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
1.4	The document was updated for Iteration 3.	16.02.2023	Elif Beril Sayli Özde Uysal Annie Yang

Samwise Service App
Use-Case Specification

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

- Service Requester: An authenticated user registered as a Service Requester on the Samwise App.
- 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 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 arriving at the details page of a service she wants to request.
2. The Service Requester clicks the available time for the service.
3. The System shows the available time for the service.
4. The Service Requester selects the desired available time.
5. The System asks the Service Requester for additional information.
6. The Service Requester provides additional information.
7. The System asks the Service Requester to confirm the additional information and available time selection.
8. The Service Requester confirms the additional information and available time selection.
9. The System sends the Service Requester a confirmation of the pending scheduled appointment.
10. The System sends the Service Provider a confirmation 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 displays a message for the cancellation.
5. The System directs the Service Requester back to the home page.
6. 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 confirmation 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 confirmation 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 and the time of the service.

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

- 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.

4.a. 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.

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

- Service Requester: The user who has received a service and needs to make a payment.
- 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 Confirmation 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 displays a message for the cancellation.
5. The System directs the Service Requester back to the payment details page.
6. The use case ends.

9.a. If the payment method is not valid:

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

10.a. If there is a payment processing failure:

1. The System displays an error message.
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 confirmation.

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.

4.Use-Case 4: Manage Proposal

4.1. Brief Description

This use case describes actions taken by a Service Provider after they receive a proposal request for a discounted service price. The Service Provider can accept the discounted price, reject the discounted price, or propose a counter price. If the Service Provider accepts the proposal, the Service Requester pays the discounted price. However, if the Service Provider rejects the proposal, the Service Provider can enter a counter price. After sending a counter price, the Service Requester can reject or accept the counter price to finalize the pending schedule appointment.

4.2. Actor Brief Descriptions

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

4.3. Preconditions

- The Users and the System must have a stable network connection.
- The service provided by the Service Provider must be available.
- The Premium Service Requester must have created and submitted a service price discount proposal.
- The Service Provider must have received a proposal confirmation.

4.4. Basic Flow of Events

1. The use case starts with the Service Provider after they received a proposal confirmation for the service they are providing.
2. The System asks the Service Provider if they want to Accept or Reject the discount proposal.
3. The Service Requester selects Accept the discount proposal and confirms.
4. The System displays a message confirming the Service Requester’s selection.
5. The System sends the Service Provider a confirmation of the approved scheduled appointment and appointment details (price, date, and time).
6. The System sends the Service Requester a confirmation of the approved scheduled appointment and appointment details (price, date, and time).
7. The use case ends.

4.5. 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 Service Provider back to their dashboard.
3. The use case ends.

3.a. If the Service Provider selects Reject and confirms the selection:

1. The System asks the Service Provider to enter a counter price and confirm.
2. The Service Provider enters a counter price and confirms.
3. The System sends the Service Provider a confirmation of the pending scheduled appointment and counter price.
4. The System sends the Service Requester a notification of the pending scheduled appointment and counter price.
5. The use case ends.

4.6. Post-conditions

- The System sends a confirmation of the approved scheduled appointment with the appointment details (price, date, and time) to the Service Requester.
- The System sends a confirmation of the approved scheduled appointment with the appointment details (price, date, and time) to the Service Provider.
- The Service Requester receives a confirmation of the approved scheduled appointment.
- The Service Provider receives a confirmation of the approved scheduled appointment.

4.7. 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 and the time of the service.

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. The objective registered account is required to fully utilize scheduling and service providing features in the Samwise Service App.

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.

6. Use-Case 6: Ask Help

6.1 Brief Description:

The chat support service aims to provide users with immediate assistance and information regarding the Samwise Service App. It facilitates issue resolution, provides guidance on app functionalities, and enriches the overall user experience through real-time interaction with support representatives. The user can ask the chat support service questions and receive prompt assistance. The objective is to provide self-service to resolve process related questions to enable easy and understandable use of the application.

6.2. Actor Brief Descriptions

- User: An unregistered user or registered user (Service Requester or Service Provider) who desires to gain information on using Samwise Service App.

6.3. Preconditions

- The User and the System must have a stable network connection.
- The User must have access to Samwise Service App.

6.4. Basic Flow of Events

1. The use case begins when the User accesses the chat support page.
2. The System asks the User how it can help the User through a chat feature.
3. The User enters the question or keywords.
4. The System stores the question or keywords.
5. The System queries the database.
6. The System returns responses that match the User's question or keywords.
   The User repeats steps 3-6 until the User indicates done.
7. The System asks for confirmation to end the chat session.
8. The User confirms the end of the session.
9. The System asks the User to enter an email address to send the chat transcript.
10. The User provides their email address and submits.
11. The System sends the chat transcript email to the provided email address.
12. The System directs the User back to the home page.
13. The use case ends.

6.5. Alternative Flows

*a. At any time, if the User cancels the chat support:

1. The System asks for confirmation to end the chat session.
2. The User confirms the end of the session.
3. The System asks the User to enter an email address to send the chat transcript.
4. The User provides their email address and submits.
5. The System sends the chat transcript email to the provided email address.
6. The System directs the User back to the home page.
7. 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.

*c. At any time, if the System can't find results that match the Users questions or keywords:

1. The System returns a message for the User to rephrase the question.
2. Return to step 3.

6.6. Post-conditions

- The System sends a chat transcript to the User's supplied email address.
- The User receives a chat transcript to their supplied email address.
- The System stores the questions and keywords to enable analysis of customer support topics.

6.7. Special Requirements

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

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 ratings and reviews for the services they've received and the Service Providers who have fulfilled those requests. Service Requesters can rate the services and offer comments to inform other users and enhance the overall quality of the app's services.

7.2. Actor Brief Descriptions

- Service Requester: An authenticated user registered as a Service Requester on the Samwise App.

7.3. Preconditions

- The User and the System must have a stable network connection.
- The Service Requester must have access to Samwise Service App.
- The Service Requester must have received at least one service from a Service Provider.

7.4. Basic Flow of Events

1. The use case begins when the Service Requester accesses the service appointment page.
2. The Service Requester selects the service to review.
3. The System displays an integer rating field and a comment field.
4. The Service Requester inputs an integer and comment(s) into the respective fields.
5. The Service Requester submits the form.
6. The System asks for submission confirmation.
7. The Service Requester confirms.
8. The System displays a message of the successful review submission.
9. The System directs the Service Requester to the dashboard.
10. The use case ends.

7.5. Alternative Flows

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

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 their dashboard .
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 their dashboard.
3. The use case ends.

7.6. Post-conditions

- The System calculates the average rating for the Service Provider’s profile.
- The System displays the review and rating on the Service Provider’s profile.
- The System stores the rating and review.

7.7. Special Requirements

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

8. Use-Case 8: Make Proposal

8.1 Brief Description:

This use case describes the process where a Service Requester, specifically Premium Service Requesters user, can make a proposal for a discounted service price. Then a Service Provider can review, accept, modify the proposed discount amount, or reject the proposal as described in Use-Case 4: Manage Proposal. The Premium Service Requester successfully submits a service price proposal, and after acceptance by the Service Provider, a discounted price is agreed upon. This agreement results in a reduced service price, enhancing the Service Requester's purchasing power.

8.2. Actor Brief Descriptions

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

8.3. Preconditions

- The Users and the System must have a stable network connection.
- The Service Requester must have access to Samwise Service App.
- The Service Requester must have a Premium account.
- The Service Requester has identified the service she wants to request.
- The service must be available.

8.4. Basic Flow of Events

1. The use case starts with the Service Requester after they select the desired available time for a service.
2. The System asks the Service Requester if they want to make a discount proposal.
3. The Service Requester provides a proposed price.
4. The System asks the Service Requester to confirm the information (proposed price, available time selection, and additional information).
5. The Service Requester confirms the information.
6. The system sends the Service Requester a confirmation of the pending scheduled appointment.
7. The system sends the Service Provider a confirmation of the pending scheduled appointment.
8. The use case ends.

8.5. Alternative Flows

*a. At any time, if the Service Requester cancels the discount proposal:

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 their dashboard.
3. The use case ends.

8.6. Post-conditions

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

- The System sends a confirmation 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 confirmation of the pending scheduled appointment.

8.7. 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 and the time of the service.

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 indicate 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:

- Service Provider: An authenticated user registered as a Service Provider on the Samwise Service App.

9.3. Preconditions:

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

9.4. Basic Flow of Events:

1. Service provider navigates to the Service Management section.
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 the 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. Post Conditions

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

9.7. 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 outlines the process where Service Providers receive payment for the services they have provided. The payment is initiated by a Service Requester through Use Case 3: Make Payment. The Service Provider is required to enter their financial details to receive the payment. The payment amount is based on the service requested. Successful payment confirmation updates both parties' payment history, maintaining a transparent and secure record of completed transactions.

10.2. Actor Brief Description:

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

10.3. Preconditions:

- The Service Provider is logged into the Samwise Service App.
- The Service Provider has fulfilled the required service to the Service Requester.
- A Service Requester has initiated payment for the provided service through Use Case 3: Make Payment.

10.4. Basic Flow of Events:

1. The Service Requester initiates the payment process for the provided service.
2. Service Requester enters payment details via Use Case 3: Make Payment.
3. The System receives the payment request and confirms the amount.
4. The Service Provider provides her financial information for receiving payment.
5. The System processes the payment and transfers the due amount to the Service Provider's account.
6. The System displays the confirmation of successful payment to both Service Provider and Service Requester.
7. The system updates payment history for both parties to maintain a record of the completed transaction.

10.5. Alternative Flows:

*a. At any time, if there is a failure in processing the payment due to technical issues or connectivity problems:

1. The system prompts for reattempt or alternative payment methods.
2. The system refreshes the page.
3. The use case ends.

4.a. The Service Provider enters her financial information wrong:

1. The System gives an error.
2. The System refreshes the page so the Service Provider can re-enter her financial information.
3. Continue to Step 4 in Basic Flow of Events.

10.6. Post Conditions

- Service Provider's account reflects the received payment amount.

- Both Service Provider and Service Requester have a record of the completed transaction in their payment history.

10.7. Special Requirements

- Error handling mechanisms to address payment processing failures.
- Secure and transparent record-keeping of completed payment transactions.

11.Use-Case 11: Troubleshoot System

11.1. Brief Description:

This use case entails the System Admin investigating reported incidents, utilizing system logs, and executing corrective actions to address system issues promptly within the Samwise Service App. The primary objective is to ensure continuous functionality and resolve any operational disruptions. The System Admin 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.

11.2. Actor Brief Description:

- System Admin: The IT personnel responsible for overseeing and managing the technical aspects of the application's infrastructure and ensuring its smooth operation.

11.3. Preconditions:

- The System Admin has access to system logs, incident reports, and relevant troubleshooting tools.
- The Samwise Service App is operational but may be experiencing technical errors.

11.4. Basic Flow of Events:

1. System Admin receives incident reports or identifies system anomalies.
2. The System Admin accesses system logs to investigate reported incidents.
3. The System Admin analyzes system logs, error messages, and user-reported issues to identify root causes.
4. System Admin formulates and executes corrective actions to resolve identified issues.
5. Upon successful implementation of corrective measures, the System Admin verifies that the issues have been resolved.

11.5. Alternative Flows:

4.a. If initial corrective actions fail to resolve the issue:

1. The System Admin iterates through further troubleshooting steps or seeks additional expertise to address the problem.
2. The System Admin takes new corrective actions according to new information gathered.

11.6. Post Conditions

- The reported incidents are resolved, ensuring uninterrupted operation of the Samwise Service App.
- System Admin documents corrective actions and resolutions for future reference.

11.7. Special Requirements

- Logging mechanisms to facilitate incident investigation.
- Knowledge base or documentation of previous incidents and resolutions for reference.

12.Use Case 12: Approve Appointment

12.1 Brief Description:

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

12.2. Actor Brief Descriptions

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

12.3. 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 confirmation for a Pending Scheduled Appointment.

12.4. Basic Flow of Events

1. The use case begins after the Service Provider logs into the System.
2. The system shows a confirmation 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 confirmation to the Service Requester.
11. The use case ends.

12.5. 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.a. 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 confirmation to the Service Requester.
6. The use case ends.

12.6. 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 confirmation to the Service Requester of the Pending Scheduled Appointment decision.
- The Service Requester receives the decision confirmation.

12.7. Special Requirements

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