options related to Post   * Create Post * View List post * Update a post * Delete a post | | **2** | The actor click “Update A Post” button | Redirect to Update A Post page | | **3** | The actor choose a post | The system displays details of the post:   * Title: Text * Content: Text * Image: PNG,... * Author: Text * Comment: Text * Update Time: Time according to the form DD-MM-YY * “Update” button | | **4** | The actor fills all attributes in the form. | [Exception 1] | | **5** | The actor click “ Update” button | System displays a notification you want to update   * “Yes” * “No” | | **6** | The actor click “Yes” button | System will update and store it in data |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | At least one input field doesn’t match the constraint. | System shows error message corresponding to the fail constrain |   **Relationships:**  N/A  **Business Rules:**  Only The actor can Update A Post | | | |

### **3.2.48 Delete A Post**

| **USE CASE-048 Delete A Post** | | | |
| --- | --- | --- | --- |
| **Use-case No.** | UC048 | **Use-case Version** | <1.0> |
| **Use-case Name** | Delete a post | | |
| **Author** | Minh Huy | | |
| **Date** | 29/01/2024 | **Priority** | High |
| **Actor:**  Admin  **Summary:**  This use case describes the process of The actor deleting a post.  **Goal:**  Enable The actor to delete a post from the system.  **Triggers**  The actor decides to delete a comment.  **Preconditions:**  PRE-1: The actor must be logged in to the system.  PRE-2: A post must exist in the system.  **Post Conditions:**  POST-1: A post is successfully deleted from the system.  **Main Success Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | The actor selects the “Post” tab. | Show a drop down list of options related to Post   * Create Post * View List post * Update A Post * Delete A Post | | **2** | The actor click “Delete a post” | Redirect to Delete A Post page and show a list of posts. | | **3** | The actor clicks the “ delete” button next to a post that they want to delete. | System displays a notification you want to delete   * “Yes” * “No” | | **4** | The actor click “Yes” button | System will delete and store it in data |   **Alternative Scenario:**   | **Step** | **Actor Action** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | N/A | N/A |   **Relationships:**  N/A  **Business Rules:**  Only The actor can Delete A Post. | | | |

# **4** **NON-FUNCTIONAL Requirements**

## **4.1** **Usability**

* HappyKid is a children's birthday party booking website with the goal of ordering services quickly and simply. HappyKid has a simple and easy-to-use online interface, helping you book parties quickly and conveniently. It doesn't take much time, you just need a few clicks to organize a wonderful party for your baby!
* HappyKid understands that every party is special and unique. That's why our party booking interface is designed to be flexible and easily adjusted to your child's specific preferences. You can customize the space, theme and various services as you like to create the perfect party.
* Our staff interface is designed to be extremely simple and easy to use. Employees can easily track their work and use the application right after a single training session. This helps optimize work performance and minimize study time.

## **4.2** **Reliability**

* Availability: The booking system should be available for use 100% of the time, allowing customers to make reservations and modifications during operating hours, which are from 8:00 AM to 8:00 PM local time, seven days a week. Maintenance access should be scheduled during off-peak hours, preferably between 12:00 AM to 6:00 AM, with a maximum downtime of 1 hour per month.
* Mean Time Between Failures (MTBF): The MTBF for the booking system should be no less than 10,000 hours, ensuring consistent functionality over an extended period. This should be measured in terms of uninterrupted service without system failures or errors.
* Mean Time To Repair (MTTR): In the event of a system failure, the MTTR should not exceed 30 minutes during operational hours and 1 hour during maintenance periods. Quick restoration of service is crucial to minimize disruption to customers and ensure a seamless booking experience.
* Accuracy: The system's output should maintain a high level of precision and accuracy, with booking details recorded with a resolution of at least one minute. Additionally, accuracy should be maintained according to industry standards for online booking systems, ensuring that reservations are processed correctly and reliably.
* Maximum Bugs or Defect Rate: The maximum allowable bug or defect rate should not exceed 0.1 bugs per thousand lines of code (0.1 bugs/KLOC). Bugs should be categorized as minor, significant, or critical based on their impact on system functionality. A critical bug is defined as any issue resulting in a complete loss of data or the inability to use essential features of the booking system.

## **4.3** **Performance**

Interfaces

* The system should be capable of handling concurrent connections from multiple users without significant performance degradation. This includes both customer-facing interfaces for party booking and internal interfaces for staff management.

Response Time:

* The average response time for processing a party booking request should be no more than 2 seconds.
* The maximum response time for processing a party booking request should not exceed 5 seconds to ensure a smooth user experience.

Throughput:

* The system should support a minimum throughput of 100 transactions per second during peak hours to accommodate high demand without causing delays or timeouts for users.

Capacity:

* The system should be capable of accommodating a minimum of 10,000 customers concurrently without experiencing performance bottlenecks or slowdowns.

Degradation Modes:

* In the event of increased traffic or system overload, the system should gracefully degrade by prioritizing essential functionalities such as booking processing over non-essential tasks like reporting generation.
* During peak usage periods, the system may prioritize real-time booking transactions over administrative tasks to ensure timely service for customers.

Resource Utilization:

* Memory: The system should utilize no more than 80% of available memory resources to maintain optimal performance and prevent memory-related issues.
* Disk: Disk space utilization should be monitored, and the system should maintain at least 20% free disk space at all times to prevent performance degradation due to disk fragmentation or lack of storage.
* Communications: Network bandwidth usage should be optimized to minimize latency and ensure timely data transmission between clients and servers.

## **4.4** **Supportability**

* The request naming conventions to ensure consistency across the codebase. This includes guidelines for formatting, commenting, and organizing code.
* Class Libraries: Utilize established class libraries or frameworks whenever possible to leverage existing solutions and reduce the need for custom code. Document the use of these libraries and their versions for future reference.
* Maintenance Access: Determine who will have access to maintain the system and define their roles and responsibilities. This includes access to source code repositories, databases, and other necessary resources.
* Maintenance Utilities: Identify and implement tools and utilities that facilitate system maintenance, such as debugging tools, performance monitoring tools, and automated testing frameworks.
* Documentation: Document the system architecture, design decisions, and implementation details comprehensively. This documentation should be regularly updated to reflect changes in the system.
* Error Logging and Monitoring: Implement robust error logging and monitoring mechanisms to track system errors and performance issues. This helps in identifying and resolving issues quickly.
* Version Control: Utilize version control systems (e.g., Git) to manage code changes effectively. Ensure that all code changes are versioned, and maintain clear commit messages for better traceability.
* Backup and Recovery: Establish backup and recovery procedures to protect against data loss and system failures. Regularly backup critical data and test the recovery process to ensure its effectiveness.
* Modular Design: Design the system in a modular and extensible manner to facilitate future updates and enhancements. Use design patterns and architectural principles to achieve modularity.
* Testing and Quality Assurance: Implement thorough testing processes, including unit tests, integration tests, and system tests, to ensure the reliability and correctness of the system. Automate testing wherever possible to streamline the maintenance process.
* Security Considerations: Integrate security measures into the system design and implementation to protect against security threats and vulnerabilities. This includes authentication, authorization, data encryption, and secure communication protocols.
* Performance Optimization: Monitor and optimize system performance regularly to ensure optimal operation. This may involve profiling the code, identifying performance bottlenecks, and implementing optimizations.

## **4.5** **Design Constraints**

Software Language:

* Use web programming languages like JavaScript, HTML, and CSS for the user interface.
* The database will be built using SQL to manage party booking and user information.

Software Process Requirements:

* Apply an agile software development model like Scrum to optimize interaction between development teams and ensure flexibility in adapting to changing requirements.
* Follow continuous testing processes to ensure product quality.

Rules for Using Development Tools:

* Use source code management tools like Git to track and control source code versions.
* Use integrated development environments like Jenkins to automate deployment and testing processes.

Architectural and Design Constraints:

* The system will have to adhere to a multi-tier architecture to separate business logic, user interface, and database.
* Use the RESTful API design model to optimize integration with other services.

Class Library:

* Use popular class libraries like Bootstrap or Material-UI to optimize user interface design and ensure consistency across devices.
* Use an online calendar and party booking processing library to reduce development time.

External Service Integration:

* Connect to online payment services like PayPal or Stripe to process payments.
* Use mapping APIs like Google Maps to assist with location determination and directions.

Security and Privacy:

* Uses a strong encryption library to protect users' personal information.
* Comply with web security standards like OAuth to manage access rights.

## **4.6** **On-line User Documentation and Help System Requirements**

### **4.6.1 Overview**

The Kids Birthday Party Booking System shall provide comprehensive online user documentation and a help system designed to assist users in navigating and utilizing the system efficiently. This documentation will be accessible through the system's user interface and is intended to provide immediate assistance and guidance to users, ensuring a smooth and user-friendly experience.

### **4.6.2 Objectives**

* Accessibility: Ensure that help and documentation are easily accessible from anywhere within the system.
* Comprehensiveness: Cover all features and functionalities of the system.
* Clarity: Present information in a clear, concise, and user-friendly manner.
* Searchability: Enable users to quickly find solutions to their queries or problems.
* Up-to-Date Information: Ensure that the documentation is regularly updated to reflect the latest features and functionalities of the system.

### **4.6.3 Requirements**

#### 4.6.3.1 Help Button

* A "Help" button must be prominently displayed on every page of the system.
* Clicking the "Help" button shall open the online documentation in a separate, easily navigable window or sidebar, ensuring that the user's current work is not disrupted.

#### 4.6.3.2 Search Functionality

* The online documentation shall include a search bar at the top of the help window or sidebar, allowing users to enter keywords or phrases related to their query.
* The search function shall provide relevant results, including topics, frequently asked questions (FAQs), and step-by-step guides related to the entered keywords.

#### 4.6.3.3 Content Requirements

The online documentation shall include, but not be limited to, the following content:

* Getting Started Guide: Instructions on how to set up an account, navigate the system, and start booking birthday parties.
* Feature Overviews: Detailed descriptions of all features, including booking processes, customization options for parties, payment methods, and cancellation policies.
* FAQs: A section dedicated to answering the most common questions users may have.
* Troubleshooting Guide: Assistance with common issues, such as login problems, booking errors, and payment issues.
* Contact Information: Information on how to contact customer support for further assistance, including email, phone, and live chat options.

#### 4.6.3.4 Help About Notices

* Every page or feature within the system shall include a "Help About" link or button that provides specific information or instructions related to that page or feature.
* This information shall include a brief description of the page/feature, how to use it, and where to find more detailed information in the online documentation.

#### 4.6.3.5 User Feedback

* The system shall include a mechanism for users to provide feedback on the online documentation and help system, allowing for continual improvement based on user experience and suggestions.

### **4.6.4 Maintenance and Updates**

* The online documentation and help system must be maintained and updated regularly to ensure accuracy and relevance. This includes updating screenshots, instructions, and FAQs to reflect changes in the system’s interface or functionality.
* Updates to the documentation shall coincide with system updates or as frequently as necessary to ensure the highest level of user support.

### **4.6.5 Performance Requirements**

* The online documentation and help system shall be optimized for fast loading times, ensuring that users can access information without significant delays.
* It should be compatible with all major browsers and responsive to different screen sizes, including tablets and smartphones.

### **4.6.6 Security Requirements**

* Access to the online documentation and help system shall not compromise the security of the user’s session or personal information.
* Any external links provided in the documentation shall be verified as secure and relevant.

## **4.7** **Purchased Components**

Additionally, the system may utilize licensed software components or third-party services to enhance functionality and streamline operations. The following purchased components may be integrated:

* Email Service Provider: Integration with an email service provider such as SendGrid or Mailchimp to facilitate communication with users, including sending booking confirmations, reminders, and promotional messages. Any associated licensing agreements and compatibility considerations will be addressed during integration.
* Customer Relationship Management (CRM) Software: Adoption of a CRM solution like Salesforce or HubSpot to manage customer interactions, track leads, and analyze customer data. Integration with the CRM system will require adherence to compatibility standards and may involve licensing agreements.
* Analytics Platform: Incorporation of an analytics platform like Google Analytics or Mixpanel to gather insights into user behavior, monitor system performance, and optimize marketing strategies. Integration will involve adherence to interface standards and potentially licensing agreements.
* Content Delivery Network (CDN): Utilization of a CDN service such as Cloudflare or Amazon CloudFront to improve the performance and reliability of content delivery, particularly for multimedia elements. Compatibility and interoperability considerations will be addressed during integration, along with any applicable usage restrictions.
* Customer Support Ticketing System: Integration with a customer support ticketing system like Zendesk or Freshdesk to streamline customer inquiries, issue resolution, and feedback management. Licensing agreements and interface standards will be taken into account during integration.
* Multilingual Support Service: Implementation of a multilingual support service such as Weglot or Bablic to facilitate localization and translation of the user interface and content. Integration will involve adherence to compatibility standards and may require licensing agreements.

Each purchased component will be carefully evaluated to ensure compatibility with existing system architecture, adherence to relevant standards, and compliance with licensing agreements. Interoperability with other system components will be verified during integration testing to guarantee seamless functionality and optimal performance.

## **4.8** **Interfaces**

### **4.8.1 User Interfaces**

* The system will provide a web-based interface for users to interact with the HappyKid website. This includes easy-to-use navigation to explore available party packages, view details of party options, party booking functionality, manage user profiles
* Accessibility standards will be followed to ensure the system is usable by all users, including those with disabilities. The interface will be responsive, ensuring compatibility across devices and screen sizes.
* The admin UI will allow system administrators to manage partners, users, system settings, and view transaction reports for HappyKid events.

### **4.8.2 Hardware Interfaces**

* As the HappyKid is predominantly a software solution with no direct hardware interaction, standard computing devices (PCs, laptops, tablets, and smartphone) are the primary hardware interfaces. These devices require standard web browsers (e.g.,Chrome, Firefox, Safari) for accessing the HappyKid
* Integration with hardware for payment processing (e.g., card readers) will be managed via third-party payment gateway APIs, without requiring direct hardware interface development by the HappyKid

### **4.8.3 Software Interfaces**

* Payment Gateway: Integration with a secure payment gateway to facilitate online transactions securely.
* Email Service: Integration with an email service provider for sending booking confirmations, reminders, and promotional emails to customers.
* Database Management System (DBMS): Utilization of a relational database management system (e.g., MySQL, PostgreSQL) for storing and retrieving booking data efficiently.
* Content Management System (CMS): Integration with a CMS for managing website content, including party package details, images, and FAQs.

### **4.8.4 Communications Interfaces**

* The application will communicate with external systems and devices using standard internet protocols such as HTTP and HTTPS. This includes communication with payment gateways for processing transactions securely over the internet. Additionally, the application will utilize email protocols (e.g., SMTP) for sending and receiving emails related to bookings and customer inquiries. All communication will adhere to industry-standard encryption and security protocols to ensure data confidentiality and integrity.

## **4.9** **Licensing Requirements**

* Type of License: Specify the type of license that applies to the software, such as open-source licenses (e.g., MIT, GPL, Apache) or proprietary licenses. Each type of license comes with different usage rights and restrictions.
* Distribution Restrictions: Define any restrictions on the distribution of the software. This may include limitations on redistribution, sublicensing, or resale of the software.
* Usage Restrictions: Define any restrictions on how the software can be used. For example, some licenses may restrict the use of the software for commercial purposes or limit the number of users who can access the software.
* Modification Restrictions: Specify whether the software can be modified and under what conditions. Some licenses require modifications to be shared with the community or distributed under the same license terms.
* Attribution Requirements: Determine whether the software requires attribution to the original authors or contributors. This may include displaying copyright notices or acknowledgments within the software.
* License Compatibility: Ensure that the chosen license is compatible with any third-party libraries or components used in the software. Incompatibility between licenses can lead to legal issues.
* License Enforcement Mechanisms: Define mechanisms for enforcing compliance with the licensing requirements. This may include implementing license key validation, usage tracking, or other technical measures.

## **4.10** **Legal, Copyright, and Other Notices**

Copyright Notice:

* The HappyKid software, along with all its components including but not limited to code, design elements, graphics, and documentation, is copyrighted © [Year] by [Your Company Name]. All rights reserved.

Trademark Notice:

* The HappyKid name and logo are trademarks or registered trademarks of [Your Company Name] in [Country/Region]. Unauthorized use of these trademarks is strictly prohibited.

Patent Notice:

* Certain features of the HappyKid software may be protected by patents owned by [Your Company Name]. Other patents may also apply.

Disclaimer:

* The HappyKid software is provided "as is" without any warranty, express or implied, including but not limited to warranties of merchantability, fitness for a particular purpose, or non-infringement. [Your Company Name] shall not be liable for any damages arising out of the use or inability to use the software.

Limitation of Liability:

* In no event shall [Your Company Name] be liable for any special, incidental, indirect, or consequential damages whatsoever arising out of or in connection with the use or performance of the HappyKid software, even if advised of the possibility of such damages.

Third-Party Components:

* The HappyKid software may include third-party software components subject to their respective licenses. Please refer to the documentation accompanying those components for license information and terms of use.

Compliance:

* Users of the HappyKid software are responsible for ensuring compliance with all applicable laws, regulations, and licensing requirements related to its use and distribution.

Acknowledgment:

* By using the HappyKid software, you acknowledge that you have read, understood, and agreed to be bound by these legal notices and terms of use.

For any inquiries regarding legal matters or licensing, please contact [Your Company Legal Department or Representative].

## **4.11** **Applicable Standards**

[This section describes by reference any applicable standard and the specific sections of any such standards which apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, and so forth.]

# **5** **Supporting Information**