Software Requirements Specification

for

StyleNest

Version 1.0 approved

Prepared by Caitlin Moeykens and Clara Scalzer

Team Dressed Up

10/8/2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4 Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4. System Features

4.1 Outfit Selection and Recommendations

4.1.1 Description and Priority

Provide users with outfit recommendations based on their preferences, moods, styles, and weather- High priority

4.1.2 Stimulus/Response Sequences

Stimulus: User inputs their location and mood/style preferences
Response: The system generates 3-4 outfit suggestions that match the weather and user's mood/style
Stimulus: User requests new outfit suggestion
Response: The system provides another set of outfit suggestions

4.1.3 Functional Requirements

REQ-1: The system shall allow users to enter their location to provide weather based outfit suggestions.
REQ-2: The system shall generate 3-4 outfit suggestions based on the user's mood, style, and current weather.
REQ-3: The system shall provide an easy to use option to refresh the recommendations and generate new outfits quickly

4.2 Closet and Outfit Organization

4.2.1 Description and Priority

Allows users to organize outfits by categories like occasion, emotion, or event and view their entire wardrobe- Medium priority

4.2.2 Stimulus/Response

Stimulus: User categories outfits by occasion or emotion

Response: The system organizes and stores these outfits for easy access later

Stimulus: User views their wardrobe

Response: The system displays all items in the user's wardrobe to help with outfit selection.

4.1.3 Functional Requirements

REQ-1: The system shall provide users with the ability to organize outfits by occasion, emotion, or style.
REQ-2: The system shall show a visual display of all wardrobe items to prevent forgetting about clothing pieces.
REQ-3: The system shall keep a record of which outfits the user felt most comfortable or confident wearing.

REQ-4: The system shall let users tag individual clothing items with custom labels or keywords for easy searching

REQ-5: The system shall provide suggestions to the user for organizing their items based on seasonal items.

4.3 Event-Based Outfit Coordination

4.3.1 Description and Priority

Allows users to be invited to events, view what other attendees are planning to wear, and showcase their own planned outfit for the event, Medium priority.

4.3.2 Stimulus/Response Sequences

Stimulus: User receives an event invitation and views details in the system
Response: The system displays the list of attendees and their planned outfits for the event.
Stimulus: User shares their planned outfit for the event
Response: The system updates the event page to include the user's outfit selection.

4.3.3 Functional Requirements

REQ-1: The system shall allow users to receive event invitations directly within the app.

REQ-2: The system shall display a list of other attendees for each event, along with their planned outfits if shared.

REQ-3: The system shall allow users to upload or select a planned outfit for the event and make it visible to other attendees.

REQ-4: The system shall notify users of any changes to the event details or if other attendees update their outfit choices.

REQ-5: The system shall allow users to RSVP to events directly through the app and specify their outfit preferences for the event.

REQ-6: The system shall notify users when other attendees have reacted to their planned outfits for an event.

4.4 Outfit Scheduling Calendar

4.4.1 Description and Priority

Allows users to schedule their outfits for specific days using a calendar feature. Medium Priority

4.4.2 Stimulus/Response Sequences

Stimulus: User selects a date on the calendar to schedule an outfit.
Response: The system opens a page to plan and save the selected outfit for that day.
Stimulus: User checks the calendar for their outfit schedule for the week.
Response: The system displays the scheduled outfits for each day in a clear and organized manner

4.4.3 Functional Requirements

REQ-1: the system shall provide a calendar view that allows users to schedule outfits for specific dates.
REQ-2: The system shall allow users to select and save outfits to the calendar in advance, showing details like the date, weather, and event.
REQ-3: The system shall allow users to edit or update their planned outfits on the calendar easily.
REQ-4: The system shall provide reminders to users about their scheduled outfits for the next day.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Outfit Suggestion Latency

The system shall generate outfit suggestions within 3 seconds of user interaction. This metric is important for user satisfaction , especially for time-sensitive situations such as morning routines.

5.1.2 Search Query Response Time

Search operations for clothing items shall return results in under 1 second. Quick and efficient search is essential for quick decisions for outfit choices.

5.1.3 Current Weather Data

The application shall refresh weather data within 5 seconds to ensure outfit recommendations are appropriate for current conditions and users can make informed, comfortable choices.

5.1.4 Previous Outfit Retrieval

Users shall retrieve previously worn outfits within 2 seconds. Quick access to this data will encourage engagement and aid in outfit planning.

5.2 Safety Requirements

5.2.1 Clothing Swap Safety Features

The application shall implement user ratings and feedback mechanisms for individuals participating in clothing swaps, ensuring users can make informed decisions about whom to meet.

5.2.2 Emergency Reporting Features

Users shall have a straightforward way of reporting negative experiences or safety concerns, triggering an administrative review to address issues promptly.

5.2.3 Regulatory Compliance

The application shall adhere to relevant safety and data protection regulations, ensuring user safety and data integrity.

5.3 Security Requirements

5.3.1 Data Encryption Protocols:
All user data must be encrypted, ensuring protection against unauthorized access and data breaches.

5.3.2 Privacy Controls Customization:
Users must have privacy settings that allow them to control who can view their outfit collections and personal data, allowing them to manage their information securely.

5.3.3 Compliance with Security Standards:
The application must comply with industry security standards and regulations, to safeguard user data and maintain trust.

5.4 Software Quality Attributes

5.4.1 Usability Metrics:
The application shall achieve a usability score of at least 85% in user testing, ensuring an intuitive interface that facilitates efficient navigation and interaction.

5.4.2 System Reliability:
The application must maintain an uptime of 99.9%, ensuring consistent availability for users at all times.

5.4.3 Maintainability Standards:
Code must adhere to clean coding practices, enabling efficient updates and bug fixes, and reducing the time required for future enhancements.

5.4.4 Feature Flexibility:
The architecture must support seamless integration of new features, such as mood-based outfit recommendations, without necessitating significant codebase alterations.

5.4.5 Testing Coverage:
The application must achieve at least 90% test coverage across the codebase, ensuring reliability and facilitating rapid identification and resolution of issues.

5.5 Business Rules

5.5.1 User Role Management:
Only registered users can create, share, and rate outfit suggestions.

5.5.2 Outfit Sharing Permissions:
Users can view and comment on outfits shared by friends only if they have granted explicit access, preserving user privacy.

5.5.3 Clothing Swap Eligibility:
Users can initiate clothing swaps only with registered friends, fostering a secure environment for exchanges.

5.5.4 Content Moderation Protocols:

User-generated content must adhere to established community guidelines. Inappropriate content will be subject to review and potential removal by application administrators.

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>