|  |  |  |
| --- | --- | --- |
| **Use case name:** | Checking the validity of the QR code visitor ticket | |
| **Scenario:** | Checking the validity of the QR Code to identify visitor’s eligibility | |
| **Triggerring event:** | Visitor wants to enter under the sea through the entrance gate | |
| **Brief description:** | The visitor wants to enter the under the sea through the entrance gate, then the officer asks for a visitor's ticket to scan the QR code on the reader machine to check validity of the ticket. | |
| **Actors:** | Entrance Gate Employee | |
| **Related use cases:** | - | |
| **Stakeholders:** | Attraction Department | |
| **Preconditions:** | * List ticket unique QR code already exists in subsystems * Available ticket visitor with unique QR code * Available reader machine integrated with subsystems | |
| **Postconditions:** | Display message validity in systems | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Entrance gate employee scan QR Code visitor ticket | * 1. System get database eligible ticket   2. System read the unique code   3. Show the message about notify validating. |
| **Exception Conditions:** | * 1. if QR code isn’t scanning, the entrance gate employee must enter the code manually | |

|  |  |  |
| --- | --- | --- |
| **Use case name:** | Recording the ticket transaction | |
| **Scenario:** | Recording the ticket transaction to the systems | |
| **Triggerring event:** | When visitor want to buy ticket and the selling ticket input the data transaction | |
| **Brief description:** | The sellling ticket employee input the data transaction about quantity the the ticket which bought by the visitor. | |
| **Actors:** | Selling Ticket Employee | |
| **Related use cases:** | Generating ticket QR Code | |
| **Stakeholders:** | Attraction Department, Financial and Accounting Department | |
| **Preconditions:** | * Available subsystem ticket sales transaction * Available subsystem payment * Quantity of ticket which bought by visitor must exists * Amount of money paid by visitor must exists | |
| **Postconditions:** | * Create data ticket sales transaction * Create data payment | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Selling ticket employee open form ticket sales transaction 2. Selling ticket employee input the number of tickets to be bought 3. Input the amount of money paid by visitors | * 1. Create ticket sales transaction   2. Display form ticket sales transaction   3. Relocate the number of tickets to be bought   4. Calculate total payment   5. Display total payment   6. Calculate the change of payment   7. Create ticket transaction |
| **Exception Conditions:** | 3.2 If quantity field empty or not numeric , so show error message  4.1 If payment amount of money paid field empty, so show error message | |

|  |  |  |
| --- | --- | --- |
| **Use case name:** | Making payment of order restaurant | |
| **Scenario:** | Making payment of order restaurant to the system | |
| **Triggerring event:** | When visitor want pay their order menu in the restaurant | |
| **Brief description:** | The receptionist input payment data of order transaction to the system based number table from visitor | |
| **Actors:** | Receptionist | |
| **Related use cases:** | - | |
| **Stakeholders:** | Restaurant Department, Financial and Accounting Department | |
| **Preconditions:** | * Available subsystem order restaurant payment * Avalable subsystem order transaction restaurant * Number table must exists * Amount of money paid must exists | |
| **Postconditions:** | Create data payment order restaurant | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Recepsionist open form order restaurant payment 2. Receptionist input the number table that visitor want to pay 3. Receptionist Input the amount of money paid by visitors | * 1. Get list order transaction restaurant   2. Display form order restaurant payment   3. Checking to match order transaction restaurant with table number   4. Calculate the total payment(number table)   5. Display the detail and total payment(number table)   6. Calculate the change of payment   7. Relocate current date   8. Save order payment restaurant   9. Display orderPaymentRestaurantID |
| **Exception Conditions:** | 2.1 If the number table not filled, so show message error  2.1 If the number table not found, the receptionist must search manually the order  31. If the amount of money paid by visitor field empty, then show message error | |

|  |  |  |
| --- | --- | --- |
| **Use case name:** | Adding feedback and rate hotel | |
| **Scenario:** | Adding feedback and rate hotel to the system | |
| **Triggerring event:** | When visitors want to tell their feedback and rating about service in hotel | |
| **Brief description:** | The visitor can input their feedback and the rate about service in hotel through a screen | |
| **Actors:** | Visitor | |
| **Related use cases:** | - | |
| **Stakeholders:** | Restaurant Department | |
| **Preconditions:** | * Available subsystem feedback rate hotel | |
| **Postconditions:** | Create data feedback rate hotel | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Visitor open form feedback rate hotel 2. Visitor input data feedback and rating hotel | * 1. Get visitor data   2. Display form feedback hotel   3. Relocate data feedback and rating hotel   4. Save data feedback and rating hotel   5. Display feedbackRateHotelID |
| **Exception Conditions:** | 3.1 if any field is empty. Then show error message | |

|  |  |  |
| --- | --- | --- |
| **Use case name:** | Requesting stock of ingredients | |
| **Scenario:** | Requesting stock of ingredients to the system | |
| **Triggerring event:** | When the chef want to restock the ingredients | |
| **Brief description:** | The chef can input request stock of ingredient when they think the ingredientsneeds to be stock | |
| **Actors:** | Chef | |
| **Related use cases:** | Sending purchase request to purchasing department | |
| **Stakeholders:** | Kitchen Department, Purchasing Department, Financial & Acconting Department | |
| **Preconditions:** | * Available subsystem ingredients * Available subsystem request stock * The ingredient going stock out | |
| **Postconditions:** | Create data request stock | |
| **Flow of activities:** | **Actor** | **System** |
| 1. Viewing the ingredients stock 2. Chef request stock that ingredient going run out | * 1. Display the data stock ingredients   2. System create request stock   3. System returns request stock detail |
| **Exception Conditions:** | 2.1 If any field is empty. Then show error message | |