



Software Requirements Specification (SRS)

Project Name: MyCloset

Irit Inbar 313332025

Mona Badarni 213971955

1. Introduction

1.1 Purpose

Describe the purpose of the application. Explain why it is being developed and the primary goal it aims to achieve.

MyCloset is developed to help with keeping track of yours and any closet you'd like. By taking photos of items and inserting its traits and qualities for fast filter and search, adding any tags the user would want to filter by. The app also allows you to create outfits, save them and share with approved users they will choose. You can also suggest or have your friends suggest different items to existing outfits or new ones.

1.2 Intended Audience

Specify who the intended audience is. Include categories like end users, developers, or project managers, and describe why they would use the system.

As part of the study for the application, a survey was conducted to examine the needs and preferences of potential users regarding the management of their wardrobe.

The survey was distributed to our social groups- friends and families, and we received responses from diverse users—students, young women, men aged 20–35, and individuals involved in fashion.

Many respondents indicated a clear need for better organization and structure in their closet. Users expressed a desire for a feature that allows them to scan, photograph, and document information about their clothing items in one place.

In addition, users mentioned they would appreciate receiving suggestions for outfit combinations, information about what matches well, and reminders about items they have not worn for a long time.

Users emphasized the following challenges:

- Difficulty remembering what items they have in their wardrobe.

- Difficulty selecting outfits quickly in the morning.
- Wasted money from purchasing duplicate items.
- A need for a system that enables creating outfits, saving ideas, and adapting clothing to seasons.

Participants were asked which features they would like to see in a wardrobe-management app. The findings indicate a clear demand for several key capabilities:

- Photo upload
- Filtering by color / type / season.
- Outfit creation
- Usage statistics
- Multiple closets.
- Sharing with friends
- Mix and match
- Favorites

The app can be useful for young women especially ones interested in fashion, big families with the need to organize their children's closets and anyone who wants to decrease the time spent dealing with clothes.

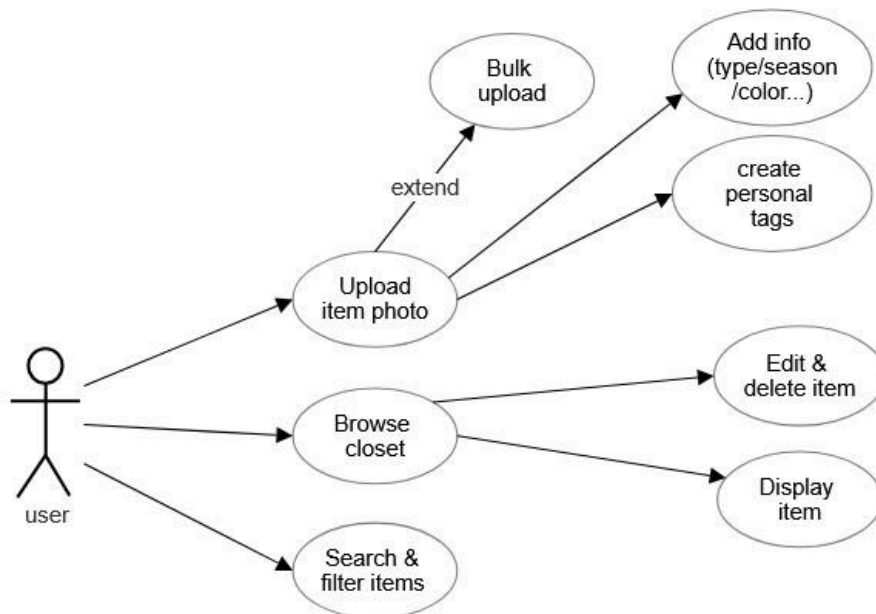
1.3 Intended Use

Describe how the application will be used. Include typical scenarios or use cases that demonstrate its main functionality.

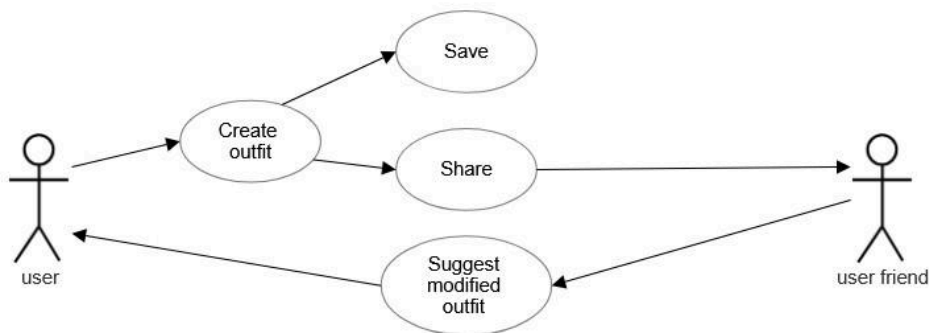
The main functionality is demonstrated in these use cases diagrams.



MyCloset Item Management Use Case



MyCloset Create & Share Outfits Use Case



Use case 1: creating outfit

Purpose: the user want to create outfit

actors: logged user

Prerequisites: there are existing items in the closet

Process steps:

1. User press the button “create outfit”
2. The user is presented with types of items to choose from. For example: pants/ dress/skirts
3. The user scrolls through photos of items, can pin specific items and continue scrolling to match another. Pressing the save button when done.

Success result: outfit added to your outfits collections

Alt results: not pressing save and exiting, going back to the previous page, outfit isn't created.

1.4 Product Scope

Outline the scope of the application. What are its primary features, and what boundaries or limitations does it have?

The primary features are as detailed in the use cases diagrams.

There won't be reference to online stores for the purpose of advertising. There would be a limit for storage that will be decided at a later date.

1.5 Definitions and Acronyms

List and define any technical terms, abbreviations, or acronyms that will appear in the Document.

User & friend describes regular users, at the viewpoint of the original user and a friend = a user that has permission to the original user closet, item and to suggest outfits.

2. Overall Description

2.1 User Needs

Identify the needs of the target users. What problems does the application solve for them?

The app solves redundant shopping, avoids mess created by searching your closet and decreases time picking outfits for you and anyone you are in charge of. You can share and get feedback from your friends.

2.2 Assumptions and Dependencies

List any assumptions made about the system's environment or users. Include any dependencies on external systems, services, or technologies.

Users most likely have a smart phone with a camera and connection to the internet. Assuming the users use the phone on a daily basis and know how to operate a smart phone at a basic level..

3. System Features and Requirements

3.1 Functional Requirements

List the main functions of the application. For each function, describe what the system should do. Examples include creating records, processing data, and providing Notifications.

Every option will be accessible by side menu.

For managing the closet: the system will display items browsing through the closet. choosing any item by clicking on a photo. The item will display the photo and every info added to it. You can click edit to add or remove any info and also can create a tag to the item.

To upload an item click on the side menu, take a photo of the item, next to it choose the correct type, color, season and other attributes. add any tags by typing it and it will create it globally and be suggested for other items while you edit.

The app can display statistics by tags, favorites and recent items and outfits created.

To create an outfit click on the side menu, choose the type of items you want to choose from by ticking boxes, to avoid displaying dresses while building an outfit with pants. Any item you want to keep, pin it and continue scrolling through the other types until you are done with matching items. The outfit will be saved and present with the option to share it.

For friends that view the shared outfit, will be displayed the option to suggest different items. After clicking, they will encounter the same screen after creating an outfit from the origin closet, and by clicking suggest the user will receive notification. The user can save the suggestion as the existing or additional outfit.

3.2 External Interface Requirements

- **User Interface:**

Describe the user interface requirements. What platforms will it support (e.g., web, mobile)?

The app will require a smart phone with android system.

- **Third-party Integrations:**

List any required integrations, such as APIs or external services.

Google drive REST API

FireBase Android SDK

- **Cloud Storage:**

Specify any data storage or synchronization requirements.

Data storage will be firebase combined with google drive.

3.3 System Features

Provide a brief description of the main features. Examples include dashboards, reports, or specific tools.

The main page will be the home page. Displaying options for search, creating outfit, browsing closet, saved outfit, profile. In your profile you can see friends, statistics, and settings.

3.4 Nonfunctional Requirements

- **Performance:**

Describe expected performance metrics, such as response times.

At basic level the goal will be a common response time for a simple app built in android studio after research (to accommodate for our beginner knowledge).

Navigation to a new screen within 300-600 ms.

Basic local operation completed most at 150 ms, Database queries at 300 ms.

Lightweight api like firebase call should return results in 500-800 ms, google drive api calls will likely 700 ms - 2 seconds.

- **Scalability:**

Indicate how the system should handle increased loads or users.

To handle increased loads or users we will use offline caching when possible with consideration for firebase cloud that automatically scales with users and traffic, and google drive api that depends on the user's network and google servers.

- **Security:**

Outline security measures, such as encryption or authentication.

The app will use security measures provided by android studio automatically, also use some that are available for authentication: Firebase authentication, google sign in. and more will be updated during development.

- **Availability:**

Define uptime or reliability expectations.

App's uptime depends and relies on firebase and google drive with uptime of at least 99%, for performance and availability we will incorporate offline caching.

4. Specific Needs of End Users

4.1 Description of User Groups

- Provide a detailed description of the user groups (primary and secondary), including unique characteristics and roles.

Users can grant access to two types of permissions, one would be a friend who can view your shared outfits and closet with suggestions, the other defined as a family that acts as co owners of their closets with your chosen permissions.

4.2 Unique User Requirements

- Detail unique requirements for users, such as special access, customized views, or specific functionalities.

Common users will use all the functions the app offers.

Premium users can also sync if they log in on different phones and added storage space.

5. Change Management

5.1 Change Tracking

- Describe how changes to the document will be managed, including version control, responsible parties, and the approval process for updates.

For every week the team, as mentioned in the document title, will discuss any changes made and update the document accordingly and save as a different file, adding it to the Trello board after approval of the whole team..

5.2 Version Control

- Each change in the document should be recorded in a version history, including the date, responsible party, and a brief description of the change.

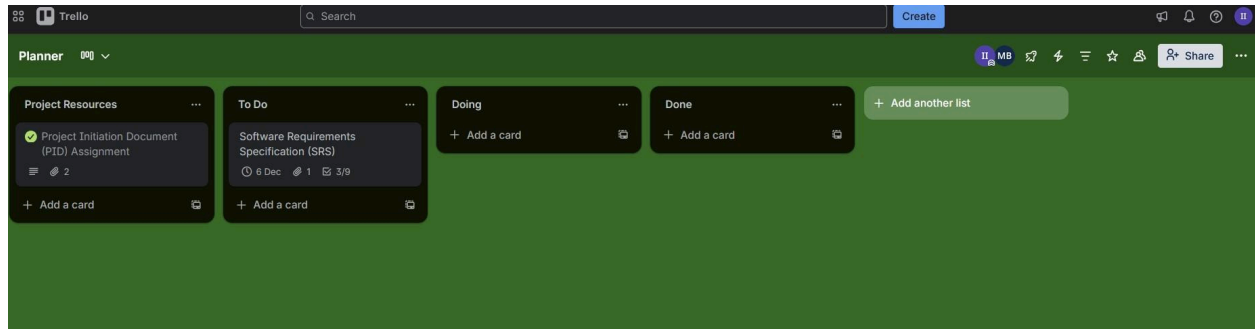
This section will be for detailing changes made, their description, reasons for change and the date of change.

6. Appendices

- Use Case Diagrams: Include diagrams that represent the core processes of your system, showing the relationships between actors and use cases

Use cases diagram detailed in 1.3.

- Task Management Tool: Insert a screenshot of your main project board (e.g., Trello, Jira, Monday). Ensure the view displays the current status of all tasks, including those completed and those currently in progress.



Using trello, but not updated with development progress yet.

- Additional Information: Add any relevant references, tables, or supplementary materials that support understanding the system.

No additional information currently