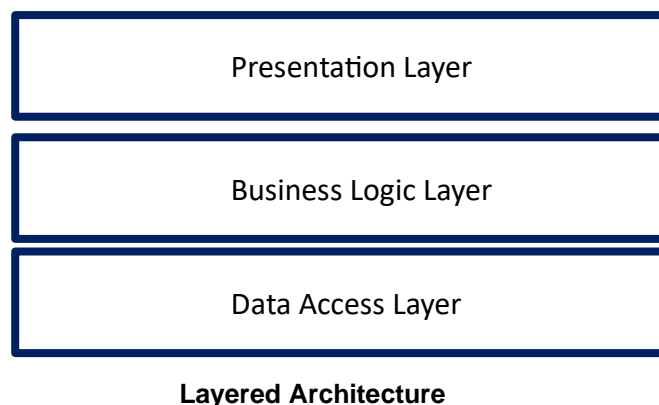


ReserveWell	V1.2
Iteration Plan	Date: 16/12/2023

## Design structure

The design structure of the ReserveWell Application is organized in a layered architecture, ensuring modularity, scalability, and maintainability. Each layer serves a specific purpose and collaborates seamlessly to achieve the overall goal of efficiently managing restaurant reservations.



## Subsystems

The application has no subsystems.

## Patterns

### [Pattern 1-Make Reservation]

#### Overview

The Reservation pattern is tailored to streamline the reservation process within the ReserveWell Application, specifically targeting the "Make Reservation" use case. With a focus on enhancing efficiency and user convenience, this pattern aims to provide a dynamic and adaptable framework for managing reservations effectively.

**Intent:** The primary intent of the Reservation Workflow pattern in the context of the "Make Reservation" use case is to create a seamless, user-friendly experience for diners while ensuring optimal utilization of restaurant resources. This pattern intends to automate and optimize the steps involved in making a reservation, offering a responsive system that caters to varying diner preferences and restaurant capacities.

**Motivation:** Traditional reservation systems often involve manual steps and may lack adaptability to real-time changes in diner preferences and restaurant availability. The Optimized Reservation Workflow

pattern is motivated by the need to address these challenges by introducing automation and dynamic decision-making. By doing so, the pattern aims to improve the overall reservation process, benefiting both diners and restaurant managers.

**Applicability:** The Optimized Reservation Workflow pattern is particularly applicable to the "Make Reservation" use case within the ReserveWell Application. It caters to the interests of various stakeholders, including diners, restaurant managers, system administrators, the Payment Authorization Service, and the Development Team. The pattern is relevant in scenarios where:

- Diners seek an efficient and convenient reservation process.
- Restaurant managers aim to optimize table utilization and ensure a smooth dining experience.
- System administrators need a stable and scalable reservation system.
- The Payment Authorization Service requires seamless integration with the bank's system.

By adopting the Optimized Reservation Workflow pattern, the ReserveWell Application aims to provide a responsive, user-centric reservation experience while ensuring system stability, scalability, and adherence to best practices.

## **Structure**

### ***Diner:***

- Role: Initiates the reservation process by interacting with the application's user interface.
- Relevant Data: Provides information such as the desired restaurant, the number of guests, preferred date and time, name, surname, email.
- Behavior: Triggers the reservation confirmation process by confirming the provided information.

### ***Restaurant Manager:***

- Role: Manages reservations effectively and ensures a seamless dining experience.
- Relevant Data: Accesses and updates reservation information, including table availability, to optimize the dining experience.
- Behavior: Utilizes tools and functionalities to oversee the reservation process and ensure a smooth flow of restaurant operations.

### ***Waitstaff:***

- Role: Provides efficient and high-quality service to customers, including arranging tables' physical availability based on reservation updates.
- Relevant Data: Receives real-time updates on reservations and allocates tables accordingly to enhance customer service.
- Behavior: Utilizes tools within the application to organize seating arrangements and deliver a positive dining experience.

## **Reservation System**

- Role: Central component responsible for managing reservations.
- Relevant Data: Stores and processes diner-provided information, generates a unique reservation ID, and updates the system with new reservations.
- Behavior: Validates and records diner reservations, ensuring the availability of the reserved table or dining area during the specified date and time.

### ***User Interface Modules:***

- Role: Facilitates interaction between the diner and the reservation system.
- Relevant Data: Displays restaurant options, reservation details, and confirmation messages.
- Behavior: Presents a user-friendly interface, collects diner inputs, and communicates reservation status.

### **Behavior**

#### ***Diner Initiates Reservation:***

The diner interacts with the user interface, selects a restaurant, specifies the number of guests, and chooses a date and time for the reservation.

#### ***Data Submission***

Diner enters personal details (name, surname, email, phone number) and submits the reservation request.

#### ***System Validation:***

The reservation system validates diner inputs, checking for availability and adherence to business rules.

#### ***Unique Reservation ID Generation:***

Upon successful validation, the reservation system generates a unique reservation ID and associates it with the diner's reservation.

#### ***Reservation Confirmation***

The reservation system updates its records, marking the reserved table or dining area as unavailable during the specified date and time.

#### ***Confirmation Message:***

The user interface modules present a confirmation message to the diner, indicating a successful reservation.

### **Example**

Consider a scenario where a diner, Sarah, uses the ReserveWell Application to make a reservation for two at her favorite restaurant. After entering the necessary details and confirming the reservation, the system generates a unique ID, reserves a table, and sends a confirmation message to Sarah's device. If payment is required, the Payment Authorization Service ensures a secure transaction with the bank. This example illustrates a seamless and optimized reservation workflow facilitated by the pattern.

# Requirement realizations

## View of participants

### Diner

- Behavior: Initiates the reservation process, provides necessary information, and confirms the reservation.
- Attributes: Name, surname, email, phone number, selected restaurant, number of guests, preferred date and time.
- Relationships: Interacts with the user interface, triggers reservation confirmation.

### *Restaurant Manager*

- Behavior: Manages reservations effectively, oversees real-time updates, and ensures a smooth dining experience.
- Attributes: Access to reservation management tools, real-time updates on reservations and table availability.
- Relationships: Interacts with the system to optimize table allocation and enhance the dining experience.

### *Waitstaff*

- Behavior: Provides efficient and high-quality service, organizes tables based on real-time reservation updates.
- Attributes: Access to tools for table arrangement, real-time updates on table availability.
- Relationships: Interacts with the system to optimize table arrangements and contribute to positive customer service.

### *Reservation System*

- Behavior: Validates diner inputs, generates unique reservation IDs, updates system records.
- Attributes: Reservation ID, restaurant availability status, reservation details.
- Relationships: Receives inputs from the diner, interacts with the Payment Authorization Service if required.

### *User Interface Modules*

- Behavior: Presents restaurant options, collects and displays reservation details, communicates confirmation messages.
- Attributes: User interface components, reservation status messages.
- Relationships: Facilitates interaction between the diner and the Reservation System.

## Basic scenario

### *Diner Initiates Reservation*

Diner interacts with the user interface, selects a restaurant, and specifies the number of guests, date, and time for the reservation.

### ***Data Submission***

Diner enters personal details, and the system collects and validates the reservation request.

### ***System Validation***

The Reservation System validates diner inputs, checks restaurant availability, and confirms the reservation's feasibility.

### ***Unique Reservation ID Generation***

Upon successful validation, the Reservation System generates a unique ID and associates it with the reservation.

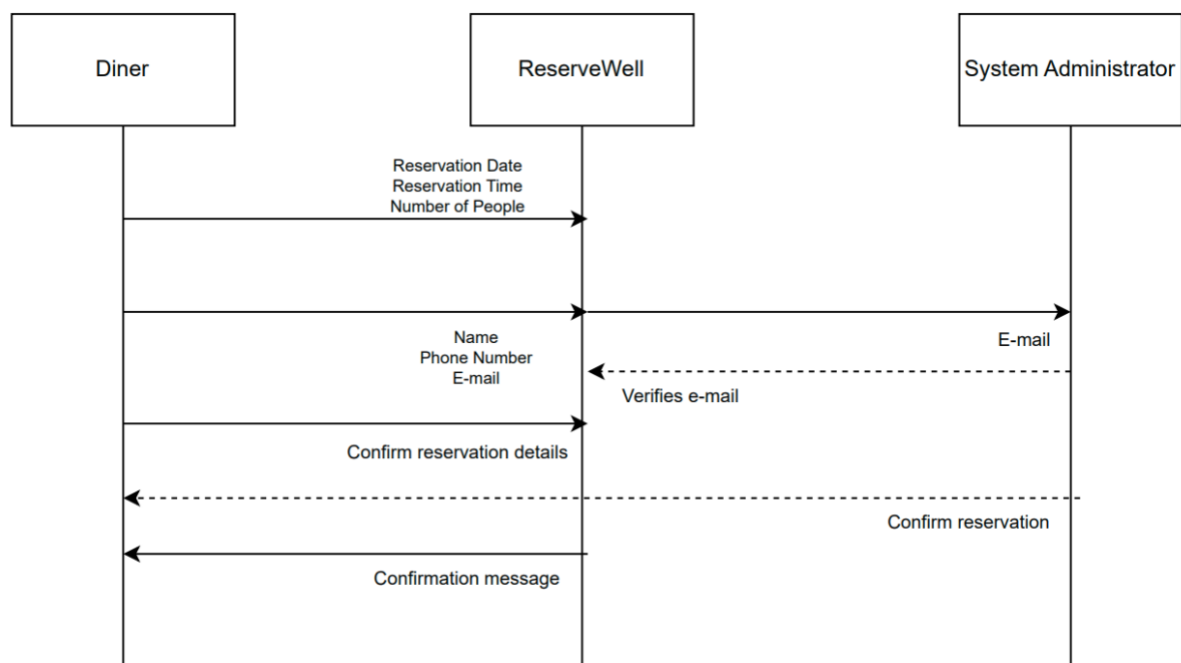
### ***Reservation Confirmation***

The Reservation System updates its records, marking the reserved table or dining area as unavailable during the specified date and time.

### ***Confirmation Message***

The User Interface Modules present a confirmation message to the diner, indicating a successful reservation.

### **Sequence Diagram For Make Reservation**



## **Additional scenarios**

### ***Scenario 1: System Failure and Recovery***

In the event of a system failure, the diner restarts the system, triggering a request for recovery of the prior state. The system then reconstructs the prior state, ensuring a seamless continuation of the reservation process.

### ***Scenario 2: Anomalies preventing recovery***

If the system detects anomalies preventing recovery, it signals an error to the diner, records the error, and enters a clean state. The diner is then prompted to start a new reservation, ensuring data integrity and system stability.

### ***Scenario: Diner Entered Incorrect Information***

In case the diner enters incorrect e-mail address, the system notifies the diner and prompts the diner to re-enter the information until successfully verified.

## **[Pattern 2-Register As A Diner]**

### **Overview**

The "Register as a Diner" pattern facilitates the process by which users, specifically diners, can create accounts within the ReserveWell application. This pattern ensures a streamlined and user-friendly registration experience, allowing diners to access personalized features such as making reservations, providing feedback, and enjoying a tailored user interface.

### **Structure**

#### ***Diner:***

Attributes: Name, email, password.

Responsibility: Initiating the registration process, entering personal details, confirming registration.

#### ***System:***

Attributes: User database, communication module, authentication logic.

Responsibility: Orchestrating the registration process, interacting with user interface and communication modules.

### **Behavior**

From a dynamic perspective, the diner collaborates with the System to enter her information and to get verified by the system.

### **Example**

John, a new user, opens the ReserveWell application and clicks on the "Register As A Diner" button. He fills out the registration form with his name, email and password. After clicking "Register," he completes the registration process, and John can now log in as a registered diner.

## **Requirement realizations**

### **View of participants**

**Diner:** Interacts with the user interface to input registration details.

**System:** Facilitates the registration process, manages user accounts, and ensures email address verification.

### **Basic Scenario**

#### ***Initiation***

The user initiates the registration process by clicking on the "Register As a Diner" button on the main screen of the ReserveWell application. The system responds by presenting the user with the registration form, prompting them to enter the required information.

#### ***Data Entry***

The enters her information, including name, surname, e-mail. The system validates the entered data in real-time, providing feedback on the correctness of the information. It ensures that mandatory fields are filled, the email format is correct, and the password meets security requirements.

#### ***Confirmation and Registration***

The user reviews the entered information and clicks the "Register" button to confirm their registration. The system processes the registration request, creating a new user account in the database. The system generates a unique user ID.

#### ***Completion***

The user is now successfully registered and can log in using their credentials. The system acknowledges the successful registration, updating the user interface to reflect the new user's registration status. The user is now a recognized and authenticated member of the ReserveWell application.

### **Additional scenarios**

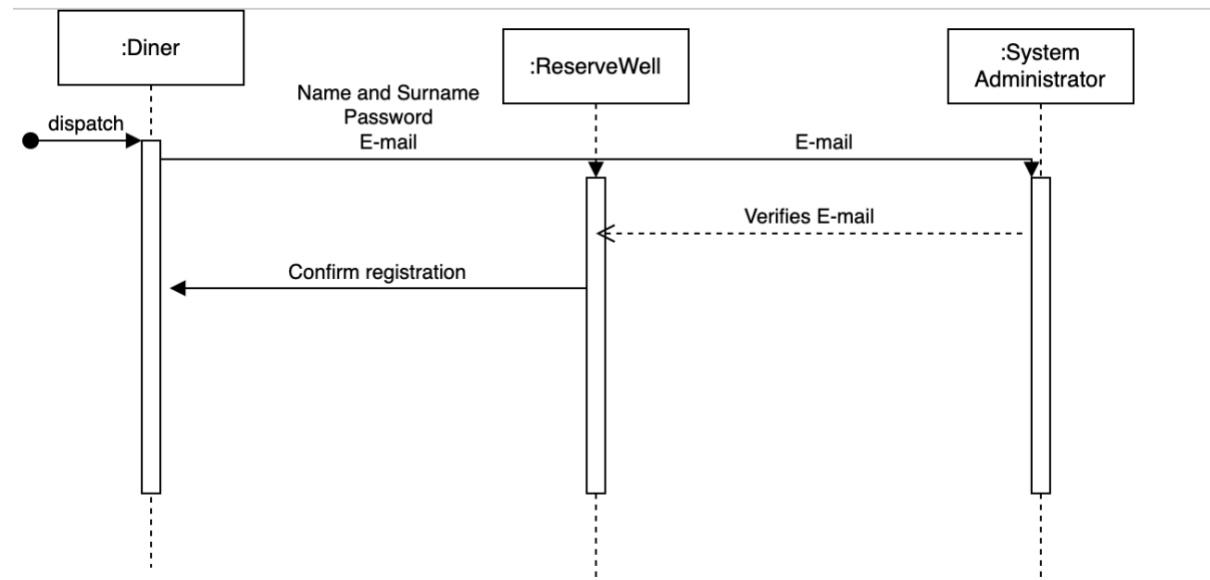
#### ***Scenario 1: System Failure and Recovery***

In the event of a system failure, the diner restarts the system, triggering a request for recovery of the prior state. The system then reconstructs the prior state, ensuring a seamless continuation of the reservation process.

#### ***Scenario 2: Diner's Email Address Cannot Be Verified***

If the system detects that the confirmation link is either expired, invalid, or does not match the user account. An error message is displayed on the user interface, Until the diner enters verified e-mail, system keeps to Show same warning.

### Sequence Diagram for “Register As A Diner”



### [Pattern 3-Register As A Restaurant Manager]

#### Overview

The "Register as a Restaurant Manager" pattern enables individuals responsible for restaurant management to create accounts within the ReserveWell application. This process allows restaurant managers to access and utilize the platform's features for effective reservation and dining experience management.

#### Structure

##### **Restaurant Manager:**

Attributes: Name, email, password and restaurant information.

Responsibility: Initiating the registration process, entering personal details and restaurant information.

##### **System:**

Attributes: User database, communication module, authentication logic.

Responsibility: Facilitates the registration process, manages restaurant manager accounts.

#### Behavior

From a dynamic perspective, the restaurant manager collaborates with the System to enter her information, restaurant information and to get verified by the system.



## **Example**

Consider a scenario where a new restaurant manager, John Smith, wants to register with the ReserveWell application to manage reservations for his restaurant. John follows the user interface prompts, enters his information and his restaurant's information. The system updates its records, and John Smith gains access to the platform's features as a registered restaurant manager.

## **Requirement realizations**

### **View of participants**

**Diner:** Interacts with the user interface to input registration details.

**System:** Facilitates the registration process, manages user accounts, and ensures email address verification.

### **Basic Scenario**

#### ***Initiation***

The user initiates the registration process by clicking on the "Register As a Diner" button on the main screen of the ReserveWell application. The system responds by presenting the user with the registration form, prompting them to enter the required information.

#### ***Data Entry***

The user enters her information, including name, surname, e-mail. Also, he/she enters restaurant information. The system validates the entered data in real-time, providing feedback on the correctness of the information. It ensures that mandatory fields are filled, the email format is correct, and the password meets security requirements.

#### ***Confirmation and Registration***

The user reviews the entered information and clicks the "Register" button to confirm their registration. The system processes the registration request, creating a new user account in the database. The system generates a unique user ID.

#### ***Completion***

The user is now successfully registered and can log in using their credentials. The system acknowledges the successful registration, updating the user interface to reflect the new user's registration status. The user is now a recognized and authenticated member of the ReserveWell application.

### **Additional scenarios**

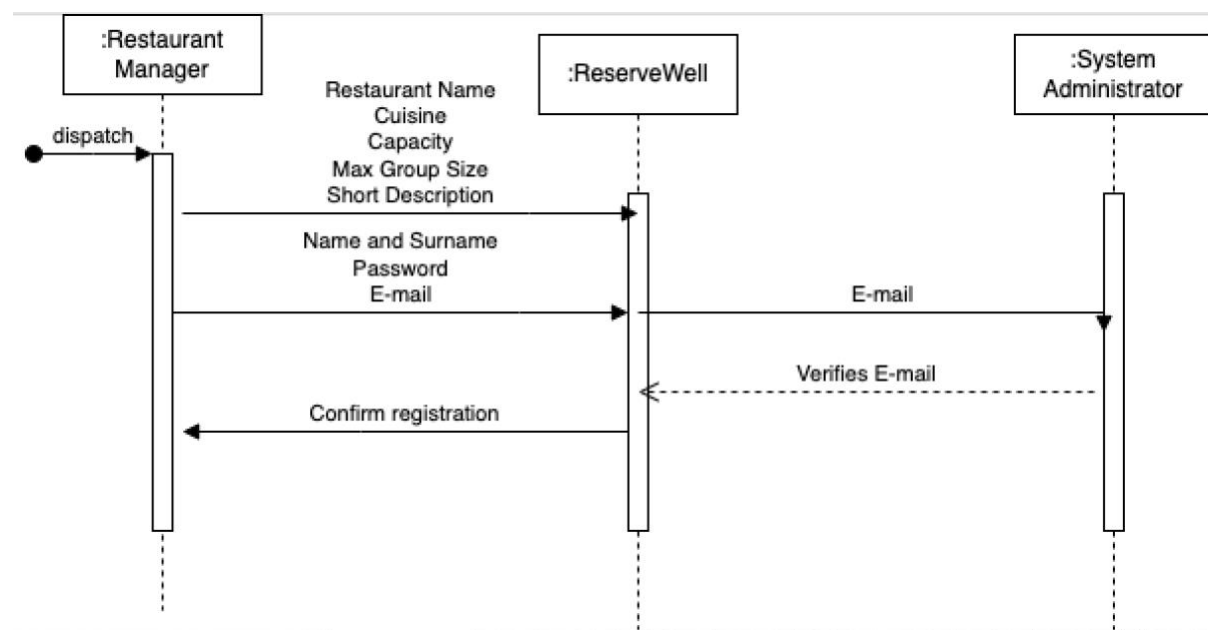
#### ***Scenario 1: System Failure and Recovery***

In the event of a system failure, the diner restarts the system, triggering a request for recovery of the prior state. The system then reconstructs the prior state, ensuring a seamless continuation of the reservation process.

### Scenario 2: Restaurant Manager's Email Address Cannot Be Verified

If the system detects that the confirmation link is either expired, invalid, or does not match the user account. An error message is displayed on the user interface, Until the restaurant manager enters verified e-mail, system keeps to show same warning.

### Sequence Diagram for “Register As A Diner”



### [Pattern 4- Edit Capacity]

#### Overview

The Capacity Edit pattern facilitates the dynamic adjustment of seating capacity in response to changing demand conditions, optimizing operational efficiency, and supporting workforce allocation planning.

#### Structure

##### **Restaurant Manager:**

**Role:** Initiates and manages the process of editing seating capacity.

**Relevant Data:** Accesses and updates seating capacity information, including the current capacity and new values for adjustment.

**Behavior:** Utilizes tools and functionalities to oversee the seating capacity editing process and ensure smooth restaurant operations.

***Waitstaff:***

**Role:** Receives real-time updates on seating capacity changes to optimize table arrangements.

**Relevant Data:** Accesses updated seating capacity information for fair distribution of seating assignments.

**Behavior:** Utilizes tools within the application to organize seating arrangements based on capacity changes.

***Diners (Customers):***

**Role:** Affected by changes in the restaurant's seating capacity, requires awareness of any modifications.

**Relevant Data:** Needs information about any changes in the restaurant's seating capacity to plan visits accordingly.

**Behavior:** Relies on accurate and up-to-date seating capacity information provided by the system to reduce inconvenience.

***Restaurant Owners:***

**Role:** Has an interest in accurate seating capacity edits for cost-effective operations, adaptability to market trends, and maintaining a positive reputation.

**Relevant Data:** Accesses historical data and logs to review changes in seating capacity for operational and strategic decision-making.

**Behavior:** Monitors the accuracy of seating capacity updates to ensure optimal restaurant performance.

***Development Team:***

**Role:** Ensures the reliability and stability of the system during and after seating capacity edits.

**Relevant Data:** Concerned with system reliability, real-time updates, mobile responsiveness, and security measures during seating capacity adjustments.

**Behavior:** Implements measures to safeguard sensitive data and maintain system stability during dynamic changes in seating capacity.

***Reservation System:***

**Role:** Central component responsible for managing seating capacity edits.

**Relevant Data:** Stores and processes information related to current and updated seating capacity.

**Behavior:** Validates and records changes in seating capacity, ensuring that the updated capacity is reflected across the system for upcoming transactions.

### ***User Interface Modules:***

**Role:** Facilitates interaction between the Restaurant Manager and the system for editing seating capacity.

**Relevant Data:** Displays current seating capacity, customization options, and confirmation messages.

**Behavior:** Presents a user-friendly interface for the Restaurant Manager to input new seating capacity, confirms changes, and communicates the success of the update.

### **Behavior**

From a dynamic perspective, the Restaurant Manager collaborates with the System to adjust seating capacity, ensuring adaptability to market trends and operational optimization.

### **Example**

Consider a scenario where a restaurant experiences a sudden increase in reservations. The Capacity Edit pattern allows the Restaurant Manager to promptly adjust seating capacity to accommodate the higher demand.

## **Requirement Realizations**

### **View of participants**

**Restaurant Manager:** Initiates and confirms capacity changes.

**System:** Facilitates capacity adjustments and updates related data in real-time.

### **Basic scenario**

#### **Restaurant Manager Initiates Capacity Edit:**

The restaurant manager interacts with the user interface to edit seating capacity and the system displays the existing capacity.

#### **Data Submission:**

The Restaurant Manager enters the new capacity and submits the update request, and the system enables adjusting the capacity based on the provided values.

#### **System Validation:**

The system validates the working hours adjustments, ensures feasibility with the existing reservations

### **Confirmation and Update:**

Upon system validation, the restaurant manager to confirm the changes, and the system updates its records with the new seating capacity.

### **Confirmation Message:**

The user interface modules present a confirmation message to the Restaurant Manager, indicating a successful seating capacity update, which will be valid for upcoming transactions.

### **Additional Scenarios**

#### **Scenario 1: Abandon Process - Restaurant Manager Quits the Page**

If the Restaurant Manager chooses to abandon the capacity editing process by quitting the page, the system offers options to discard, review, save changes, or cancel quitting. Discarding changes reconstructs the prior state, reviewing changes allows discarding or saving, and canceling quitting returns to the edit capacity page.

#### **Scenario 2: Abandon Process - Restaurant Manager Saves the Changes**

If the Restaurant Manager saves changes before quitting, the system confirms the update, displays a "Capacity Updated" message, and returns to the home page.

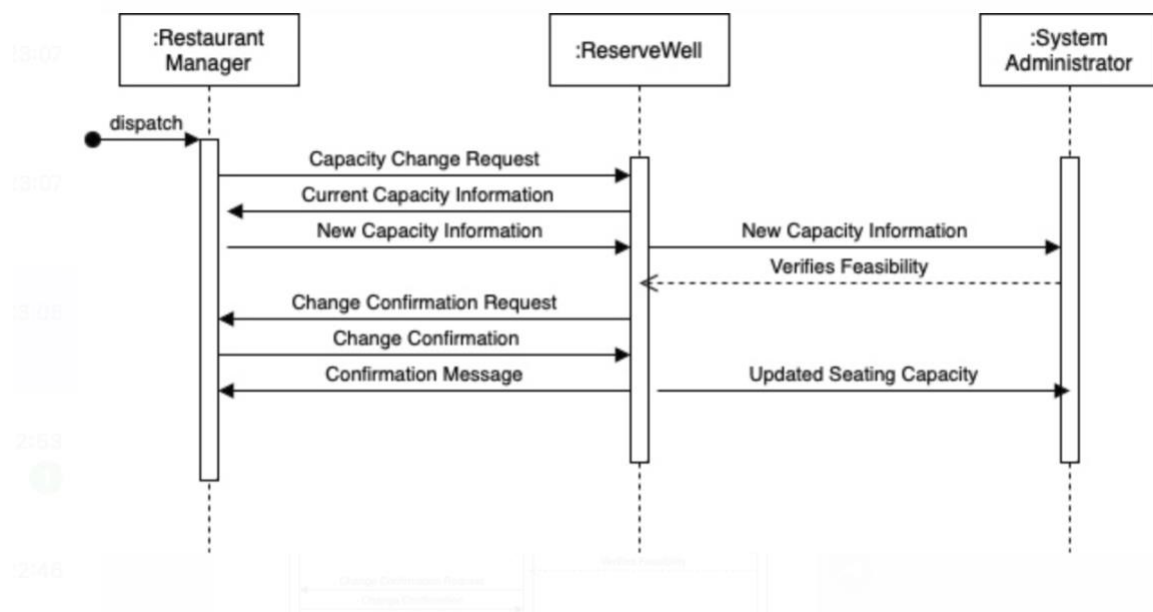
#### **Scenario 3: Abandon Process - Restaurant Manager Cancels Quitting**

Choosing to cancel quitting gracefully returns the system to the edit capacity page, enabling the Restaurant Manager to continue seamlessly.

#### **Scenario 4: System Failure and Recovery**

In case of a system failure, the Restaurant Manager restarts the system, triggering recovery. The system reconstructs the prior state unless anomalies preventing recovery are detected. In such cases, an error is signaled, recorded, and sent for review.

### **Sequence Diagram For Edit Capacity**



## [Pattern 5- Edit Work Hours]

### Overview

The Edit Working Hours pattern enables dynamic adjustments to restaurant working hours, allowing the Restaurant Manager to optimize operational efficiency, adapt to demand conditions, and plan workforce allocation effectively.

### Structure

#### **Restaurant Manager:**

**Role:** Initiates and manages the process of editing working hours.

**Relevant Data:** Accesses and updates working hours information, including the existing work hours and new values for adjustment.

**Behavior:** Utilizes tools and functionalities to oversee the working hours editing process and ensure smooth restaurant operations.

#### **Waitstaff:**

**Role:** Receives real-time updates on working hours changes and adjusts schedules accordingly to enhance workforce planning.

**Relevant Data:** Accesses updated working hours information for fair distribution of shifts.

**Behavior:** Utilizes tools within the application to organize staff schedules and maintain a healthy work-life balance.

#### **Diners (Customers):**

**Role:** Affected by changes in the restaurant's operating hours, requires awareness of any modifications.

**Relevant Data:** Needs informaton about any changes in the restaurant's working hours to plan visits accordingly.

**Behavior:** Relies on accurate and up-to-date working hours informaton provided by the system to reduce inconvenience.

#### ***Restaurant Owners:***

**Role:** Has an interest in accurate working hours edits for cost-effective operatons, adaptability to market trends, and maintaining a positive reputaton.

**Relevant Data:** Accesses historical data and logs to review changes in working hours for operational and strategic decision-making.

**Behavior:** Monitors the accuracy of working hours updates to ensure optmal restaurant performance.

#### ***Development Team:***

**Role:** Ensures the reliability and stability of the system during and aKer working hours edits.

**Relevant Data:** Concerned with system reliability, real-tme updates, mobile responsiveness, and security measures during working hours adjustments.

**Behavior:** Implements measures to safeguard sensitive data and maintain system stability during dynamic changes in working hours.

#### ***Reservation System:***

**Role:** Central component responsible for managing working hours edits.

**Relevant Data:** Stores and processes informaton related to existng and updated working hours.

**Behavior:** Validates and records changes in working hours, ensuring that the updated hours are reflected across the system for upcoming transactons.

#### ***User Interface Modules***

**Role:** Facilitates interacton between the Restaurant Manager and the system for editng working hours.

**Relevant Data:** Displays existng working hours, customizaton optons, and confirmaton messages.

**Behavior:** Presents a user-friendly interface for the Restaurant Manager to input new working hours, confirms changes, and communicates the success of the update.

#### **Behavior**

The Restaurant Manager collaborates with the System to adjust working hours dynamically, ensuring responsiveness to market trends and operational optimizaton.

#### **Example**

Consider a scenario where a restaurant decides to extend its operating hours during peak demand seasons. The Edit Working Hours pattern allows the Restaurant Manager to efficiently make these adjustments.

## **Requirement Realizations**

### **View of participants**

**Restaurant Manager:** Initiates and confirms working hours changes.

**System:** Facilitates working hours adjustments and updates related data in real-time.

### **Basic scenario**

#### **Restaurant Manager Initiates Work Hours Edit:**

The restaurant manager interacts with the user interface for editing working hours and the system displays the existing working hours.

#### **Data Submission:**

The Restaurant Manager enters the new working hours and submits the update request, and the system enables adjusting the working hours based on the provided values.

#### **System Validation:**

The system validates the working hours adjustments, ensures feasibility with the existing reservations

#### **Confirmation and Update:**

Upon system validation, the restaurant manager confirms the changes, and the system updates its records with the new working hours.

#### **Confirmation Message:**

The user interface modules present a confirmation message to the Restaurant Manager, indicating a successful working hours update, which will be valid for the upcoming transactions

### **Additional scenarios**

#### **Scenario 1: Abandon Process - Restaurant Manager Quits the Page**



If the Restaurant Manager chooses to abandon the work hours editing process by quitting the page, the system offers options to discard, review, save changes, or cancel quitting. Discarding changes reconstructs the prior state, reviewing changes allows discarding or saving, and canceling quitting returns to the edit workhours page.

### Scenario 2: Abandon Process - Restaurant Manager Saves the Changes

If the Restaurant Manager saves changes before quitting, the system confirms the update, displays a "Work Hours Updated" message, and returns to the home page.

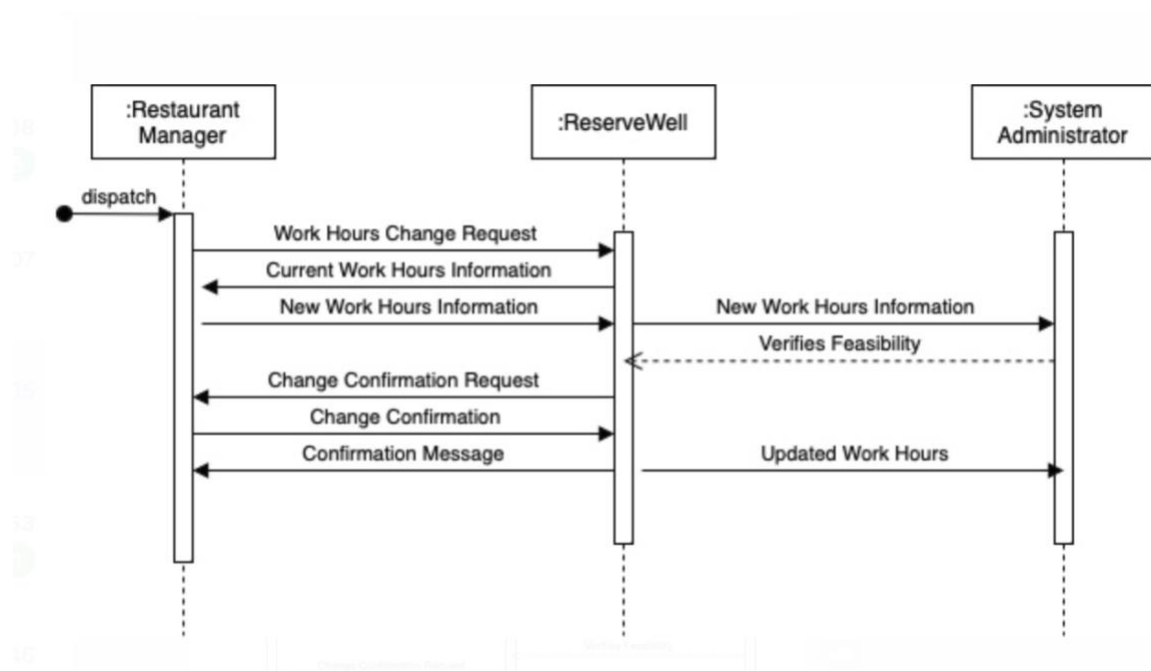
### Scenario 3: Abandon Process - Restaurant Manager Cancels Quitting

Choosing to cancel quitting gracefully returns the system to the edit work hours page, enabling the Restaurant Manager to continue seamlessly.

### Scenario 4: System Failure and Recovery

In case of a system failure, the Restaurant Manager restarts the system, triggering recovery. The system reconstructs the prior state unless anomalies preventing recovery are detected. In such cases, an error is signaled, recorded, and sent for review.

### Sequence Diagram For Edit Work Hours



Version	Date	Description
v1.0	26.11.2023	-
v1.1	25.11.2023	<ul style="list-style-type: none"><li>• According to advisor feedback, subsystems part is updated.</li><li>• Additional scenario part is revised.</li></ul>
v1.2	16.12.2023	<ul style="list-style-type: none"><li>• 4 new patterns are added to the document.</li></ul>