

Samwise Service App	Version 1.3
Use-case Specification	Date: 02.12.2023

Revisions			
Version.	Description	Date	Person
1.0	The document was created.	04.11.2023	Elif Beril Sayli Betül Ergin Özde Uysal Annie Yang
1.1	The document was revised according to Iteration 1 feedback.	11.11.2023	Elif Beril Sayli Betül Ergin Özde Uysal Annie Yang
1.2	Admin Use Cases were added.	25.11.2023	Özde Uysal Annie Yang
1.3	The document was revised according to Iteration 2 feedback.	02.12.2023	Özde Uysal Annie Yang

# Samwise Service App

## 1.Use-Case 1: Schedule Appointment

### 1.1. Brief Description

This use case describes the process of scheduling an appointment between a Service Provider and a Service Requester in the Samwise App.

### 1.2. Actor Brief Descriptions

- Actor 1: Service Requester: An authenticated user registered as a Service Requester on the Samwise App.
- Actor 2: Service Provider: An authenticated user registered as a Service Provider on the Samwise App.

### 1.3. Preconditions

- The Service Requester is logged into the Samwise App.
- The Service Provider and Service Requester have completed their Profile.
- The Service Provider has specified the service she will provide.
- The Service Provider has specified the available time slots for the service.
- The Service Requester has identified the service she wants to request.
- The Service Requester and the System must have a stable network connection.

### 1.4. Basic Flow of Events

1. The use case starts with the Service Requester having identified a service to request.
2. The Service Requester views the available time slots for the service.
3. The system shows the available time slots for the service.
4. The Service Requester selects the desired available time slot.
5. The system asks the Service Requester for additional information.

Samwise Service App	Version 1.3
Use-case Specification	Date: 02.12.2023

6. The Service Requester provides additional information.
7. The system asks the Service Requester to confirm the additional information and available time slot selection.
8. The Service Requester confirms the additional information and available time slot selection.
9. The system sends the Service Requester a confirmation of the pending scheduled appointment.
10. The system sends the Service Provider a notification of the pending scheduled appointment.
11. The use case ends.

### 1.5. Alternative Flows

\*a. At any time after Step 4, if the Service Requester cancels the service request:

1. The System asks for a cancellation confirmation.
2. The Service Requester confirms the cancellation.
3. The operation is canceled.
4. The System directs the Service Requester back to the home page.
5. The use case ends.

\*b. At any time, if the System fails:

1. The System displays an error message with the failure type.
2. The System directs the User back to the home page.
3. The use case ends.

### 1.6. Post-conditions

- The System sends a confirmation email/text of the pending scheduled appointment to the Service Requester.
- The System sends a notification of the pending scheduled appointment to the Service Provider.
- The Service Requester receives a confirmation of the pending scheduled appointment.
- The Service Provider received a notification of the pending scheduled appointment.

### 1.7. Special Requirements

- The app must ensure the privacy and security of user data.
- There must be at least 12 hours between the time Service Requester selects the available time slot and the time of the service.

Samwise Service App	Version 1.3
Use-case Specification	Date: 02.12.2023

## 2. Use-Case 2: Search Service

### 2.1. Brief Description

This use case describes how users of the Samwise Service App can utilize filter options to locate specific services that match their criteria.

### 2.2. Actor Brief Descriptions

- Actor 1: User: Any individual who uses the Samwise Service App to search for home maintenance and care services.

### 2.3. Preconditions

- The User and the System must have a stable network connection.
- The Samwise Service App is accessible on the user's device.
- There must exist at least one service in the System.

### 2.4. Basic Flow of Events

1. The use case begins when the User enters the Samwise Service App home page.
2. The System displays a search feature with filtering options.
3. The User selects one or more of the available filtering options.
4. The System applies the chosen filters to the search results.
5. The System displays results that match the Service Provider's criteria.
6. The User selects a result.
7. The System displays details about the selected result.
8. The use case ends.

### 2.5. Alternative Flows

\*a. At any time, if the System fails:

1. The System displays an error message with the failure type.
2. The System directs the User back to the home page.
3. The use case ends.

\*b. If there are no matching results:

1. The System displays a message to change the search criteria or filters.
2. The use case ends.

### 2.6. Post-conditions

- The User selects the result given by the System that matches her search criteria or filters.

### 2.7. Special Requirements

- Filtering options must reflect the service.

Samwise Service App	Version 1.3
Use-case Specification	Date: 02.12.2023

### 3.Use-Case 3: Make Payment

#### 3.1. Brief Description

The Service Requesters need to make payments for services they have received through the Samwise Service App.

#### 3.2. Actor Brief Descriptions

- Actor 1: Service Requester: The user who has received a service and needs to make a payment.
- Actor 2: Payment Authorization: Provides secure payment processing services.

#### 3.3. Preconditions

- The Service Requester and the System must have a stable network connection.
- The Samwise Service App is accessible on the Service Requester's device.
- The Service Requester is logged into their account on the app.
- The Service Requester has an Approved Scheduled Service.
- The Service Requester received the service and is ready to make a payment.

#### 3.4. Basic Flow of Events

1. The use case begins when the Service Requester navigates to the section of the Samwise Service App where payment is made.
2. The Service Requester selects the specific service for which they want to make a payment.
3. The System displays the details of the selected service.
4. The Service Requester selects her preferred payment method.
5. The System displays payment method fields.
6. The Service Requester provides the required payment information.
7. The System asks the Service Requester to confirm the payment information.
8. The Service Requester confirms the payment information.
9. The System verifies the payment method.
10. The System processes the payment using a Payment Authorization.
11. The System displays the Payment Confirmation to the Service Requester.
12. The System sends a Payment Confirmation to the Service Requester.
13. The System sends a Payment Notification to the Service Provider.
14. The use case ends.

#### 3.5. Alternative Flows

\*a. At any time before Step 9, if the Service Requester cancels payment:

1. The System asks for a cancellation confirmation.
2. The Service Requester confirms the cancellation.
3. The System cancels the operation.
4. The System directs the Service Requester back to the payment details page.
5. The use case ends.

\*b. At Step 9, if the Payment Method Not Valid:

1. The System displays an error message.
2. The use case returns to Step 4 until success.

\*c. At Step 10, if Payment Processing Failure:

1. The System displays an error message.

Samwise Service App	Version 1.3
Use-case Specification	Date: 02.12.2023

2. The use case returns to Step 4 until success.

### 3.6. Post-conditions

- The Service Requester successfully makes a payment for the received service.
- A Payment Confirmation has been provided to the Service Requester.
- The Service Provider receives a Payment Notification.

### 3.7. Special Requirements

- Payment processing must be secure and follow industry standards.
- In case of payment method issues or processing failures, appropriate error handling and guidance should be provided to the Service Requester.

Samwise Service App	Version 1.3
Use-case Specification	Date: 02.12.2023

## 4. Use-Case 4: Manage Time Slot

### 4.1. Brief Description

This use case describes the process of managing the Time Slot for Service Providers on the Samwise app. Service providers can create, edit, delete, and organize their Time Slot to efficiently manage their provided services.

### 4.2. Actor Brief Descriptions

4.2.1. Service Provider: An authenticated user registered as a Service Provider on the Samwise app.

### 4.3. Preconditions

4.3.1. The Service Provider is logged into the Samwise app.

4.3.2. The Service Provider has completed their registration and profile setup.

4.3.3. The service associated with the Time Slot must be available (not scheduled by Service Requester yet).

### 4.4. Basic Flow of Events

1. The use case begins when the Service Provider logs into the app and navigates to their dashboard.
2. The Service Provider selects the option to organize time slots in their dashboard.
3. The app displays the existing Time Slot of the Service Provider, if any.
4. The Service Provider can choose to organize, create, edit, or delete Time Slots.  
The Service Provider repeats steps 2-4 until the Service Provider indicates done.
5. The use case ends.

### 4.5. Alternative Flows

#### \*a. At any time, Service Provider cancels the operation:

1. Service provider presses the button to cancel the operation.
2. The system takes the changes back.

#### \*b. At any time, if the System fails:

4. The System displays an error message with the failure type.
5. The System directs the User back.
6. The use case ends.

### 4.6. Key Scenarios

#### 4.6.1. Organizing Time Slot

1. The Service Provider selects "Organize Time Slot."
2. The Service Provider rearranges or groups existing Time Slot.
3. The Service Provider saves changes and exits Time Slot.

#### 4.6.2. Creating Time Slot

1. The Service Provider selects "Create Time Slot."
2. The Service Provider enters a new Time Slot, including date, time, and availability.
3. The Service Provider enters the saves and exits the Time Slot.

Samwise Service App	Version 1.3
Use-case Specification	Date: 02.12.2023

#### 4.6.3. Editing Time Slot

1. The Service Provider selects "Edit Time Slot" and selects an existing Time Slot.
2. The Service Provider sees the details of the selected Time Slot.
3. The Service Provider can make necessary changes, such as modifying time slots or updating availability.
4. The Service Provider saves the edited Time Slot and exits the Time Slot.

#### 4.6.4. Deleting Time Slot

1. The Service Provider selects "Delete Time Slot" and selects an existing Time Slot that they wish to remove.
2. The Service Provider confirms the deletion.
3. The Service Provider saves and exits Time Slot.

### 4.7. Post-conditions

4.7.1. Time Slot adjustments, including organization, creation, editing, and deletion, are stored in the app.

4.7.2. The Service Provider has up-to-date Time Slots to effectively manage service creations and scheduled appointments made by Service Requesters.

### 4.8. Special Requirements

4.8.1. Time Slot adjustments should be reflected in real-time to ensure accurate scheduling of services and appointments.

4.8.2. The privacy and security of Time Slot data must be maintained to protect Service Providers' availability information.

Samwise Service App	Version 1.3
Use-case Specification	Date: 02.12.2023

5.

## Use-Case 5: Manage User Account

### 5.1 Brief Description:

This use case describes how unregistered users can create a new account to use the system.

### 5.2. Actor Brief Descriptions

- End Users: An unregistered user who desires to create a new account to use the system.

### 5.3. Preconditions

- The User and the System must have a stable network connection.
- The User must have access to Samwise Service App.
- The User must not have an account in the system.

### 5.4. Basic Flow of Events

1. The use case begins when the User accesses the application.
2. The System displays a page to create a new account by entering in account information.
3. The User inputs account information.
4. The System displays the account information.
5. The Systems asks the User to confirm and submit.
6. The User confirms and submits.
7. The System sends a verification to the User's contact information.
8. The System displays a verification screen for the User to enter a verification.
9. The User enters the verification.
10. The System verifies the User's identity.
11. The System directs the User to the sign in page.
12. The use case ends.

### 5.5. Alternative Flows

\*a. At any time, if the Service Requester cancels the service request:

1. The System asks for a cancellation confirmation.
2. The Service Requester confirms the cancellation.
3. The operation is canceled.
4. The System directs the Service Requester back to the home page.
5. The use case ends.

\*b. At any time, if the System fails:

1. The System displays an error message with the failure type.
2. The System directs the User back to the home page.
3. The use case ends.

### 5.6. Post-conditions

- The User is a registered User and has an account in the system with a unique username.

### 5.7. Special Requirements

- The app must ensure the privacy and security of user data.



Samwise Service App	Version 1.3
Use-case Specification	Date: 02.12.2023

6.

## Use-Case 6: Ask Help

### 6.1 Brief Description:

The chat support service facilitates users in obtaining immediate assistance and information regarding the Samwise Service App. It ensures swift issue resolution, provides guidance on app functionalities, and enriches the overall user experience through real-time interaction with support representatives. The user successfully engages with the chat support service, receives prompt assistance, and either resolves the issue or gains necessary information, resulting in a positive user experience.

## 7. Use-Case 7: Write Review

### 7.1 Brief Description:

This use case entails the functionality in the Samwise Service App that allows Service Requesters to provide feedback and reviews for the services they've received and the Service Providers who have fulfilled those requests. Requesters can share their experiences, rate the services, and offer comments to inform other users and enhance the overall quality of the app's services.

## 8. Use-Case 8: Make Proposal

### 8.1 Brief Description:

This use case describes the process where Service Requesters, specifically Premium users, can make proposals for discounts on the service price, while Service Providers can review, accept, modify the proposed discount amount, or reject the proposal. The Premium Service Requester successfully submits a discount proposal, and after negotiation or acceptance by the Service Provider, a mutually beneficial discount is agreed upon. This agreement results in a reduced service price, enhancing the requester's satisfaction and fostering positive interactions between both parties.

## 9. Use-Case 9: Manage Service

### 9.1. Brief Description:

This use case describes the process where Service Providers create, edit and delete the services they provide. While creating the service, Service Providers mention the service type they provide and available times. Service providers can edit this service type and available times and also delete the service.

### 9.2. Actor Brief Description:

9.2.1. Service Provider: An authenticated user registered as a Service Provider on the Samwise app.

### 9.3. Preconditions:

- 9.3.1. Service provider is logged into the Samwise Service App.
- 9.3.2. The Service Provider has completed their registration and profile setup.
- 9.3.3. Provider has access permissions to manage services.

### 9.4. Basic Flow of Events:

1. Service provider navigates to the Service Management section.

Samwise Service App	Version 1.3
Use-case Specification	Date: 02.12.2023

2. Provider fills the areas of Service Details.
3. Provider presses the button to create the service.
4. The system generates a UUID for the created service.

## 9.5. Alternative Flows:

### 4.a. Service provider needs to edit service information:

1. Provider enters Service UUID of the service to be edited.
2. Provider enters Service Details which are going to be edited.
3. Provider presses the button to edit service.
  - 3.a. Service UUID not found:
    1. The system gives an error stating that Service UUID does not exist.
    2. Provider enters a new UUID.
4. The system updates the service information with the edits made.

### 4.b. Service provider needs to delete the service:

1. Provider enters Service UUID of the service to be deleted.
2. Provider presses the button to delete the service.
  - 2.a. Service UUID not found:
    1. The system gives an error stating that Service UUID does not exist.
    2. Provider enters a new UUID.
3. The system deletes the service.

## 9.6. Key Scenarios:

### 9.6.1. Creating a New Service:

1. Provider initiates the creation process.
2. Provider enters service details.
3. System generates a unique Service UUID for the new service.

### 9.6.2. Editing Service Details:

1. Provider selects an existing service for modification.
2. Modifies service type and/or available times as needed.
3. System updates service information according to the changes made.

### 9.6.3. Deleting a Service:

1. Provider selects a service for removal.
2. Confirms deletion to remove the service and associated data.

## 9.7. Post Conditions

- Created services display in the provider's list of offerings.
- Edited service details accurately reflect in the System.

Samwise Service App	Version 1.3
Use-case Specification	Date: 02.12.2023

- Deleted services are removed from the provider's offerings.

## 9.8. Special Requirements

- Error handling mechanisms to ensure data integrity during creation, editing, and deletion.

## 10. Use-Case 10: Receive Payment

### 10.1. Brief Description:

This use case describes the process where Service Providers receive payment for the service they provide. The payment is made by a Service Requester using the Use-Case 3: Make Payment. The Service Provider must provide their financial information to receive the payment. The payment amount is determined by the service that is provided. The Service Provider successfully provides the necessary financial information and receives the payment for the delivered service. The system confirms the successful transaction, updating both parties' payment history and maintaining a record of the completed payment, ensuring transparent and secure payment processing.

## 11. Use-Case 11: Analyze System Performance

### 11.1. Brief Description:

This use case entails the System Admin utilizing specialized monitoring tools to analyze various system performance metrics. The objective is to identify performance bottlenecks, trends, and potential optimization areas within the Samwise Service App. The System Admin successfully identifies performance bottlenecks, discerns root causes, and formulates effective optimization strategies. Recommendations are communicated to relevant teams, leading to system enhancements and improved overall performance of the Samwise Service App.

## 12. Use-Case 12: Troubleshoot System

### 12.1. Brief Description:

This use case involves the System Admin investigating reported incidents, utilizing system logs, and executing corrective actions to address system issues promptly. The objective is to ensure continuous functionality and resolve any operational disruptions within the Samwise Service App. The System Admin effectively investigates reported incidents, identifies root causes, and implements corrective actions. The system's functionality is restored, and the incidents are resolved, ensuring uninterrupted operation of the Samwise Service App.

Samwise Service App	Version 1.3
Use-case Specification	Date: 02.12.2023

### 13. Use Case 13: Approve Appointment

#### 13.1 Brief Description:

This use case describes the possible actions the Service Providers can take after their service has been scheduled by a Service Requester.

#### 13.2. Actor Brief Descriptions

- Actor 1: Service Requester: An authenticated user registered as a Service Requester on the Samwise App.
- Actor 2: Service Provider: An authenticated user registered as a Service Provider on the Samwise App.

#### 13.3. Preconditions

- The User and the System must have a stable network connection.
- The User must have access to Samwise Service App on .
- The Service Provider is logged into the System.
- The Service Provider received a notification for a Pending Scheduled Appointment.

#### 13.4. Basic Flow of Events

1. The use case begins after the Service Provider logs into the System.
2. The system shows a notification that the Service Provider has a Pending Scheduled Appointment that needs approval.
3. The Service Provider views the Pending Schedule appointment details.
4. The System presents options to Approve or Decline
5. The Service Provider selects Approve.
6. The System asks the Service Provider to confirm the selection.
7. The Service Provider confirms the selection.
8. The System changes the scheduled appointment status from Pending to Approved.
9. The System sends a confirmation to the Service Provider.
10. The System sends a notification to the Service Requester.
11. The use case ends.

#### 13.5. Alternative Flows

\*a. At any time, if the Service Requester cancels the service request:

1. The System asks for a cancellation confirmation.
2. The Service Requester confirms the cancellation.
3. The operation is canceled.
4. The System directs the Service Requester back to the home page.
5. The use case ends.

\*b. At any time, if the System fails:

1. The System displays an error message with the failure type.
2. The System directs the User back to the home page.
3. The use case ends.

Samwise Service App	Version 1.3
Use-case Specification	Date: 02.12.2023

\*c. At Step 5, if the Service Provider selects Decline:

1. The System asks the Service Provider to confirm the selection.
2. The Service Provider confirms the selection.
3. The System changes the scheduled appointment status from Pending to Denied.
4. The System sends a confirmation to the Service Provider.
5. The System sends a notification to the Service Requester.
6. The use case ends.

#### **13.6. Post-conditions**

- The system changes the status of the Pending Scheduled Appointment.
- The system sends a confirmation to the Service Provider of the Pending Scheduled Appointment decision.
- The Service Provider receives the decision confirmation.
- The System sends a notification to the Service Requester of the Pending Scheduled Appointment decision..
- The Service Requester receives the decision notification.

#### **13.7. Special Requirements**

- The app must ensure the privacy and security of user data.