|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **123Use Case ID** | UC.01 | | | | |
| **Use Case Name** | Đặt bàn và gọi món | | | | |
| **Created by** | Huỳnh Vũ | | **Last updated by** | | Huỳnh Vũ |
| **Date Created** | Dec 01, 2024 | | **Date last updated** | | Dec 01, 2024 |
| **Actors** | Khách hàng (Customer) | | | | |
| **Brief Description** | Khách hàng có thể quét mã QR để đặt bàn và chọn món ăn. Sau khi quét mã QR, hệ thống sẽ tự động cập nhật trạng thái bàn đã được đặt | | | | |
| **Goal** | Đảm bảo khách hàng đặt bàn và gọi món một cách tiện lợi, đồng thời hệ thống cập nhật trạng thái bàn ngay lập tức để tránh trùng lặp | | | | |
| **Trigger** | Khi khách hàng quét mã QR để truy cập vào hệ thống | | | | |
| **Pre-conditions** |  Hệ thống đã có thông tin bàn trống và menu món ăn   Khách hàng có thiết bị quét mã QR và truy cập được vào hệ thống | | | | |
| **Post-conditions** | * Trạng thái bàn được cập nhật là "Đang được đặt". * Đơn đặt món được lưu trữ trong hệ thống. * Khách hàng nhận được xác nhận qua giao diện hệ thống | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Customer arrives at the restaurant and scans the QR code at the table | | The system checks whether the table is in a reserved status | |
|  |  | | The system identifies the table number and displays a welcome message along with the menu | |
| **2** | Customer reviews the menu displayed on their mobile device | | The system provides detailed menu items, including descriptions, prices, and customization options | |
| **3** | Customer selects the desired dishes | | The system updates the order summary in real-time, showing selected items | |
| **4** | Customer confirms their order | | The system processes the order, generates an order ID | |
| **5** | Customer receives their order | |  | |
| **Alternative Flow** | **Luồng phụ 1: Bàn đã được đặt** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Khách hàng quét mã QR để truy cập | | Hệ thống kiểm tra trạng thái bàn và phát hiện bàn đã được đặt | |
| **2** |  | | Hệ thống hiển thị thông báo: "Bàn này đã được đặt. Vui lòng chọn bàn khác hoặc liên hệ nhân viên để được hỗ trợ." | |
| **3** | Khách hàng chọn một bàn khác | | Hệ thống kiểm tra trạng thái bàn mới (Quay lại bước 2 với luồng chính) | |
| **Luồng phụ 2: Món ăn hết hàng hoặc không sẵn có** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Khách hàng quét mã QR để truy cập | | Hệ thống hiển thị giao diện đặt bàn và tự động cập nhật trạng thái bàn tương ứng là "Đang được đặt" | |
| **2** | Khách hàng chọn món ăn từ menu | | Hệ thống kiểm tra trạng thái món ăn và phát hiện món ăn đã hết hàng hoặc không sẵn có. | |
| **3** |  | | Hệ thống hiển thị thông báo: "Món này hiện đã hết. Vui lòng chọn món khác." | |
| **4** | Khách hàng chọn một món khác từ menu | | Hệ thống cập nhật danh sách món trong giỏ hàng (và tiếp tục bước 2 luồng chính) | |
| **Exception Flow** | **Luồng ngoại lệ 1: Quá thời gian đặt bàn** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Khách hàng quét mã QR để truy cập | | Hệ thống hiển thị giao diện đặt bàn và tự động cập nhật trạng thái bàn tương ứng là "Đang được đặt" | |
| **2** | Khách hàng chọn món ăn từ menu | | Hệ thống hiển thị chi tiết món ăn và yêu cầu khách hàng xác nhận đơn hàng | |
| **3** |  | | Hệ thống kiểm tra và xử lý yêu cầu trong khoảng thời gian quy định. | |
| **5** | Thời gian xử lý quá lâu (hết thời gian chờ). | | Hệ thống hiển thị thông báo: "Yêu cầu của bạn không được xử lý kịp thời. Vui lòng thử lại sau." | |
| **6** | Khách hàng thử lại đặt lại | |  | |
| **Priority** | High | | | | |
| **Business Rule** | Mã QR duy nhất cho mỗi bàn  Cập nhật trạng thái bàn ngay khi đặt  Tối thiểu và tối đa số người trên mỗi bàn  Đặt lại bàn khi khách hàng rời đi  Hủy đặt bàn | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.02 | | | | |
| **Use Case Name** | Cancel Ordered Dish | | | | |
| **Created by** | Huỳnh Vũ | | **Last updated by** | | Huỳnh Vũ |
| **Date Created** | Dec 01, 2024 | | **Date last updated** | | Dec 01, 2024 |
| **Actors** | Customer, System,Waiter | | | | |
| **Brief Description** | This use case allows the customer to cancel a dish they have ordered, provided the dish has not yet entered the "in preparation" stage. The system checks the dish's status and, if eligible, processes the cancellation request | | | | |
| **Goal** | To allow the customer to cancel an order they have placed, provided the dish has not been processed for cooking | | | | |
| **Trigger** | The customer selects the "Cancel Order" option in the system after realizing they no longer wish to proceed with their order, and the dish has not yet started the cooking process | | | | |
| **Pre-conditions** |  The customer has already placed an order for food through the system   * The dish is not yet in the "in preparation" stage | | | | |
| **Post-conditions** | * The selected dish is removed from the customer's order. * The status of the dish is updated to "Cancelled" in the system * If the cancellation was successful, the customer is notified of the successful cancellation | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | **Customer** selects the "Cancel Order" option from the system | | **System** displays a list of the customer's current orders | |
| **2** | **Customer** selects the dish they want to cancel from the list | | System to display a notification about the item you want to cancel | |
| **3** | Customer confirm the cancelation | | System to check if the dish is in the preparation stage | |
| **4** |  | | **System** removes the selected dish from the order and updates the status of the dish to "**Cancelled**" and **System** notifies the customer | |
| **5** | **Customer** exits the cancellation screen or proceeds with their remaining order | |  | |
| **Alternative Flow** | **Alternative Flow 1: The dish has already been prepared** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | **Customer** selects the "**Cancel Order**" option | | **System** displays the list of the customer's current orders | |
| **2** | **Customer** selects the dish they want to cancel | | **System** checks the status of the selected the dish is already in the "**In preparation**" or "**Prepared**" stage | |
| **3** |  | | **System** notifies the customer: "**The dish has already been prepared and cannot be canceled.**" | |
| **4** | **Customer** acknowledges the notification and decides to keep the order | |  | |
| **Exception Flow** | **Exception Flow 1: Contact the restaurant directly** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | The customer meets the restaurant staff in person to cancel the order | | System notification that the item has been canceled by an authorized user | |
| **2** | Customer comfirm success | |  | |
| **Priority** | High | | | | |
| **Business Rule** | A dish can only be canceled before preparation  Cancellation must be confirmed by the system  Product cancellation notice to customers  Cancelling an item does not affect other items in the order  The dish can only be canceled if it is not prepared. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.03 | | | | |
| **Use Case Name** | Pay Bill | | | | |
| **Created by** | Huỳnh Vũ | | **Last updated by** | | Huỳnh Vũ |
| **Date Created** | Dec 01, 2024 | | **Date last updated** | | Dec 01, 2024 |
| **Actors** | Customer, Cashier, System | | | | |
| **Brief Description** | Customers can pay their bills quickly and conveniently right at the dining table through the application that has scanned the login code. | | | | |
| **Goal** | To ensure customers can pay their bills quickly, accurately, and securely right at their table | | | | |
| **Trigger** | Customer clicks on payment on the app | | | | |
| **Pre-conditions** |  Customer requests payment   The invoice has been printed or displayed on the screen | | | | |
| **Post-conditions** | * The bill has been paid * Order status has been updated * Customer receives invoice | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Customer selects "Pay Now" on their mobile device or requests payment from the cashier | | The system displays payment options | |
| **2** | Customer chooses a payment method and confirms the amount | | The system displays the payment interface corresponding to the selected payment method | |
| **3** | Customer selects a payment method | | The system validates the payment method | |
|  | **4** | Customer enters payment details | | The system validates the payment transaction | |
|  | **5** |  | | System notification payment success | |
|  | **6** | Cashier confirms receipt of payment | | The system updates the order status to "**Paid**," | |
| **Alternative Flow** | **Alternative Flow 1: Payment fails due to insufficient funds** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** |  | | System confirms payment fails due to insufficient funds | |
| **2** | Customer selects a different payment method | | Return flow 3 | |
| **Exception Flow** | **Exception Flow 1: Payment system connection failure** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | The customer has a free meal voucher and enters the correct voucher code | | System confirms successful payment | |
| **2** | Cashier confirms receipt of payment | | The system updates the order status to "**Paid**," | |
| **Priority** | High | | | | |
| **Business Rule** | Payment confirmation  Multiple payment methods  Payment timeout  Order status update  Receipt issuance | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.04 | | | | |
| **Use Case Name** | Submit Feedback and Report Issues | | | | |
| **Created by** | Huỳnh Vũ | | **Last updated by** | | Huỳnh Vũ |
| **Date Created** | Dec 01, 2024 | | **Date last updated** | | Dec 01, 2024 |
| **Actors** | Customer, System | | | | |
| **Brief Description** | Allows the customer to submit feedback or report an issue regarding their experience at the restaurant. The system records the feedback or issue and notifies the appropriate staff for resolution | | | | |
| **Goal** | To enable customers to provide feedback or report issues to improve service quality and address problems promptly | | | | |
| **Trigger** | The customer selects the “Submit Feedback” or “Report Issue” option in the system | | | | |
| **Pre-conditions** |  The customer has completed an order or service experience   * The customer has access to the feedback or issue reporting function in the system | | | | |
| **Post-conditions** | * The feedback or issue is recorded in the system. * The customer receives a confirmation that their feedback or report has been submitted * The system notifies the appropriate staff to address the feedback or issue | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Customer selects “Submit Feedback” or “Report Issue” from the menu | | The system displays a form for submitting feedback or reporting an issue | |
| **2** | Customer enters feedback or issue details and any relevant information | | The system checks whether the customer information is left blank | |
| **3** |  | | The system allows sending, and the "Send" button turns blue | |
| **4** | Customer submits the form by clicking “Send” | | The system saves the feedback or issue report and displays a confirmation message" **Phản hồi đã được gửi thành công**" | |
| **5** | Customer comfirm successful | | The system notifies the relevant staff about the feedback or issue | |
| **Alternative Flow** | **Alternative Flow 1: The customer has not filled in all the required information** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** |  | | The system checks if the customer has not filled in all the required information | |
| **2** |  | | The system returns to step 2, and the customer is required to re-enter their feedback | |
| **3** | Customer comfirm the system notification | |  | |
| **Exception Flow** | **Exception Flow 1: System error** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Customer selects “Submit Feedback” or “Report Issue” from the menu | | The system displays a form for submitting feedback or reporting an issue | |
| **2** | Customer enters feedback or issue details and any relevant information | | The system captures the entered information | |
| **3** | Customer decides not to submit feedback and clicks “**Cancel**.” | | The system cancels the submission and returns to the previous screen | |
| **Priority** | High | | | | |
| **Business Rule** | Notification of issue resolution  Feedback retention | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.05 | | | | |
| **Use Case Name** | Timekeeping | | | | |
| **Created by** | Huỳnh Vũ | | **Last updated by** | | Huỳnh Vũ |
| **Date Created** | Dec 01, 2024 | | **Date last updated** | | Dec 01, 2024 |
| **Actors** | Staff,System | | | | |
| **Brief Description** | This use case allows staff to log their working hours by clocking in and out through the system. The system records the timestamps and updates the staff attendance log accordingly. | | | | |
| **Goal** | To accurately track and record the working hours of all staff | | | | |
| **Trigger** | The staff selects the "Clock In" or "Clock Out" option in the system at the beginning or end of their shift | | | | |
| **Pre-conditions** | · The staff is registered in the system  · The system is operational and accessible | | | | |
| **Post-conditions** | * The clock-in or clock-out time is recorded in the system * The staff attendance log is updated with the recorded time * The staff receives confirmation of the successful clock-in or clock-out | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Staff logs into the system | | The system displays the timekeeping interface | |
| **2** | Staff selects “**Clock In**” at the start of their shift. | | The system accesses the database to check the stored data | |
| **3** |  | | The system records the clock-in time and displays a confirmation message | |
| **4** | Staff works their shift | | The system verifies the information and starts processing the payment | |
|  | **5** | Staff selects “**Clock Out**” at the end of their shif | | The system checks whether the employee has completed their timecard entry. | |
|  | **6** |  | | The system records the clock-out time and updates the attendance log | |
|  | **7** | Staff receive success notification | |  | |
| **Alternative Flow** | **Alternative Flow 1: All staff forgets to clock out** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** |  | | The system detects a missing clock-out time and sends a notification to the staff | |
| **2** | Staff requests a manual clock-out adjustment | | The system logs the request and notifies the manager for approval | |
| **Alternative Flow 2: The employee could not be found in the data** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** |  | | The system checks and does not find the employee's data | |
| **2** |  | | The system returns to the current step. | |
| **3** | Staff try again login | |  | |
| **Exception Flow** | **Exception Flow 1:** **Clocking out is done from an account with a higher rank or privilege level** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** |  | | Clocking out is done from an account with a higher rank or privilege level | |
| **2** |  | | The system successfully records the time entry | |
| **Priority** | High | | | | |
| **Business Rule** | Valid login  Clock-in before shift start  Clock-out after shift end  Duplicate clock-in/out prevention | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.07 | | | | |
| **Use Case Name** | Confirm Food Order | | | | |
| **Created by** | Huỳnh Vũ | | **Last updated by** | | Huỳnh Vũ |
| **Date Created** | Dec 01, 2024 | | **Date last updated** | | Dec 01, 2024 |
| **Actors** | Kitchen Staff, System | | | | |
| **Brief Description** | This use case allows kitchen staff to confirm the receipt of a food order for preparation. Once confirmed, the system updates the order status | | | | |
| **Goal** | To ensure that food orders are acknowledged and prepared efficiently | | | | |
| **Trigger** | The system sends a new food order notification to the kitchen staff | | | | |
| **Pre-conditions** |  The kitchen staff is logged into the system   The system is operational and able to receive food orders | | | | |
| **Post-conditions** | * The order status is updated to "**In Preparation**" * Relevant staff are notified of the updated status | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Kitchen staff logs into the system | | The system displays the dashboard showing all new food orders | |
| **2** | Kitchen staff selects a new food order to view the details | | System check whether orders must be in "Pending" status | |
|  |  | | The system displays the order details, including dish name, quantity, note each dish | |
| **3** | Kitchen staff confirms receipt of the order by clicking "**Confirm Order**" | | The system updates the order status to "In Preparation" and notifies the service staff | |
| **4** | Kitchen staff begins preparing the dish | |  | |
| **Alternative Flow** | **Alternative Flow 1: Kitchen staff declines the order** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** |  | | The system checks if the order has been confirmed by another employee | |
| **2** |  | | The system returns to the menu page and notifies that the order has already been confirmed by someone else | |
| **4** | Staff comfirm the dish | |  | |
| **Exception Flow : If the employee cancels a menu that has already been confirmed** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | If the employee cancels a menu that has already been confirmed | | The system displays the order details, including dish name, quantity, note each dish | |
| **2** | The kitchen staff comfirm | |  | |
| **Priority** | High | | | | |
| **Business Rule** | Order confirmation  Order notification Timeliness  Order status updates  Handling declined orders  Network onnectivity requirement | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.08 | | | | |
| **Use Case Name** | Confirm Dish Status | | | | |
| **Created by** | Huỳnh Vũ | | **Last updated by** | | Huỳnh Vũ |
| **Date Created** | Dec 01, 2024 | | **Date last updated** | | Dec 01, 2024 |
| **Actors** | Kitchen Staff, System | | | | |
| **Brief Description** | This use case allows kitchen staff to confirm the current status of a dish, such as "In Preparation," "Ready to Serve," or "Delayed." The system records and updates the status to keep other stakeholders informed | | | | |
| **Goal** | To ensure that the status of dishes is accurately tracked and communicated in real-time | | | | |
| **Trigger** | The kitchen staff selects a dish from the order list to update its status | | | | |
| **Pre-conditions** |  The kitchen staff is logged into the system   The dish is part of an active order | | | | |
| **Post-conditions** | * The system updates the status of the dish and notifies relevant staff * The dish status is visible to both kitchen and serving staff | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Kitchen staff logs into the system and views the active order list | | The system displays the list of dishes with their current statuses | |
| **2** | Kitchen staff selects a dish to confirm its status | | The system checks whether the dish is in the "preparing" status | |
|  |  | | The system displays the dish details and available status options | |
| **3** | Kitchen staff selects the appropriate status | | The system updates the dish status and notifies serving staff | |
| **Alternative Flow 1: Dish status conflict** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** |  | | The system checks for conflicts in updating the dish status | |
| **2** |  | | The system returns to the menu page and it notifies that there is a conflict | |
| **3** | Kitchen staff reviews the conflict notification and verifies the status with the manager | |  | |
| **Exception Flow** | **Exception Flow 1: Network connectivity issue** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** |  | | The system attempts to update the status but fails due to lost network connectivity | |
| **2** | Kitchen staff manually records the dish status and reports the issue to the manage | | The manager records the status change manually and updates the system once connectivity is restored. | |
| **Priority** | High | | | | |
| **Business Rule** | Status options  Timely status updates  Accurate notifications  Conflict resolution  Network requirement  Status synchronization | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.09 | | | | |
| **Use Case Name** | Track Work Schedule | | | | |
| **Created by** | Huỳnh Vũ | | **Last updated by** | | Huỳnh Vũ |
| **Date Created** | Dec 01, 2024 | | **Date last updated** | | Dec 01, 2024 |
| **Actors** | Employee, System | | | | |
| **Brief Description** | This use case allows employees to independently view their work schedules, including assigned shifts, start and end times, and any updates or changes. Employees can track their schedules without managerial intervention | | | | |
| **Goal** | To enable employees to access accurate and updated personal work schedules | | | | |
| **Trigger** | An employee logs into the system to check their schedule | | | | |
| **Pre-conditions** |  The employee is registered in the system   The work schedule has been entered into the system | | | | |
| **Post-conditions** | * The employee has successfully viewed their schedule * Any schedule updates are highlighted for the employee to acknowledge | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Employee logs into the system | | The system displays the employee's dashboard, including their work schedule | |
| **2** | Employee selects the "**Work Schedule**" option | | The system displays the employee’s schedule, including assigned shifts, dates, and times | |
| **4** | Employee reviews their schedule | | The system highlights any updates or changes schedule | |
| **3** | Staff press exit to display main menu | | The system displays the main menu as if you had just logged in | |
| **Alternative Flow** | **Alternative Flow 1: Viewing past schedules** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Employee logs into the system | | The system displays the employee's dashboard, including their work schedule | |
| **2** | Employee selects the "**Work Schedule**" option | | The system displays the employee’s schedule, including assigned shifts, dates, and times | |
| **4** | Employee selects the "View Past Schedules" option | | The system prompts the employee to select a date range | |
|  | Employee selects the desired date range and submits the request | | The system retrieves and displays the employee’s past schedules for the selected date range | |
| **3** | Staff selects return to return to current schedule | | Current calendar display system | |
| **Alternative Flow 2**: **Notifications of schedule changes** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | The system sends a notification about a schedule update | | The notification appears on the employee’s dashboard and mobile device | |
| **2** | Employee logs into the system | | The system displays the employee's dashboard, including their work schedule | |
| **3** | Employee selects the "**Work Schedule**" option | | The system displays the employee’s schedule, including assigned shifts, dates, and times | |
| **4** |  | | The system highlights the updated shifts or schedules | |
| **5** | Employee acknowledges the updated schedule | | The system logs the acknowledgment and updates the record | |
| **Alternative Flow 3**: **Employee temporarily not going to work** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Employee logs into the system | | The system displays the employee's dashboard, including their work schedule | |
| **2** | Employee notifies the system of temporarynot going to work | | The system forwards the notice to the manager and updates the schedule accordingly | |
| **3** | The manager reviews and approves the unavailability request | | The system notifies the employee of the decision and updates the records | |
| **Exception Flow** | **Exception Flow 1: System error while loading schedule** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 1 | Employee logs into the system | | Employee logs into the system | |
| **2** | Employee selects the "**Work Schedule**" option | | The system encounters an error and fails to load the schedule | |
| **3** |  | | The system says "Error while viewing calendar, please try again" | |
| 4 | Employee retries accessing the schedule or contacts the manager support for assistance | | The manager records the status change manually and updates the system once connectivity is restored. | |
| **Exception Flow 2**: **Schedule update not acknowledged** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Employee logs into the system | | The system displays the employee's dashboard, including their work schedule | |
| **2** | The system sends a schedule update notification to the employee | | The employee does not acknowledge the update within the required timeframe | |
| **3** | The system escalates the issue to the manager for follow-up | | The manager contacts the employee directly to confirm the schedule update | |
| **Priority** | High | | | | |
| **Business Rule** | Timely status updates  Accurate notifications  Conflict resolution  Network requirement  Status synchronization | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.010 | | | | |
| **Use Case Name** | Track Attendance Schedule | | | | |
| **Created by** | Huỳnh Vũ | | **Last updated by** | | Huỳnh Vũ |
| **Date Created** | Dec 01, 2024 | | **Date last updated** | | Dec 01, 2024 |
| **Actors** | Employee, System | | | | |
| **Brief Description** | This use case allows employees to track their attendance schedules, including clock-in/out times, total hours worked, and overtime | | | | |
| **Goal** | To provide employees and managers with accurate and up-to-date attendance records | | | | |
| **Trigger** | An employee logs into the system to review attendance records | | | | |
| **Pre-conditions** |  The employee is registered in the system   The work schedule has been entered into the system | | | | |
| **Post-conditions** | * The employee has reviewed and verified their attendance records | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Employee logs into the system | | The system displays the employee's dashboard, including their attendance schedule | |
| **2** | Employee selects the "**Attendance Schedule**" option | | The system displays clock-in/out times, total hours worked, and any overtime | |
| **4** | Employee reviews their attendance records | |  | |
| **3** | Staff press exit to display main menu | | The system displays the main menu as if you had just logged in | |
| **Alternative Flow** | **Alternative Flow 1: Accessing past attendance records** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Employee logs into the system | | The system displays the employee's dashboard, including their attendance schedule | |
| **2** | Employee selects the "**Attendance Schedule**" option | | The system displays clock-in/out times, total hours worked, and any overtime | |
| **4** | Employee selects the "**View Past** **Records**" option | | The system prompts the employee to select a date range | |
|  | Employee selects the desired date range and submits the request | | The system retrieves and displays the attendance records for the selected period | |
| **3** | Staff selects return to return to current schedule | | Current calendar display system | |
| **Alternative Flow 2**: **Reporting attendance discrepancies** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Employee logs into the system | | The system displays the employee's dashboard, including their attendance schedule | |
| **2** | Employee selects the "**Attendance Schedule**" option | | The system displays clock-in/out times, total hours worked, and any overtime | |
| **3** | Employee identifies an error in their attendance record | | The system provides an option to flag the discrepancy | |
| **4** | Employee selects the "**Report Discrepancy**" option and enters details about the error | | The system logs the discrepancy for further review and displays a confirmation message | |
| **Exception Flow** | **Exception Flow 1: Network connectivity issue** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Employee logs into the system | | The system displays the employee's dashboard, including their attendance schedule | |
| **2** | Employee selects the "**Attendance Schedule**" option but loses network connectivity | | The system displays an error message: "**Network error. Please check your connection and try again.**" | |
| **3** | Employee reconnects to the network and retries accessing the schedule | | The system queues the request and processes it once the connection is restored | |
| **Exception Flow 2**: **System error during data retrieval** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Employee logs into the system | | The system displays the employee's dashboard, including their attendance schedule | |
| **2** | Employee selects the "**Attendance Schedule**" option, but the system encounters an error while retrieving data | | The system displays an error message: "**Error retrieving attendance data. Please try again later**" | |
| **3** | Employee retries accessing the data | | The system logs the error and attempts to retrieve the data again | |
| **Priority** | High | | | | |
| **Business Rule** | Real-Time Updates  Discrepancy Reporting  Attendance Data Accessibility  Network Dependency | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.09 | | | | |
| **Use Case Name** | Manage User Accounts | | | | |
| **Created by** | Huỳnh Vũ | | **Last updated by** | | Huỳnh Vũ |
| **Date Created** | Dec 01, 2024 | | **Date last updated** | | Dec 01, 2024 |
| **Actors** | Admin, System | | | | |
| **Brief Description** | This use case allows the admin to manage user accounts, including creating, updating, disabling, and deleting accounts. Admins can also reset passwords and assign roles to users | | | | |
| **Goal** | To ensure secure and efficient management of user accounts | | | | |
| **Trigger** | An admin logs into the system to manage user accounts | | | | |
| **Pre-conditions** |  The admin has the necessary permissions to manage user accounts   The system contains user account data | | | | |
| **Post-conditions** | * User account changes are successfully applied * All account modifications are logged for audit purposes | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Admin logs into the system with admin credentials | | The system verifies the credentials and displays the admin dashboard | |
| **2** | Admin selects the "Manage User Accounts" option | | The system displays a list of all user accounts | |
| **3** | The administrator selects an action for an individual account to update specific information | | The system checks whether the account is in a confidential or highly restricted status | |
|  | **4** |  | | The system processes the action and applies changes to the account | |
|  | **5** | Admin comfirm when complete | |  | |
|  | **Alternative Flow 1: Insufficient privileges** | | | | |
|  | **Step** | **Actor Action** | | **System Response** | |
|  | **1** |  | | The system detects that users with a higher rank are unable to view the information | |
|  | **2** |  | | The system return displays a list of all user accounts | |
|  | **3** | Admin comfirm | |  | |
| **Exception Flow** | **Exception Flow 1: The admin is adding high-level information** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** |  | | If the admin is adding high-level information | |
| **2** |  | | The system processes the action and applies changes to the account | |
| **Priority** | High | | | | |
| **Business Rule** | Order confirmation  Order notification Timeliness  Order status updates  Handling declined orders  Network onnectivity requirement | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.05 (sửa rồi) | | | | |
| **Use Case Name** | Food Order Confirmation | | | | |
| **Created by** | Huỳnh Vũ | | **Last updated by** | | Huỳnh Vũ |
| **Date Created** | Dec 01, 2024 | | **Date last updated** | | Dec 01, 2024 |
| **Actors** | Customer, Waiter, System | | | | |
| **Brief Description** | This use case allows the customer to place an order and ensures that the waiter confirms and serves the correct dish to the customer. The system tracks the order status throughout the process, updating it from "New" to "Order Confirmed," and finally to "Served" and "Completed." | | | | |
| **Goal** | Ensure that the correct dish is served to the customer efficiently and accurately | | | | |
| **Trigger** | The use case is triggered when a customer places an order via the system, either through a mobile application or at the table using the restaurant's digital interface | | | | |
| **Pre-conditions** | · Customer has successfully placed an order through the system  · The waiter is logged into the system with access to the order details | | | | |
| **Post-conditions** | * The order status is updated to "**Completed**." * The customer receives the correct dish as per the order | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Customer places an order via the system | | The system records the order and assigns a unique order ID | |
| **2** | Waiter views the order details on the system interface | | The system checks whether the customer has changed tables | |
| **3** |  | | The system updates the order status to "Order Confirmed." | |
| **4** | Waiter serves the dish to the customer and marks the order as "Served" in the system | | The system logs the transaction and updatethe status to "**Completed**." | |
| **5** | The customer confirms that the dish has been served | |  | |
| **Alternative Flow** | **Alternative Flow 1:**  **Customer Changes Table After Placing Order** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** |  | | The customer changes tables in the system | |
| **2** |  | | The system logs the transaction and updatethe status to "**Completed**." | |
| **Exception Flow** | **Exception Flow 1:** Receives a report from the customer about changing tables | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** |  | | The system receives a report from the customer about changing tables | |
| **2** |  | | The system successfully records the time entry | |
| **Priority** | High | | | | |
| **Business Rule** | Valid login  Clock-in before shift start  Clock-out after shift end  Duplicate clock-in/out prevention | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.010 | | | | |
| **Use Case Name** | Menu Management | | | | |
| **Created by** | Huỳnh Vũ | | **Last updated by** | | Huỳnh Vũ |
| **Date Created** | Dec 01, 2024 | | **Date last updated** | | Dec 01, 2024 |
| **Actors** | Admin, System | | | | |
| **Brief Description** | This use case allows the admin to manage the restaurant's menu effectively. The admin can view, add, update, or delete dishes in the menu | | | | |
| **Goal** | The goal of this use case is to provide the admin with a seamless and efficient way to manage the restaurant's menu | | | | |
| **Trigger** | The use case is triggered when the admin logs into the system and selects the "Menu Management" option from the system's interface to view, add, update, or delete menu items | | | | |
| **Pre-conditions** |  The admin must have valid credentials to log into the system and access the menu management interface   The system must be functional and connected to the database for retrieving and saving menu data | | | | |
| **Post-conditions** | * Any changes made by the admin * The updated menu contributes to smoother restaurant operations and enhances the customer experience | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Admin logs into the system using valid credentials | | The system verifies the credentials and grants access to the menu management interface | |
| **2** | Admin views the current menu with all dishes and their details | | The system displays the menu data in a sortable | |
| **3** | Admin selects an action: add a new dish, update an existing dish | | The system provides an interface for the selected action | |
|  | **4** | Admin selects an action: add a new dish, update an existing dish | | The system validates the input | |
|  | **5** |  | | The system validates the changes and applies them to the menu | |
|  | **6** | Admin comfirm when complete | | The system updates the database | |
|  | **Alternative Flow 1: The name menu is duplicated** | | | | |
|  | **Step** | **Actor Action** | | **System Response** | |
|  | **1** |  | | Menu checking system for duplicates | |
|  | **2** |  | | Displays an error message: "Unable to update the menu. Please try again later." | |
|  | **3** | Admin comfirm | |  | |
| **Exception Flow** | **Exception Flow 1: System Error During Menu Update** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** |  | | The system encounters an error while applying updates to the menu | |
| **2** |  | | The system validates the changes and applies them to the table layout | |
| **Priority** | High | | | | |
| **Business Rule** | Unique Dish Names  Validation Before Changes  Admin Permissions | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.011 | | | | |
| **Use Case Name** | Table Layout Management | | | | |
| **Created by** | Huỳnh Vũ | | **Last updated by** | | Huỳnh Vũ |
| **Date Created** | Dec 01, 2024 | | **Date last updated** | | Dec 01, 2024 |
| **Actors** | Admin, System | | | | |
| **Brief Description** | This use case allows the admin to manage the restaurant's table layout effectively. The admin can view, update, add, or delete tables, as well as configure their details, such as table numbers, seating capacity, and status | | | | |
| **Goal** | The goal of this use case is to enable the admin to efficiently organize and manage the restaurant's table layout | | | | |
| **Trigger** | An admin logs into the system to manage user accounts | | | | |
| **Pre-conditions** | · The admin must have valid credentials to log into the system and access the table layout management interface  · The system must be operational and connected to the database for retrieving and updating table information | | | | |
| **Post-conditions** | * Any changes made by the admin are successfully applied and saved in the system * The updated table layout is reflected in the waiter and customer interfaces immediately | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Admin logs into the system using valid credentials | | The system verifies the credentials and grants access to the table layout management interface | |
| **2** | Admin views the current table layout | | The system displays the table layout, including table numbers, statuses | |
| **3** | Admin selects an action: add a new table, update table details, or delete a table | | The system provides the appropriate interface for the selected action | |
|  | **4** | Admin confirms the changes | | The system checks whether the changes are duplicated | |
|  | **5** |  | | The system validates the changes and applies them to the table layout | |
|  | **6** | Admin comfirm when complete | |  | |
|  | **Alternative Flow 1: the table number is duplicated** | | | | |
|  | **Step** | **Actor Action** | | **System Response** | |
|  | **1** |  | | The system checks if the table number is duplicated | |
|  | **2** |  | | The system return displays the table layout, including table numbers, statuses | |
|  | **3** | Admin comfirm | |  | |
| **Exception Flow** | **Exception Flow 1: If the table numbers are duplicated** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** |  | | If the table number is duplicated, the system will combine the tables into a larger one | |
| **2** |  | | The system validates the changes and applies them to the table layout | |
| **Priority** | High | | | | |
| **Business Rule** | Unique Table Numbers  Validation Before Applying Changes  Admin Permissions | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.012 | | | | |
| **Use Case Name** | Bill Management | | | | |
| **Created by** | Huỳnh Vũ | | **Last updated by** | | Huỳnh Vũ |
| **Date Created** | Dec 01, 2024 | | **Date last updated** | | Dec 01, 2024 |
| **Actors** | Admin, System | | | | |
| **Brief Description** | This use case allows the admin to manage the restaurant's bill effectively. The admin can view, add, update, or delete dishes in the menu | | | | |
| **Goal** | The goal of this use case is to provide the admin with a seamless and efficient way to manage the restaurant's bill | | | | |
| **Trigger** | The use case is triggered when the admin logs into the system and selects the "Bill Management" option from the system's interface to view, add, update, or delete bill items | | | | |
| **Pre-conditions** | · The admin must have valid credentials to log into the system and access the bill management interface  · The system must be functional and connected to the database for retrieving and saving bill data | | | | |
| **Post-conditions** | * Any changes made by the admin * The updated bill contributes to smoother restaurant operations and enhances the customer experience | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Admin logs into the system using valid credentials | | The system verifies the credentials and grants access to the bill management interface | |
| **2** | Admin views the current bill with all dishes and their details | | The system displays the bill data in a sortable | |
| **3** | Admin selects an item to update its price | | The system provides an interface to update the price of the selected item | |
|  | **4** | Admin enters the new price | | The system check validates the input to the price is a positive value | |
|  | **5** |  | | The system updates the database with the new price | |
|  | **6** | Admin comfirm when complete | | The system updates the database | |
|  | **Alternative Flow 1: Price is a negative value** | | | | |
|  | **Step** | **Actor Action** | | **System Response** | |
|  | **1** |  | | The system detects that the entered price is a negative number | |
|  | **2** |  | | Return to the initial interface | |
|  | **3** | Admin comfirm | |  | |
| **Exception Flow** | **Exception Flow 1: System Error During Menu Update** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** |  | | The system encounters an error while applying updates to the bill | |
| **2** |  | | The system validates the changes and applies them to the bill | |
| **Priority** | High | | | | |
| **Business Rule** | Unique Price per Item  Positive Price Value  Admin Permissions | | | | |