Software Requirements Specification

SRS

MisMatch

DOCUMENT HISTORY

Version	Status	Date	Responsible person	Reason for change
1.0	Draft	04.11.2025	Sabhyeta Khadka, Nancy Mroke, Rana Soliman, Fabio Jindrak	Started Requirements and Use case diagram
1.1	Draft	10.11.2025	Sabhyeta Khadka, Rana Soliman	Finished creating functional Requirements und Use Case Diagram as well as 6 use case descriptions.
1.2	Draft	12.11.2025	Nancy Mroke, Fabio Jindrak	Non-functional requirements created. Final edits made.
1.3	Final	25.11.2025	Fabio Jindrak, Sabhyeta Khadka, Nancy Mroke, Rana Soliman	Revisions made to the functional and non-functional requirements, use case diagram and descriptions edited.

INTRODUCTION

Purpose

This document specifies the software requirements for a virtual wardrobe web application that enables users to create and visualize outfit combinations using a collection of common clothing items. The system allows manual mix-and-match functionality and occasion-based suggestions.

Product scope / Goals

The application will provide:

- A digital wardrobe interface with pre-loaded clothing items
- Manual outfit creation through selection mechanisms
- 2D visualization of outfit combinations
- Optional occasion-based outfit filtering

Definitions, Acronyms, Abbreviations

- **SRS:** Software Requirements Specification
- **MoSCoW:** Must have, Should have, Could have, Won't have (prioritization method)
- **API:** Application Programming Interface
- **UI:** User Interface
- **2D:** Two-dimensional

References

- Rupp's Requirements Template Methodology
- MoSCoW Prioritization Framework

PRODUCT APPLICATION

Target Audience & Background

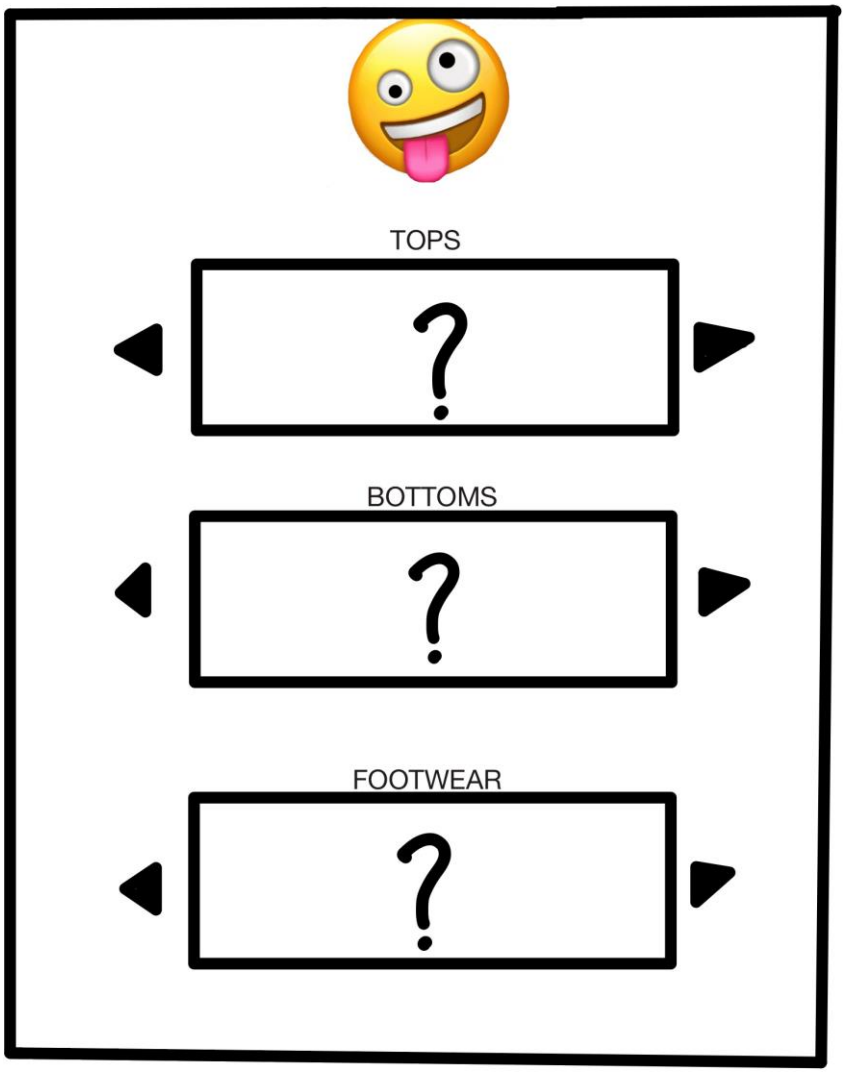
MisMatch is designed for everyday individuals who experience decision fatigue when choosing daily outfits and want a simple solution to streamline their morning routine. The primary audience includes busy professionals who need to look polished for work, college students managing limited wardrobes, and young adults seeking to maximize their existing clothing collection without purchasing new items.

Secondary users include fashion enthusiasts who enjoy experimenting with different style combinations and parents looking to reduce morning chaos through advance planning. These users value simplicity, efficiency, and privacy. They want a straightforward tool that helps them visualize outfit combinations quickly without the complexity of social features, shopping integrations, or data sharing.

The application appeals particularly to desktop computer users who prefer focused planning sessions over mobile multitasking, and who appreciate having a completely offline tool that respects their privacy while providing practical wardrobe organization benefits.

MOCKUP

Image 1:



FUNCTIONAL REQUIREMENTS

Must-criteria

FR-M1: The app must be accessible as a web application.
FR-M2: The app must use images of basic clothing items provided by the database.
FR-M3: The user must be able to create an account and login to the application.
FR-M4: The app must start with a default screen – containing question marks "?" instead of clothing items in the given boxes.
FR-M5: The user must be able to see a screen containing boxes displaying the clothing items as images in said boxes – including tops, bottoms and footwear, as seen in the Image 1.
FR-M6: The user must be able to move from one item to another using arrows, displayed to the left and right of the boxes containing the images of the clothing items.
FR-M7: The user must be able to save an outfit, which they are able to retrieve from the list of saved outfits displayed on the web application (it is saved in form of a default name e.g. Outfit 1).
FR-M8: The user must be able to edit saved outfits and save those made edits.
FR-M9: The user must be able to delete saved outfits.

Should-criteria

FR-S1: The user should be able to filter the clothing using a "hamburger menu" which consists of the following sorting options: top, bottom and footwear.
FR-S2: The user should be able to use a "MisMatch" button, to create an outfit with randomly chosen clothing items from our list.
FR-S3: The user should be able to rename the already saved outfits.
FR-S4: The app must include a camera upload feature, so the user can include their own clothing items.

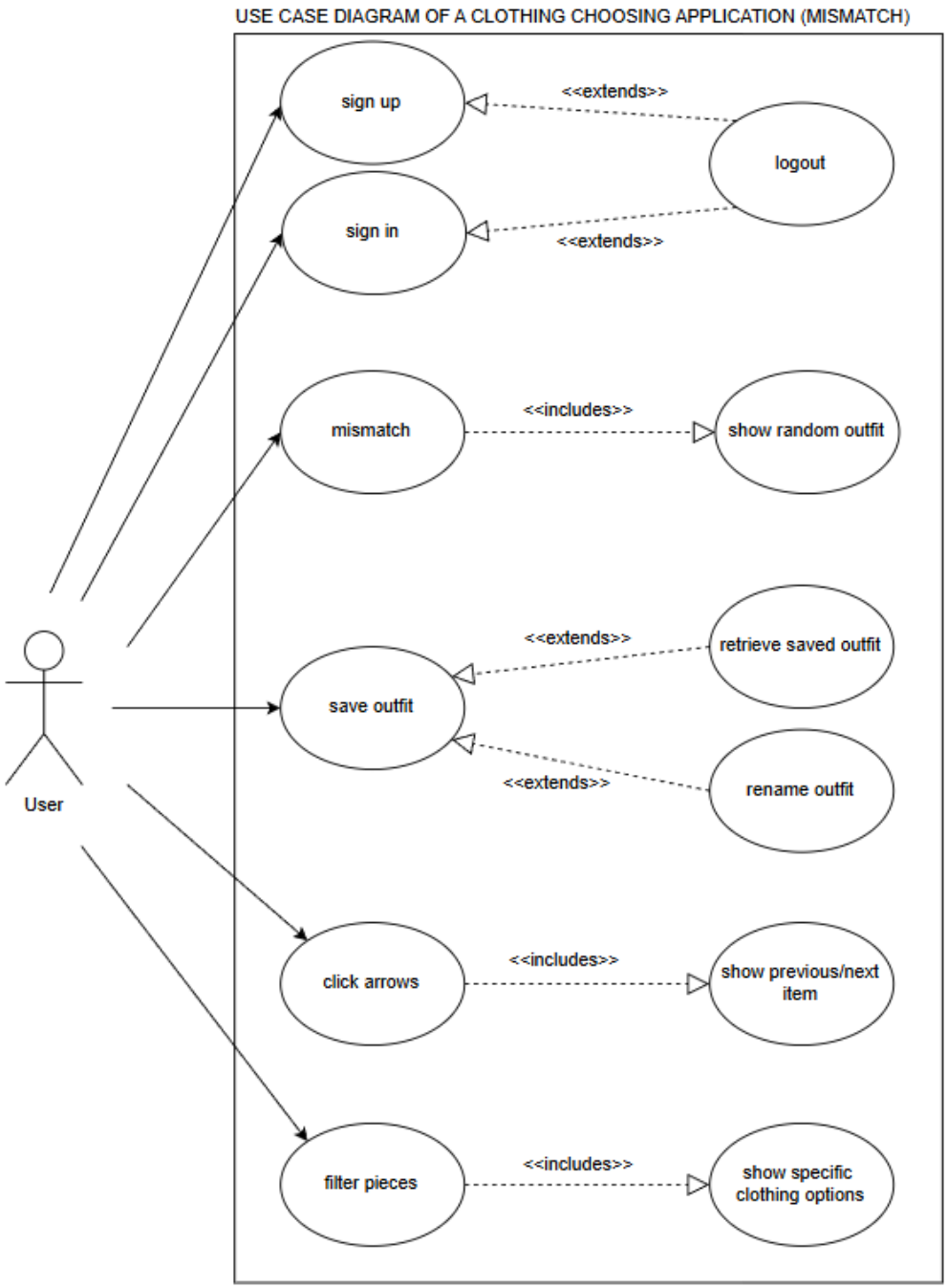
Could-criteria

FR-C1: The user could be able to filter the saved outfits (e.g. newest first, a-z)
FR-C2: The user could be able to filter the clothing items by colour.
FR-C3: The app could include a drop-down menu for user settings.
FR-C4: The user could change their username and password using the edit button shown in the drop-down menu.
FR-C5: The user could be able to choose their avatar (shown as the head of the character) from the given options.

Won't-criteria

FR-W1: The user will not be able to rate outfits.
FR-W2: The user will not be able to like/dislike items.
FR-W3: The app will not include an AI assistant.
FR-W4: The app will not include one-piece items such as dresses, overalls, onesies etc.

USE CASE DIAGRAM



USE CASE DESCRIPTIONS

Use Case	Sign Up
Actor	User
Description	A new user creates an account in the system by providing necessary registration information
Stimulus	User wants to access the application features and needs to create an account
Response	System validates the registration data, creates a new user account, and confirms successful registration
Comment	This is typically a one-time action per user. Should include validation for email/username uniqueness.

Use Case	Login
Actor	User
Description	An existing User logs into the system using their credentials
Stimulus	User wants to access their personal wardrobe and outfit management features
Response	System authenticates the credentials and grants access to the user's account
Comment	Should include error handling for incorrect credentials

Use Case	Mismatch
Actor	User
Description	User requests the system to automatically/ randomly create an outfit combination from their wardrobe items.
Stimulus	User wants outfit inspiration or suggestions based on the available clothing items in the application.
Response	System selects compatible clothing items and displays the generated outfit to the user.
Comment	The generated outfit is automatically shown (includes "show outfit")

Use Case	Show Outfit
Actor	User
Description	System displays a complete outfit
Stimulus	User has generated or selected an outfit and wants to view it
Response	System delivers a visual representation of the entire outfit
Comment	This is included in “mismatch” (it is the display component of outfit generation)

Use Case	Save Outfit
Actor	User
Description	User stores a created or generated outfit for future reference
Stimulus	User wants to keep an outfit combination they like for later use
Response	System prompts for outfit name, saves the outfit to the user's collection, and confirms successful save
Comment	Includes naming the outfit. Saved outfits should be retrievable later

Use Case	Arrows to choose Items
Actor	User
Description	User chooses to bypass the current clothing item when browsing through wardrobe items, but can also refer to the previous one
Stimulus	User is browsing items (possibly during outfit generation) and wants to move past/move back to the current item without selecting it
Response	System moves to the next/previous item in the sequence and displays it
Comment	Useful during outfit creation or wardrobe browsing. Includes showing the next/previous item automatically

NON-FUNCTIONAL REQUIREMENTS

NFR-1: The app shall be coded in Python and is displayed in English.

NFR-2: The app shall be implemented as a web-based solution and work on modern web browsers (Firefox, Chrome, Safari, Microsoft Edge)

NFR-3: The app shall use MongoDB as a database for storing user data.

NFR-4: The app shall respond in a timely manner, the limit being a maximum of 5 seconds.

NFR-5: User sessions shall remain active until logout or browser close and shall be managed securely using Flask sessions.

NFR-6: User passwords must be stored securely using appropriate hashing techniques such as Argon2id.

NFR-7: Core application functions shall be covered by automated unit tests to verify correctness.