

**TABLE RESERVATION**

**System Requirement Specification**

**Team 1 - HCM23\_CPL\_BA\_01**

**Trần Hoàng Kim Ngân**

**Nguyễn Việt An**

**Vũ Mạnh Cường**

**Nguyễn Thành Trung**

**Nguyễn Minh Duy**

**Lê Quốc Cường**

**Document Revision History**

| **Ver.** | **Date** | **Changed by** | **Modifications** |
| --- | --- | --- | --- |
| 0.1 | 15/06/23 | Trần Hoàng Kim Ngân | Introduction and Scope |
| 0.2 | 24/06/23 | Trần Hoàng Kim Ngân | Use Case Diagram |
| 0.3 | 30/06/23 | Trần Hoàng Kim Ngân | Use Case Detail |
| 0.4 | 04/07/23 | Trần Hoàng Kim Ngân | Delete Resercation Management Use Case |
| 0.5 | 13/07/23 | Trần Hoàng Kim Ngân | Add Check in , Check out Use Case |
| 0.6 | 14/07/23 | Trần Hoàng Kim Ngân | Update Asign table Use Case |
| 0.7 | 19/07/23 | Trần Hoàng Kim Ngân | Non-Functional Requiment |

**Table of Contents**

[**● 7**](#_3znysh7)

[**1. Introduction 7**](#_2et92p0)

[1.1. Purpose 7](#_tyjcwt)

[1.2. Scope 8](#_3dy6vkm)

[1.3. Out of Scope 9](#_8g7w7vo0vqqn)

[1.4. System perspective 9](#_wdsfx21p0cw)

[1.4.1. Assumptions 9](#_ctiokh1v2o6n)

[1.4.2. Constraints 10](#_4w3226tdd3ou)

[**2. Business Process overview 10**](#_81ciwinpj589)

[2.1. Business Process 10](#_1rkwpslr44gk)

[■ Figure 2.1\_1:Process 10](#_wgjanc4lftv6)

[2.2. Use Case Story 10](#_k9evnl4ikuhb)

[■ 10](#_bxglgd92p5dn)

[**3. System Diagram 10**](#_buxhiww8bev9)

[3.1. Use case diagram 10](#_klogqbnzo146)

[■ Figure 3.1 Table Reservation Diagram 11](#_zc0mpc27i359)

[3.2. Entity Relationship diagram 12](#_4d34og8)

[■ Figure 3.2\_1: Entity Relationship Diagram 12](#_3xmu2vf5mk4w)

[**4. Functional Requirements 12**](#_9w84z16ve8sf)

[4.1. Use case detail 12](#_q2q2furebwzb)

[4.1.1. Login 12](#_khzmyluglb2d)

[4.1.2. Logout 13](#_uwj9tubthz3h)

[4.1.3. Register 13](#_vasxxcqxli3y)

[4.1.4. Manage Profile 14](#_hqma52v6uuuy)

[4.1.5. Check Table Availability 14](#_tx1l22vpity4)

[4.1.6. Create Table Reservation 15](#_8pv29f556nh0)

[4.1.7. View Reservation Details 16](#_9326pt5bfxmf)

[4.1.8. Pre-Order Food 16](#_c65a9t35nzcc)

[4.1.9. Payment 17](#_61yb8vw4eltw)

[4.1.10. Modify Reservation 17](#_fo4b8uf5pema)

[4.1.11. Cancel Reservation 18](#_p1wfs57gewoj)

[4.1.12. Assign Table 18](#_4rhh2mffhr1g)

[4.1.13. Check-in 19](#_y84hj2cjtrb)

[4.1.14. Check out 20](#_6lzqf8nrdwme)

[4.1.15. Manage Table 20](#_dx2zisai2dlt)

[4.1.16. Manage User 21](#_9r8n6bov06nh)

[**5. Non-Functional Requirement 22**](#_sqyw64)

[5.1. Usability 22](#_3gnlt4p)

[5.2. Reliability 22](#_6x3nln2amyv5)

[5.3. Performance 22](#_sbno8vbglfvu)

[5.4. Supportability 22](#_ka5n2w2y8yit)

[5.5. Legal, Copyright, and Other Notices 22](#_djai3ttsrjsl)

[5.6. Summary 22](#_hyhxsim3l0jt)

[**6. References 23**](#_dfyw6s8pmggl)

# 

# Introduction

## Purpose

The purpose of this document is to define the requirements and specifications for the development of a 5-star restaurant table reservation application. The document aims to provide a comprehensive understanding of the desired functionalities, features, and constraints of the application. It serves as a foundation for the development team, stakeholders, and project management to ensure a clear understanding of the project's objectives and deliverables.

## Scope

| No. | Feature | Description | Actor |
| --- | --- | --- | --- |
| 1 | Login | Users log into the table reservation application using their Google account credentials | Customer, Reception |
| 2 | Logout | Users log out of the table reservation application | Customer, Reception |
| 3 | Register | new users register for the table reservation application using their Google account | Customer, Reception |
| 4 | Manage Profile | users to manage their profile information within the table reservation application. Users can view and update their personal details, such as name, contact information. | Customer, Reception |
| 5 | Check Table Avaibility | users can check the availability of tables in the restaurant for a specific date and time. Users can view the available tables and make informed decisions for their reservation | Customer, Reception |
| 6 | Create Table Reservation | users create a table reservation. Users can specify the desired date, time, type of table and the number of guests. The system checks table availability and allows users to proceed with the reservation if a table is available at least 1 hour in advance. | Customer, Reception |
| 7 | View Reservation Details | Users view the details of their existing reservations. Users can access information such as reservation date, time, table number, and any pre-ordered food items. | Customer, Reception |
| 8 | Pre-Order Food | Users pre-order food items while creating a table reservation. Users can select the desired menu items for their meal. | Customer, Reception |
| 9 | Payment | users make payments for their pre-ordered food items by Zalo Pay | Customer, Reception |
| 10 | Modify Reservation | users modify their existing reservations. Users can update the reservation date, time, or the number of guests, provided that the reservation is still pending or assigned. Modifications are allowed up to 1 hour before the booked time. | Customer, Reception |
| 11 | Cancel Reservation | users cancel their existing reservations. Users can request the cancellation of a reservation if it is still pending or assigned. Cancellations are allowed up to 1 hour before the booked time. | Customer, Reception |
| 12 | Assign Table | the receptionist assign a table to a reservation. Once a reservation is confirmed and the booking time is addressed, the receptionist assigns an available table to the reservation. The assigned table ensures that the reserved table is reserved for the specific reservation and prevents double bookings or conflicts. | Reception |
| 13 | Check In | This use case involves the check-in process for customers when they arrive at the restaurant. The receptionist checks in the customers in the system to mark their arrival and update the reservation status. | Reception |
| 14 | Check Out | This use case involves the check-out process for customers when they leave the restaurant or arrive more than 30 minutes late. The receptionist checks out the customers in the system to mark their departure and update the reservation status. | Reception |
| 15 | Manage Table | the receptionist or authorized personnel to manage tables within the restaurant. They can create, update, and remove tables as necessary, ensuring an accurate representation of table availability in the system. | Reception |
| 16 | User Management | This use case involves the management of user accounts within the table reservation application. Authorized personnel can create, update, or deactivate user accounts, ensuring proper user access and security. | Reception, System Admin |

## Out of Scope

* Feedback
* Inventory management

## System perspective

### Assumptions

1. The restaurant already has a physical infrastructure in place to handle table reservations and food preparation.
2. The restaurant has a reliable network and internet connectivity for seamless operation of the application.
3. Customers will have access to smartphones or other devices capable of running the application.

### Constraints

1. The application will be developed for a specific 5-star restaurant, tailored to its branding and requirements.
2. The development will be conducted within a specific timeline and budgetary constraints.
3. The application will be developed for a specific platform(s) or devices as determined by the restaurant's target audience and technological infrastructure.

# Business Process overview

# System Diagram

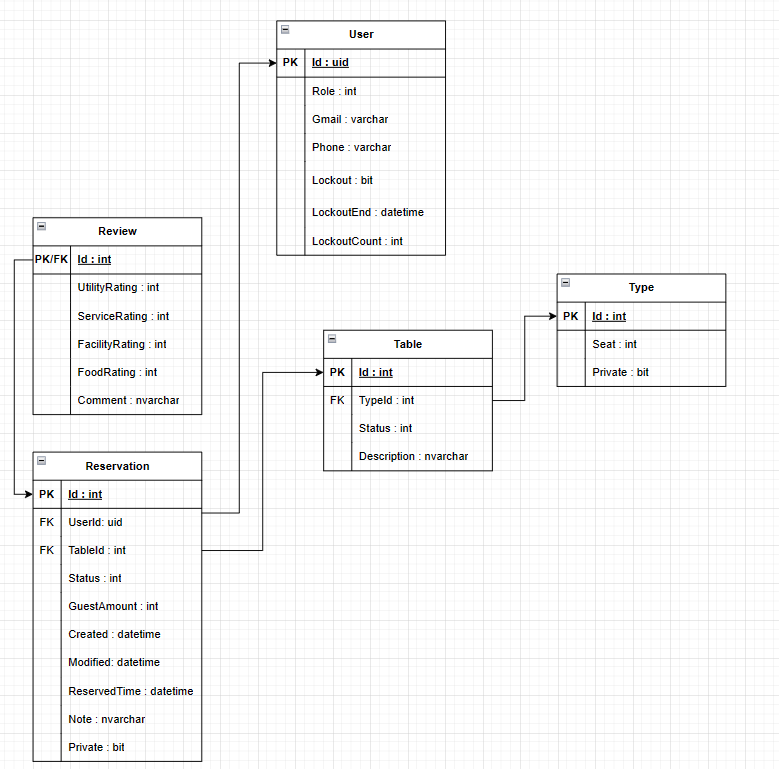
## Use case diagram

### 



###### Figure 3.1 Table Reservation Diagram

## Entity Relationship diagram



###### Figure 3.2\_1: Entity Relationship Diagram

# Functional Requirements

## Use case detail

### Login

| UC ID | 01 | Precondition:  The user must have a valid Google account. |
| --- | --- | --- |
| Description | Users log into the table reservation application using their Google account credentials. | Postcondition:  The user is successfully logged into the table reservation application. |
| Actor | Customer, Reception | Dependency:  N/A |
| Main Flow | 1. The user opens the application and selects the login option. 2. The application prompts the user to enter their Google account credentials. 3. The user enters their credentials. 4. The application verifies the credentials with the Google authentication service. 5. If the authentication is successful, the application logs the user into the system. | |
| Alternative Flow | N/A | |
| Exception flow | Step 5:.  If there is a network issue or the Google authentication service is unavailable, the application displays an error message. | |
| Business Rule | Users must have a valid Google account to log into the application | |

### Logout

| UC ID | 02 | Precondition:  The user must be logged into the table reservation application. |
| --- | --- | --- |
| Description | Users log out the table reservation application. | Postcondition:  The user is successfully logged out of the table reservation application. |
| Actor | Customer, Reception | Dependency:  N/A |
| Main Flow | 1. The user selects the logout option within the application. 2. The application clears the user's session and logs them out. | |
| Alternative Flow | N/A | |
| Exception flow | N/A | |
| Business Rule | N/A | |

### Register

| UC ID | 03 | Precondition:  The user must have a valid Google account. |
| --- | --- | --- |
| Description | Users register for the table reservation application using their Google account. | Postcondition:  The user is successfully registered and logged into the table reservation application. |
| Actor | Customer, Reception | Dependency:  N/A |
| Main Flow | 1. The user opens the application and selects the registration option. 2. The application prompts the user to enter their Google account credentials. 3. The user enters their credentials. 4. The application verifies the credentials with the Google authentication service. 5. If the authentication is successful, the application registers the user and logs them into the system. | |
| Alternative Flow | N/A | |
| Exception flow | Step 5:.  If there is a network issue or the Google authentication service is unavailable, the application displays an error message. | |
| Business Rule | Users must have a valid Google account to register the application | |

### Manage Profile

| UC ID | 04 | Precondition:  The user must be logged into the table reservation application. |
| --- | --- | --- |
| Description | users to manage their profile information within the table reservation application. Users can view and update their personal details, such as name, contact information.. | Postcondition:  The user's profile information is successfully updated. |
| Actor | Customer, Reception | Dependency:  N/A |
| Main Flow | 1. The user Click to User Avatar to go to User Profile 2. The application displays the user's current profile information. 3. The user can update their profile details such as name, contact information .etc 4. The user saves the changes. 5. The application validates and updates the user's profile information. | |
| Alternative Flow | N/A | |
| Exception flow | Step 5:.  If the user enters invalid information, the application displays an error message. | |
| Business Rule | N/A | |

### Check Table Availability

| UC ID | 05 | Precondition:  The user must be logged into the table reservation application |
| --- | --- | --- |
| Description | users can check the availability of tables in the restaurant for a specific date and time, base on number of customer and type of table. Users can view the available time to booking table. | Postcondition:  The user views the time availability of tables reservation for the specified date, base on number of customer and type of table. |
| Actor | Customer, Reception | Dependency:  N/A |
| Main Flow | 1. The user click the button "+" within the application. 2. The application prompts the user to enter the desired date. 3. The user enters the date. 4. The application prompts the user to enter the desired number of people . 5. The user enters number of people. 6. The application prompts the user to select type of table. 7. The user select the type of table . 8. The application checks the availability of tables for above requiment. 9. The application displays the list of available time to the user. | |
| Alternative Flow | N/A | |
| Exception flow | Step 5:.  N/A | |
| Business Rule | N/A | |

### Create Table Reservation

| UC ID | 06 | Precondition:  The user must be logged into the table reservation application and a table must be available for the specified date and time |
| --- | --- | --- |
| Description | users create a table reservation. Users can specify the desired date, time, type of table and the number of guests. The system checks table availability and allows users to proceed with the reservation if a table is available table and make a reservation at least 1 hour in advance. | Postcondition:  The table reservation is successfully created. |
| Actor | Customer, Reception | Dependency:  Check Table Availability |
| Main Flow | 1. After Check Table Availity,The application displays the list of available time to the user . 2. The user select the Time. 3. The user Click button “ Create New Reservation.” 4. The application Save the new reservation in the system with status “pending”. | |
| Alternative Flow | Step 2: If no time period are available for the specified date, number of people and type of table, the application displays an message and prompts the user to choose a different date or number of people, or type of table. | |
| Exception flow | Step 4 :.  If there is a network issue or the system fails to create the reservation, the application displays an error message. | |
| Business Rule | Users must make a reservation at least 1 hour in advance. | |

### View Reservation Details

| UC ID | 077 | Precondition:  The user must be logged into the table reservation application and have an existing reservation. |
| --- | --- | --- |
| Description | This use case allows users to view the details of their existing reservations. | Postcondition:  The user views the details of their reservation. |
| Actor | Customer, Reception | Dependency:  Manage Reservation |
| Main Flow | 1. The user selects the "Reservation History" option within the application. 2. The application displays a list of the user's existing reservations. 3. The user selects a specific reservation. 4. The application retrieves and displays the details of the selected reservation, including date, time, number of people, type of table, table number. | |
| Alternative Flow | N/A | |
| Exception flow | N/A | |
| Business Rule | Users can only view reservation details if they have existing reservations. | |

### Modify Reservation

| UC ID | 8 | Precondition:  The user must be logged into the table reservation application and have an existing reservation that is still pending or assigned. |
| --- | --- | --- |
| Description | This use case allows customers to modify their existing reservations. | Postcondition:  The reservation is successfully modified with the updated details |
| Actor | Customer, Reception | Dependency:  Manage Reservation |
| Main Flow | 1. The user selects the "Reservation History" option within the application. 2. The application displays the user's existing reservations. 3. The user selects the reservation they wish to modify. 4. The application retrieves and displays the details of the selected reservation. 5. The user updates the reservation details such as date, time, type of table or the number of guests. 6. The user saves the changes. 7. The application validates and updates the reservation with the new details and change reservation status to “pending”. | |
| Alternative Flow | N/A | |
| Exception flow | Step 7:  If the user tries to modify a reservation less than 1 hour before the booked time, the application displays an error message.  If there is a network issue or the system fails to modify the reservation, the application displays an error message. | |
| Business Rule | Users can only modify reservations that are still pending or assigned, and modifications are allowed up to 1 hour before the booked time. | |

### Cancel Reservation

| UC ID | 9 | Precondition:  The user must be logged into the table reservation application and have an existing reservation that is still pending or assigned　and a customer with a “Active” reservation must leave or arrive more than 30 minutes late. |
| --- | --- | --- |
| Description | This use case allows customers to cancel their existing reservations or when they arrive more than 30 minutes late the reservation status change to cancel | Postcondition:  The reservation is successfully canceled. |
| Actor | Customer, Reception | Dependency:  Manage Reservation |
| Main Flow | 1. The user selects the "Reservation History" option within the application. 2. The application displays the user's existing reservations. 3. The user selects the reservation they wish to cancel. 4. The application retrieves and displays the details of the selected reservation. 5. The user confirms the cancellation. 6. The application cancels the reservation and updates the reservation status. | |
| Alternative Flow | Step1 :  If the assigned reservation is not "Active" after 30 minutes, the application will change the reservation status to "Complete" | |
| Exception flow | Step 6:  If the user tries to cancel a reservation less than 1 hour before the booked time, the application displays an error message.. | |
| Business Rule | * Users can only cancel reservations that are still pending or assigned, and cancellations are allowed up to 1 hour before the booked time. * when customer arrive more than 30 minutes late the reservation status change to “cancel” | |

### Assign Table

| UC ID | 10 | Precondition:  The receptionist or system must have a confirmed reservation that requires a table assignment. |
| --- | --- | --- |
| Description | the receptionist assign a table to a reservation. Once a reservation is confirmed and the booking time is addressed, the receptionist assigns an available table to the reservation. The assigned table ensures that the reserved table is reserved for the specific reservation and prevents double bookings or conflicts. | Postcondition:  A table is successfully assigned to the reservation. |
| Actor | Reception | Dependency:  N/A |
| Main Flow | 1. The user selects the "Pending Reservation" option within the application. 2. The application displays the pending reservations. 3. The receptionist checks the available tables for the reservation's date and time. 4. The receptionist or system selects an available table and assigns it to the reservation. 5. The assigned table is marked as reserved for the specific reservation in the system. | |
| Alternative Flow | N/A | |
| Exception flow | N/A | |
| Business Rule | Users can only Assign Table to the reservation which are still pending | |

### Check-in

| UC ID | 11 | Precondition:  The receptionist must be logged into the table reservation application and a customer with a assigned reservation must be present. |
| --- | --- | --- |
| Description | This use case involves the check-in process for customers when they arrive at the restaurant. | Postcondition:  The customer's check-in is successfully recorded in the system. |
| Actor | Reception | Dependency:  N/A |
| Main Flow | 1. The user selects the "Assigned Reservation " option within the application. 2. The application displays the assigned reservations. 3. The user selects the reservation they wish to check in. 4. The user click check-in in the system. 5. The application change reservation status to “Active”. | |
| Alternative Flow | N/A | |
| Exception flow | N/A | |
| Business Rule | Check-in can only be performed by the receptionist when a customer with a assigned reservation arrives at the restaurant. | |

### Check-out

| UC ID | 12 | Precondition:  The receptionist must be logged into the table reservation application |
| --- | --- | --- |
| Description | This use case involves the check-out process for customers when they leave the restaurant | Postcondition:  The customer's check-out is successfully recorded in the system |
| Actor | Reception | Dependency:  N/A |
| Main Flow | 1. The user selects the "Active Reservation " option within the application. 2. The application displays the assigned reservations. 3. The user selects the reservation they wish to check in. 4. The user click check-in in the system. 5. The application change reservation status to “Complete”. | |
| Alternative Flow | N/AA | |
| Exception flow | N/A | |
| Business Rule | Check-out can only be performed by the receptionist when a customer with a checked-in reservation leaves the restaurant . | |

### Manage Table

| UC ID | 13 | Precondition:  The receptionist or authorized personnel must be logged into the table reservation application. |
| --- | --- | --- |
| Description | the receptionist to manage tables within the restaurant. They can create, update, and remove tables as necessary, ensuring an accurate representation of table availability in the system. | Post condition:  The table management task (create, update, or remove) is successfully performed. |
| Actor | Reception | Dependency:  N/A |
| Main Flow | 1. The receptionist selects the "Manage Table" option within the application. 2. The application displays the current list of tables in the restaurant. 3. The receptionist can perform tasks such as creating a new table, updating existing table details, or removing a table from the list. 4. The application updates the table information in the system. | |
| Alternative Flow | N/A | |
| Exception flow | N/A | |
| Business Rule | In restaurant just have table for 2 seats, 4 seats and 10 seats. In restaurant just have 2 type of table public and private. | |

.

### Manage User

| UC ID | 14 | Precondition:  The Actor must be must be logged into the table reservation application.. |
| --- | --- | --- |
| Description | This use case involves the management of user accounts within the table reservation application. Actor can create, update, or deactivate user accounts, ensuring proper user access and security. | Post condition:  The user management task (create, update, or deactivate) is successfully performed. |
| Actor | System Admin, Reception | Dependency:  N/A |
| Main Flow | 1. The Actor selects the "User Management" option within the application. 2. The application displays the list of user accounts. 3. The Actor can perform tasks such as creating a new user account, updating existing user account details, or deactivating a user account. 4. The application updates the user account information in the system.. | |
| Alternative Flow | N/A | |
| Exception flow | N/A | |
| Business Rule | N/A | |

# Non-Functional Requirement

## ***Usability***

Followed by UI/UX rules to design the system. Make sure users can easily use the system without any training before.

## ***Reliability***

System should provide reliable and relevant search results 100% of times

## ***Performance***

For the number of users increase or decrease, it still does not affect the tasks (performance) of the application. It has maximum 8s to load a page and search a result displayed within 5s.

## ***Supportability***

This system supports interface interaction with Fsoft authentication. It’s also supported to import and export files in excel, doc, pdf, and attach the other website links.

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

The application’s legal, copyright and other notices belong to 5 Star Restaurant.

## ***Summary***

| Response Time and Net Processing Time | Response within 5s |
| --- | --- |
| Operation time | 24/7 |
| Business continuity requirements | In disaster case, system can recover data |
| Data recovery range | Can recover data up to 24h ago |
| Number of concurrent users | 1000 |
| Secure coding, web server setting | Implement measures against threats and vulnerabilities specific to mobile applications |
| Session timeout | 30 mins |

# References

The complete list of documents referenced in a Software Requirements Specification (SRS) typically includes various types of out-of-stream data for SRS completion:

All the screens UI and mock-ups are stored at:

* https://www.figma.com/file/YYwPAVar2c9h0LL6Bje0Za/Restaurant-App-(PRM)?type=design&node-id=0%3A1&mode=design&t=dratzSPZ2TVyworY-1

All the Diagram and ERD are stored at:

* https://drive.google.com/file/d/1NjYv\_Tlb\_poqoicfBOyrRRnxgiRdcZBy/view?usp=sharing