Bilkent University
Department of Computer Engineering

Senior Design Project
T2327
Capsule

Analysis and Requirement Report

22003186, Ali Emir Güzey, emir.guzey@ug.bilkent.edu.tr;
22003229, Alp Afyonluoğlu, alp.afyonluoglu@ug.bilkent.edu.tr;
22003158, Ceren Akyar, ceren.akyar@ug.bilkent.edu.tr;
22003530, Deniz Mert Dilaverler, mert.dilaverler@ug.bilkent.edu.tr;
22002379, Mehmet Kağan İlbak, kagan.ilbak@ug.bilkent.edu.tr

Supervisor: Asst. Prof. Hamdi Dibeklioğlu
Course Instructors: Dr. Atakan Erdem & Mert Bıçakçı

08.12.2023

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

Analysis and Requirement Report

Project Short-Name: Capsule

1 Introduction

Our clothing choices reflect our personalities and often don't get the credit they deserve. People, ranging from high school students to middle-aged adults, are increasingly paying attention to their clothing choices. This comes with a set of challenges, one of which is the overwhelmingness of tracking daily outfits and finding clothes to dress differently every day. As a result, people are prone to shopping beyond their needs despite having perfectly good outfit options in their wardrobes. Research shows that the average U.S. citizen annually spends 1434$ on clothing, which has its own economic and environmental drawbacks [1]. Capsule aims to solve this problem.

We embarked on this journey with the idea of providing people with a “capsule wardrobe,” which is described as “a collection of clothing composed of thoughtfully curated, easily interchangeable items designed to maximize the number of outfits that [one] can create.” [2] Our objective is to enable users to optimize their wardrobes by curating stylish outfits without the burden of possessing an unnecessary amount of clothing.

In this report, we cover our competitors in the market and Capsule’s functional and non-functional requirements. This report will also visualize the project structure through many different diagrams and UI mockups. Apart from the design choices, this report will also cover the risks and possible alternatives, the plan of implementation, and all around our plans on how we are going to work during the implementation stage.

2 Current System

Table 1: Competitive market analysis for Capsule

App Name	Capsule (Our Project)	Whering	Pronti	Pureple
Outfit Recommendation	Yes	Yes	Yes	No
Manual Outfit Builder	Yes	Yes	Yes	Yes
UI Quality	Good	Good	Cluttered and Unintuitive	Bad
Wardrobe Analytics	Yes	Yes	No	No
Shopping Recommendation	Yes (with Turkish retailers)	Yes, but inaccurate and no Turkish retailers	Yes, but no Turkish retailers	No
Outfit Collaboration for Events	Yes	No	No	No
Business Plan	Affiliate Marketing, Sponsored Content	No ads or subscriptions, likely affiliate marketing	No ads or subscriptions, likely affiliate marketing	₺28/month on Android, ₺99/month on iOS, Unsubscribed use is nearly impossible due to overwhelming ads

While the market of smart wardrobes isn't new we believe that Capsule can do things better than the competition, especially for the Turkish market. We have picked 3 of the most prevalent competitors in the market. Pureple was the worst out of all of the smart wardrobe solutions we have seen. They went with a subscription model, which isn't worth the price, and without a subscription, it is impossible to use the app as every click causes a 5 to 30-second advertisement to be run.

Another competitor of ours is Pronti. Unlike Pureple, they have redeemable qualities, and it is usable. However, the UI and UX of Pronti aren't desirable. It is hard to navigate and all around ugly.

Whering is the main concern of ours when it comes to market competition. They have a clean UI with a good feature set; however, with the event feature and the better implementation of the core features, we believe that we can steal a significant amount of market share from them. Despite its qualities, Whering doesn't have much presence or support for the Turkish market, which gives us the opportunity to start from the Turkish market and grow our customer base.

3 Proposed System

3.1 Overview

Capsule is a mobile wardrobe solution powered by image processing and machine learning. With Capsule, users can track when and what they wear and get recommended outfits according to user preference and clothing item matching. By tracking users' clothing preferences, not only will we be able to recommend outfits but also recommend them to sell certain items since they haven't been worn in a while. The user will also be using our platform to log and plan outfits. Our knowledge about the user's outfit preference will also enable us to direct clothing retailers to their preferred demographics via advertisements and/or affiliate marketing.

In addition to the core feature, Capsule will also provide a platform for users to discuss and set dress codes for events they are attending with their friends. For example, if you are going out to a fast food restaurant or going out for drinks, you and your friends can decide on an outfit to wear on the platform. The event feature will be more useful than a regular instant messaging app like "WhatsApp" as the user will have their outfits and clothes integrated with the messaging section and can seamlessly send outfits/clothes and get feedback on them.

Our vision is to create a mobile application to track and advise users on their wardrobe usage. In order to achieve our vision, a lot of image processing and machine learning is needed as we need to segment the outfit from a given picture, match it with an item from your wardrobe, label it with appropriate properties, and use the combination of multiple properties to generate an optimal outfit.

3.2 Functional Requirements

3.2.1 Events

- Users can create an event with date, time, location (optional), and dress code selection (optional). Event title and description. Max participant 8 (32 for paid users)
- The event owner can generate an invite link to invite more members.
- People can click and join through the link
- Event owners can open a dress-code selection poll or select the dress code themselves. The dress codes are as follows:
    - formal
    - business
    - night out/date
    - streetwear
    - sports
- Participants can send pictures of clothes and outfits from their wardrobes
- Participants can also generate outfits in accordance with the dress code
- Participants can freely chat with each other in the event group
- Groups are archived after the event finished (at the end of day)
- Participants are notified when the event is coming up
- Participants can see their events and proposed outfits in a week-view
- Participants can set an outfit for an event, visible by all event members

3.2.2 Registering Clothes

- Users can take pictures from the app of a single clothing item or a mirror selfie showing multiple clothing items to be registered
- Users can upload full-body pictures or a picture of a single clothing item from their gallery
- Users can enter a web-url of an outfit listing from a supported online retailer to add the item to register the clothing.

3.2.3 Building Outfits Manually

- Users can create an outfit from the clothes in their wardrobe. These outfits are stored and can be used in event chats and outfit logging
- Users can view and edit their outfits' properties such as name, and tags (formality of clothing, season)
- Users select clothing items from each clothing type's list

3.2.4 Login

- User can log in to the platform through their email and password if they signed up with them
- User can log in with their Google account if they signed up with it
- User can log in with their Apple acount if they signed up with it

3.2.5 Sign-up

- Usesr can sign up with their Google account, Apple account or email and password on the platform
- If the user signed up with email, they verify their account with an authentication code sent to their email address.
- Users enter full name, date of birth, and username.
- Users can upload a profile picture
- Users select a style preference from the following: Masculine, Neutral, Feminine

3.2.6 Reset Password

- If the user signed up with password and email, they can reset password whether they are signed in or not
- Users can authorize password change from a link they receive through email
- Users enter a new password

3.2.7 Generating Outfits with Machine Learnings

- User requests an outfit to be generated by ML model
- User views the generated outfit
- User swipes right to like the outfit which saves the outfit
- User swipes left to reject the generated outfit and generate another one.
- User swipes down to select the outfit for the day, which saves the outfit if it doesn't exist
- User occasionally sees an outfit built from sponsored clothes

3.2.8 Selecting Outfits For a Day

- User selects “Today” or a day from the calendar view
- User selects an outfit by
    - selecting a saved outfit from the built outfits in the wardrobe
    - building an outfit for the day
    - taking a mirror selfie to match worn outfits to the ones in the system
- User adds the outfit to the selected day

3.2.9 Viewing Wardrobe

- User can select between outfits and clothing views
- User filters clothes by color, and type (top, overwear, bottom, full-body, shoes)
- User can addclothing or outfit to favorites
- User can view favorites
- User views outfit/clothing properties
    - ML-recommended
        - Colors
        - Tags
        - Seasons
        - Texture
    - Usage counts
    - Last worn date
    - favorited or not
    - create an outfit with this
- User deletes outfit/clothing
    - If clothing is in an outfit, the user is prompted with “You have outfits with this item do you also want to delete them?” This prompt will only be asked once. This selection can be changed from the settings page.
        - Yes: the clothing is deleted, and the outfit has an empty slot at the palace of the clothing
        - no: The clothing is not deleted
- User adds the outfit/clothing to wardrobe

3.2.10 View Analytics

- User opens weekly and monthly wardrobe reports at the end of the week/month.
- User vies the most/least worn clothes for the time period
- User views the most worn colors during the time period
- User views the pie chart of outfit categories in wardrobe
- User views the new outfits / new clothes added to wardrobe (3 examples most and count of items (weekly))
- User views wardrobe utilization percentage (weekly/monthly)

3.2.11 View Advertised Outfits

1. Show advertised pieces once in a certain amount for outfit generation.
2. User triggers shopping recommendation from a single clothing piece
    a. Show the user recommended shopping pieces

3.2.12 User Profile Creation and Management

- User creates and manages their profiles, including adding personal information, profile pictures, and wardrobe details.

3.2.13. Event Discussion and Dress Code Setting

- Users within an event are able to engage in discussions related to the event.
- Users propose dress codes for the event and invite friends to participate in setting dress codes.
- Event participants vote on proposed dress codes.
- Participants can propose clothes and outfits for the event, along with discussions.

3.3 Non-functional Requirements

In this subsection, we outline the critical performance, security, and quality criteria that Capsule must adhere to. These non-functional requirements encompass aspects such as data privacy, system responsiveness, scalability, and usability.

3.3.1. Usability

The application should be intuitive to use. It should not have complex user interface elements and paths that could confuse the user. The user should be able to learn and navigate the basics of the app in under 5 minutes. Furthermore, some features of the application should be usable without an internet connection. Features such as viewing wardrobe and previous outfits should be available offline.

3.3.2. Reliability

The application should have a reliable infrastructure and should not crash. In the worst case, the backend should not be down for more than 10 minutes. The application should also produce reliable outfits. Test accuracy results of the machine learning models should be higher than 90%.

3.3.3. Performance

The user interface of the application should be fast in order to maximize the user experience. Rendering each component should take less than 100ms. The machine learning models should produce decent results in under 1 second. Furthermore, backend resource consumption should be less than 80% at all times. This would be a measure for usage spikes, which could halt the backend.

3.3.4. Supportability

The application should be supportable, such that it could receive further development in the future. It should be supportable for at least a year due to potential contracts with brands. The application should not have predefined instructions that could hinder its development in the future.

3.3.5. Scalability

The application should be scalable to more than 100,000 users without a major financial burden. It should be easy to expand the capacity to over 1,000,000 users in order to serve the incoming users.

3.4 Pseudo Requirements

- Python 3.11 will be used for the backend with Django version 4.1 and for ML with Torch 2.1.1.
- Github will be used for both the repository hosting tool and issue tracker.
- The relational data will be stored in Postgres DB.
- Media files will initially be stored locally in the web container and then be migrated to an Azure Blob instance.
- The frontend will be implemented on React Native as a cross-platform mobile app.
- Git will be used for version control.
- An extensive terms of service contract must be approved by every customer on signup.
- The Pre-commit library will be used on the backend to ensure the changes in commits are well formatted.
- The application language will be English, with plans on porting it to Turkish.
- All of the hosting will be done on Azure cloud services.
- Commits directly on the main branch are prohibited and disabled on GitHub.
- PRs on the main branch must be reviewed and approved by another team member. Otherwise, it won't be allowed to be merged.
- PRs must be merged through rebasing and fast-forwarding.
- ML models will be connected to the main Django server through REST APIs.
- The segmentation engine will be served to the main backend using FastAPI

3.5 System Models

3.5.1 Scenarios

3.5.1.1 Manual Outfit Creation

Purpose: The scenario describes the use of a manual outfit creation feature by a default user
Individual: An authenticated Capsule user (premium or default)
Equipment: Any mobile phone with a supported system version
Scenario:

1. The user presses the Add a new outfit button
2. The user then selects the option to create a manual outfit
3. The user selects the outfit size (one piece and shoes, top-bottom and shoes, or these options with an additional coat)
4. Then, according to the selected outfit size, the relevant clothing items are displayed
5. User selects their preferred items
6. The user selects to continue if satisfied with the answers
7. The user views the displayed outfit
8. If preferred, the user enters an outfit name and some tags about the outfit
9. The user saves the outfit

3.5.1.2 AI-Generated Outfit Creation

Purpose: The scenario describes the use of an ai-generated outfit creation feature by a default user
Individual: An authenticated Capsule user (premium or default)
Equipment: Any mobile phone with a supported system version
Scenario:

1. The user presses the add a new outfit button
2. The user then selects the option to create an AI-generated outfit
3. User types the event they are going to or the kind of outfit they want to a chat area
4. Then, the system creates some relevant dress codes
5. Outfits relevant to the dress code are suggested to the user one by one
6. The User swipes left if they do not like the outfit
7. If they do not want to see an outfit again, they can state that they are not interested
8. The user swipes right if they like the outfit and saves the outfit
9. After that, the user can enter an outfit name or edit some tags about the outfit
10. The user saves the outfit

3.5.1.3 Event Creation

Purpose: The scenario describes the use of event creation by a default user
Individual: An authenticated Capsule user (premium or default)
Equipment: Any mobile phone with a supported system version
Scenario:

1. The user presses to create a new event
2. The user enters the date, time, and name of the event
3. The user decides on whether to select a dress code or decide on it via a poll
4. The user generates an invitation link and creates the event
5. The user shares the link with other users

3.5.1.4 Clothing Item Recommendation

Purpose: Scenario describes the use of clothing item recommendation
Individual: An authenticated Capsule user (premium or default)
Equipment: Any mobile phone with a supported system version
Scenario:

1. The user goes to the item recommendation screen
2. The user selects one or more outfit pieces
3. Then, the system suggests to the user some more pieces that will go well with the selected pieces
4. The user can swipe left until they see something they like
5. The user can click on the items to go to the seller's page and buy the item easily
6. Users can add the items to their wardrobe easily if they like an item and decide to buy

3.5.2 Use-Case Model

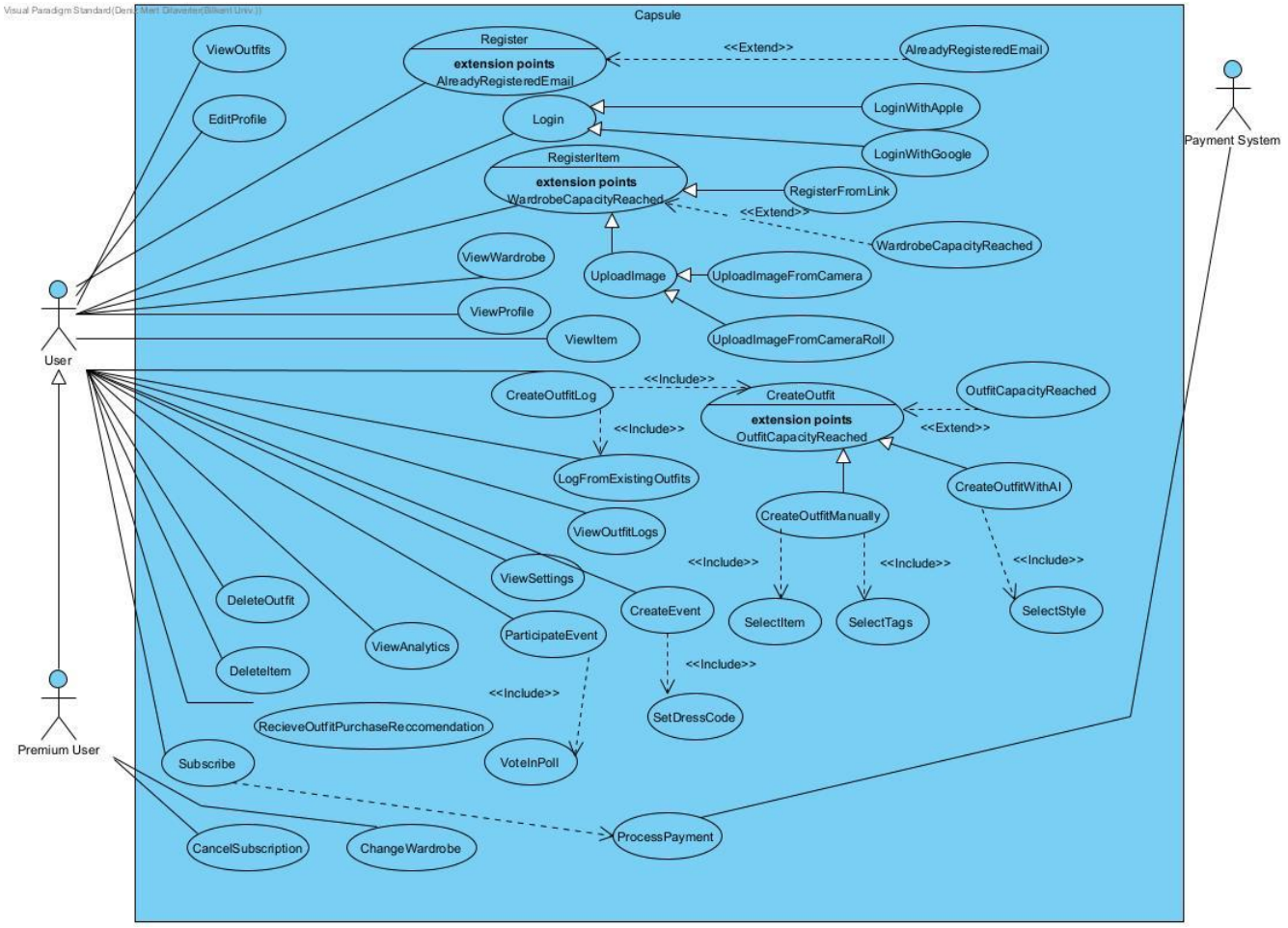

Fig. 1: Use-Case Model

3.5.2.1 Login

Use Case Name: Login
Participating Actor: User
Entry Condition: User opens the app and comes up to this screen if did not open the app in a month
Exit Condition:

- User logins successfully
- An error occurs

Flow of Events:

1. User can log in to the platform through their email and password if they signed up with them
2. User can log in with their Google account if they signed up with it
3. User can log in with their Apple account if they signed up with it

3.5.2.2 Register

Use Case Name: Register
Participating Actor: User
Entry Condition: User opens the app, and comes up to this screen if pressed on “register”.
Exit Condition:

- User registers successfully
- An error occurs (For instance, user already has an account)

Flow of Events:

1. User enters a username, password, email address, birth date, and profile picture
2. User selects their clothing preferences (feminine, gender neutral, masculine)
3. User selects body type
4. User selects clothing sizes
5. User creates account

3.5.2.3 EditProfile

Use Case Name: Edit Profile
Participating Actor: User
Entry Condition: User clicks on the profile section at the bottom navigation bar and clicks on edit profile
Exit Condition:

- User edits successfully
- An error occurs (For instance, already taken username)

Flow of Events:

1. User edits their username, password, email address, birth date, or profile picture
2. User edits their clothing preferences (feminine, gender neutral, masculine)
3. User edits body type
4. User edits clothing sizes
5. User clicks on save to save the new information

3.5.2.4 ViewWardrobe

Use Case Name: ViewWardrobe
Participating Actor: User
Entry Condition: User clicks on the wardrobe section at the bottom navigation bar and clicks on items
Exit Condition:

- User clicks on another section

Flow of Events:

1. User views their clothing items
2. User can filter items according to their categories, subcategories, tags, and colors
3. User can view their favorite items

3.5.2.5 ViewOutfit

Use Case Name: ViewOutfit
Participating Actor: User
Entry Condition: User clicks on the wardrobe section at the bottom navigation bar and click on outfits
Exit Condition:

- User clicks on another section

Flow of Events:

1. User views their outfits
2. User can filter outfits according to their categories and colors
3. User can view their favorite outfits

3.5.2.6 ViewProfile

Use Case Name: ViewProfile
Participating Actor: User
Entry Condition: User clicks on the profile section at the bottom navigation bar
Exit Condition:

- User clicks on another section

Flow of Events:

1. User views their username, password, email address, birth date, profile picture, clothing preferences, body type, clothing sizes

3.5.2.7 ViewItem

Use Case Name: ViewItem
Participating Actor: User
Entry Condition: User clicks on the wardrobe section at the bottom navigation bar, click on items, and click on a specific item
Exit Condition:

- User clicks on to the “go back” icon

Flow of Events:

1. User can see an item’s image, category, subcategory, color, tag
2. User can add the item to the favorites, or remove from favorites
3. User can edit item properties

3.5.2.8 ViewSettings

Use Case Name: ViewSettings
Participating Actor: User
Entry Condition: User clicks on the profile section at the bottom navigation bar and click on the settings icon
Exit Condition:

- User clicks on another section
- User saves the settings

Flow of Events:

1. User can edit notification settings
2. User can view security and privacy

3. User can view contact information and FAQ
4. User can view terms and conditions
5. User can log out
6. User can delete account
7. User can leave a review to the app

3.5.2.9 CreateEvent

Use Case Name: CreateEvent
Participating Actor: User
Entry Condition: User clicks on the event section at the bottom navigation bar and click on new event
Exit Condition:

- User clicks on another section
- User clicks on add event and creates the event

Flow of Events:

1. User enters event name, event date, event time
2. User can select a dress code or can skip this part for later
3. User generates an invitation link and shares the link with friends

3.5.2.10 SetDressCode

Use Case Name: SetDressCode
Participating Actor: User
Entry Condition: User clicks on new event
Exit Condition:

- User clicks on another section
- User creates the event successfully

Flow of Events:

1. User selects a dress code from a list of dress code options

3.5.2.11 ParticipateEvent

Use Case Name: ParticipateEvent
Participating Actor: User
Entry Condition: User clicks on the link shared with them
Exit Condition:

- User clicks on another section
- User joins to the event successfully

Flow of Events:

1. User clicks on join event
2. User can chat with friends
3. User can share outfits with friends
4. User can vote in a poll if one is opened

3.5.2.12 VoteInPoll

Use Case Name: VoteInPoll
Participating Actor: User
Entry Condition: A poll is opened on an event chat
Exit Condition:

- User clicks on another section
- User votes in the poll successfully

Flow of Events:

1. User can add a new dress code option to the poll
2. User can vote for one or more of the added dress code options

3.5.2.13 DeleteItem

Use Case Name: DeleteItem
Participating Actor: User
Entry Condition: User clicks on the wardrobe section at the bottom navigation bar, clicks on items, and clicks on a specific item
Exit Condition:

- User clicks on to the “go back” icon
- User deletes item successfully

Flow of Events:

1. User clicks on the delete item
2. User selects whether the outfits containing the item will also be deleted or keep remaining

3.5.2.14 DeleteOutfit

Use Case Name: DeleteOutfit
Participating Actor: User
Entry Condition: User clicks on the wardrobe section at the bottom navigation bar, click on outfits, and click on a specific outfit
Exit Condition:

- User clicks on to the “go back” icon
- User deletes outfit successfully

Flow of Events:

1. User clicks on the delete outfit

3.5.2.15 ViewAnalytics

Use Case Name: ViewAnalytics
Participating Actor: User
Entry Condition: User clicks on the wardrobe section at the bottom navigation bar, click on analysis
Exit Condition:

- User clicks on to the “go back” icon

Flow of Events:

1. User can view weekly and monthly wardrobe reports at the end of the week/month
2. User can view most/least worn clothes
3. User can view the most worn colors
4. User can view the pie chart of outfit categories in wardrobe
5. User can view new outfits / new clothes added to wardrobe
6. User can view wardrobe utilization percentage

3.5.2.16 ViewAnalytics

Use Case Name: ViewAnalytics
Participating Actor: User
Entry Condition: User clicks on the wardrobe section at the bottom navigation bar, click on analysis
Exit Condition:

- User clicks on the “go back” icon

Flow of Events:

1. User can view weekly and monthly wardrobe reports at the end of the week/month
2. User can view most/least worn clothes
3. User can view the most worn colors
4. User can view the pie chart of outfit categories in wardrobe
5. User can view new outfits / new clothes added to wardrobe
6. User can view wardrobe utilization percentage

3.5.2.17 RegisterItem

Use Case Name: RegisterItem
Participating Actor: User
Entry Condition: User clicks on the wardrobe section at the bottom navigation bar, click on add new item button
Exit Condition:

- User clicks on the “go back” icon
- User adds an item successfully
- Wardrobe capacity is reached

Flow of Events:

1. User selects one of the options from uploading from a link, from photo gallery, or to take a photo
2. User edits the category and subcategory of the item
3. User edits the color of the item
4. User edits the tags of the item
5. User clicks on save

3.5.2.18 RegisterFromLink

Use Case Name: RegisterFromLink
Participating Actor: User
Entry Condition: User clicks on the register from link section
Exit Condition:

- User clicks on to the “go back” icon
- User adds an item successfully

Flow of Events:

1. Users can enter a web URL of an outfit listing from a supported online retailer to add the item to register the clothing.

3.5.2.19 UploadImageFromCamera

Use Case Name: UploadImageFromCamera
Participating Actor: User
Entry Condition: User clicks on take photo
Exit Condition:

- User clicks on the “go back” icon
- User adds an item successfully

Flow of Events:

1. Users can take pictures from the app of a single clothing item or a mirror selfie showing multiple clothing items to be registered

3.5.2.20 UploadImageFromCameraRoll

Use Case Name: UploadImageFromCameraRoll
Participating Actor: User
Entry Condition: User clicks on register from camera roll
Exit Condition:

- User clicks on to the “go back” icon
- User adds an item successfully

Flow of Events:

1. Users can upload full-body pictures or a picture of a single clothing item from their gallery

3.5.2.21 CreateOutfitManually

Use Case Name: CreateOutfitManually
Participating Actor: User
Entry Condition: User clicks on create outfit manually
Exit Condition:

- User clicks on to the “go back” icon
- User creates outfit successfully
- Outfit capacity reached

Flow of Events:

1. Users can create an outfit from the clothes in their wardobe. These outfits are stored and can be used in event chats and outfit logging
2. Users can view and edit their outfits’ properties such as: Name, tags (formality of clothing, season)
3. Users select clothing items from each clothing type’s list

3.5.2.22 CreateOutfitWithAI

Use Case Name: CreateOutfitWithAI
Participating Actor: User
Entry Condition: User clicks on create outfit with AI
Exit Condition:

- User clicks on the “go back” icon
- User creates outfit successfully
- Outfit capacity reached

Flow of Events:

1. User requests an outfit to be generated by ML model
2. User views the generated outfit
3. User swipes right to like the outfit, which saves the outfit
4. User swipes left to reject the generated outfit and generate another one
5. User swipe down to select the outfit for the day, which saves the outfit if it doesn’t exist
6. User occasionally sees an outfit built from sponsored clothes

3.5.2.23 CreateOutfitLog

Use Case Name: CreateOutfitLog
Participating Actor: User
Entry Condition: User clicks on outfit log
Exit Condition:

- User clicks on the “go back” icon
- User creates outfit log successfully
- **Flow of Events:**

1. User selects “Today” or a day from the calendar view
2. User selects an outfit by
    a. selecting a saved outfit from the built outfits in the wardrobe
    b. building an outfit for the day
    c. taking a mirror selfie to match worn outfits to the ones in the system
3. User adds the outfit to the selected day

3.5.2.24 Subscribe

Use Case Name: Subscribe
Participating Actor: Premium User
Entry Condition: User clicks on subscribe
Exit Condition:

- User clicks on the “go back” icon
- User subscribes successfully

Flow of Events:

1. User enters payment information
2. User clicks on process payment

3.5.2.25 CancelSubscription

Use Case Name: CancelSubscription
Participating Actor: Premium User
Entry Condition: User clicks on cancel subscription button
Exit Condition:

- User clicks on the “go back” icon
- User cancels subscription successfully

Flow of Events:

1. User clicks on cancel subscription button

3.5.2.26 ChangeWardrobe

Use Case Name: ChangeWardrobe
Participating Actor: Premium User
Entry Condition: User clicks on different wardrobes on the wardrobe section
Exit Condition:

- User clicks on the “go back” icon
- User changes the wardrobe successfully

Flow of Events:

1. User clicks on the wardrobe that they want to view

3.5.2.27 ProcessPayment

Use Case Name: ProcessPayment
Participating Actor: Payment System
Entry Condition: User clicks on process payment button
Exit Condition:

- Payment is done successfully
- An error occurs

Flow of Events:

1. Redirect to third party
2. Process payment

3.5.3 Object and Class Model

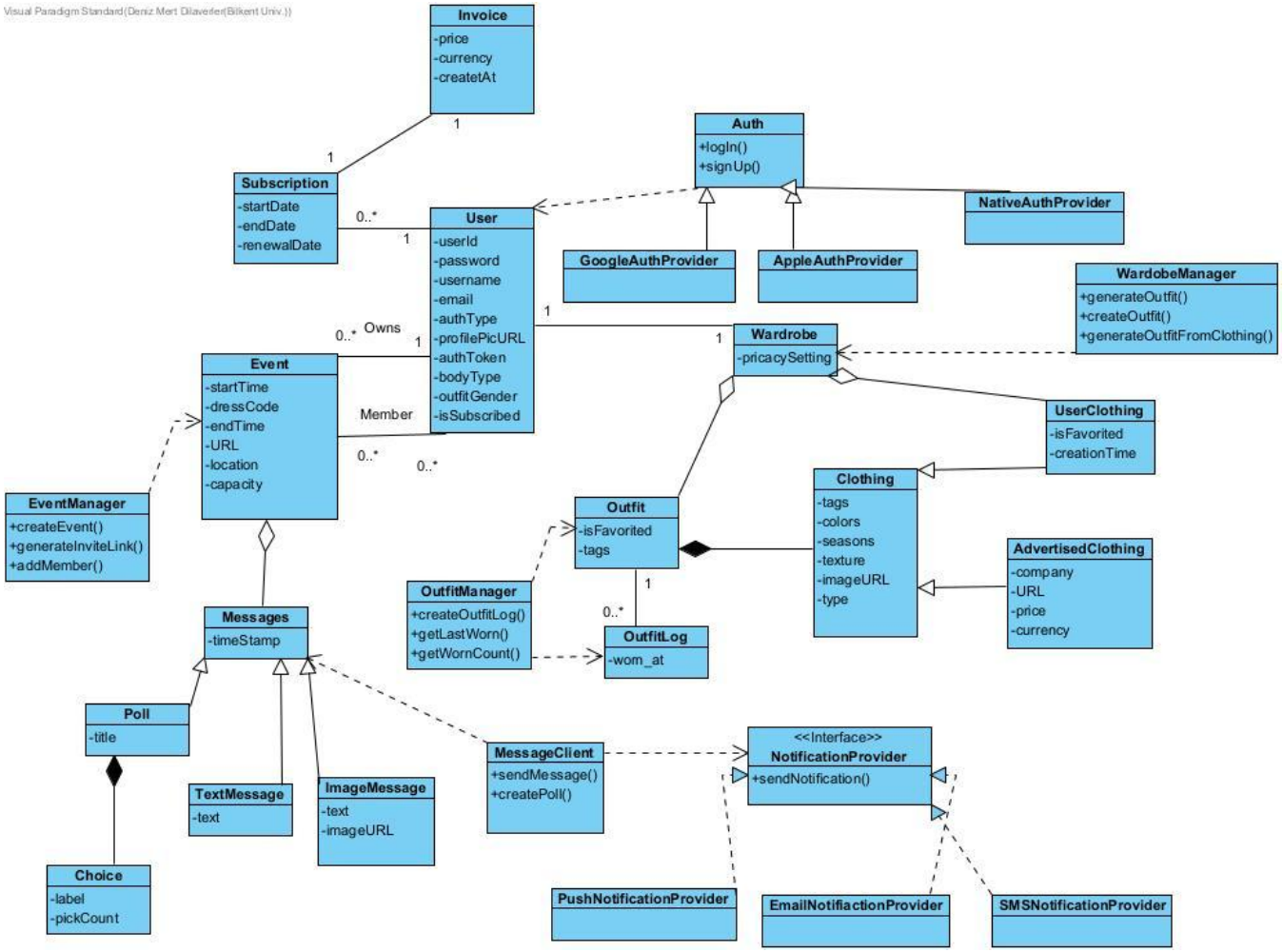

Fig. 2: Object and Class Model

3.5.3.1 User

The User class represents a user of the system. It has the following attributes:

- userId: A unique identifier for the user.
- username: The user's username.
- email: The user's email address.
- password: The user's password.
- profilePicURL: A URL to the user's profile picture.

3.5.3.2 Auth

The Auth class represents an authentication provider that the user can use to log in to the system. This class is extended by GoogleAuthProvider, AppleAuthProvider, and NativeAuthProvider, which implement the Auth classes' methods according to their contexts. The Auth class also has the following operations:

- login(): Logs the user into the system.
- signUp(): Creates a new user account.

3.5.3.3 Subscription

The Subscription class represents a user's subscription to the system. It has the following attributes:

- startDate: The date on which the subscription started.
- endDate: The date on which the subscription ends.
- renewalDate: The date on which the subscription will be renewed if it is a recurring subscription.

3.5.3.4 Invoice

The Invoice class represents a log of a purchase made for a purchase. It has the following attributes:

- price: Amount of the purchase
- currency: Currency the amount corresponds to
- createdAt: Time the invoice was created

3.5.3.5 Wardrobe

The Wardrobe class provides a singular sharable wardrobe object where the clothes and outfits are contained. It has the following attributes:

- privacySetting: Whether the wardrobe is open for public view or not

3.5.3.6 WardrobeManager

The WardrobeManager class manipulates Wardrobe objects to edit its clothing and/or outfit contents. It has the following operations:

- generateOutfit(): Generates an outfit using our ML recommendation model.
- createOutfit(): Manually creates an outfit through user input.
- generateOutfitFromClothing(): Generates an outfit like the generateOutfit() method but locks in an outfit manually and generates an outfit around it.

3.5.3.7 Clothing

The Clothing class represents a piece of clothing. It has the following attributes:

- bodyType: The body type that the piece of clothing is designed for.
- tags: A list of tags that describe the piece of clothing.
- seasons: A list of seasons that the piece of clothing is suitable for.
- texture: The texture of the piece of clothing.
- imageURL: A URL to an image of the piece of clothing.
- colors: The colors found within a clothing item.
- type: The type of clothing it is.

3.5.3.8 UserClothing

The UserClothing class is the extension of the Clothing class. This class also includes user-relevant attributes. It has the following attributes as well as the Clothing class's:

- isFavorited: Whether this clothing is in the user's favorites list.
- creationTime: When the clothing item was created.

3.5.3.9 AdvertisedClothing

The AdvertisedClothing class extends the Clothing class in addition to more commercial properties. It has the following properties:

- company: The company selling the product.
- URL: URL of the listing.
- price: The price of the listing.
- currency: Currency the listing is posted on.

3.5.3.10 Outfit

The outfit class aggregates clothing objects to create a full outfit. Its properties are as follows:

- isFavorited: The list of favorited outfits.
- tags: The tags for the outfit.

3.5.3.11 OutfitLog

The OutfitLog class's purpose is to record when an outfit is worn. Its properties are as follows:

- wornAt: When the outfit was worn.

3.5.3.12 OutfitManager

The OutfitManager manages outfit objects by creating new outfits or creating outfit objects.

- createOutfitLog(): Creates a log of an outfit.
- getLastWorn(): Returns the last worn date of the outfit.
- getWornCount(): The amount of times the outfit is worn.

3.5.3.13 Event

The Event class encapsulates an event that people are attending and want to discuss outfits for attending the said event. The properties are as such:

- startTime: The time the event is starting.
- dressCode: The selected dress code for the event.
- endTime: The time the event is ending.
- URL: Event URL for adding users.
- location: Location of the event.
- capacity: Max number of participants an event can handle

3.5.3.14 EventManager

The EventManager class manipulates an Event object. Its properties are as follows:

- createEvent(): Creates an event.
- generateInviteLink(): Creates an invite link for the event.
- addMember(): Adds a member to the event.

3.5.3.15 Message

The Message class represents a message sent in an event group. It has the following properties:

- timeStamp: The timestamp the message was sent

3.5.3.16 TextMessage

The TextMessage class extends the Message class and corresponds to a text message. Its properties are as follows:

- text: The content of the text message

3.5.3.17 ImageMessage

The ImageMessage class extends the Message class and corresponds to a message with an image attachment. It has the following properties:

- text: The text content of the message
- imageURL: Url of the image.

3.5.3.18 Poll and Choice

The Poll class extends the Message class, which implements a poll as a message. Poll has the following properties:

- title: Title of the poll.

and choice has the following properties:

- label: The label of the choice field
- pickCount: The number of times the field was picked.

3.5.3.19 MessageClient

The MessageClient class provides an interface to manage messages. It has the following operations:

- sendMessage(): Adds a message to the event.
- createPoll(): Create a poll for the event.

3.5.3.20 NotificationProvider

The NotificationProvider is an interface that all other notification providers implement. The classes PushNotificationProvider, EmailNotificationProvider, and SMSNotificationProvider. The implemented operation of the interface is as follows:

- sendNotification(): Sends a notification to the user.

3.5.4 Dynamic Models

3.5.4.1 State Machine Models

There are three options for registering a new clothing item to one's wardrobe: from camera, from gallery/camera roll, or from the web. After the user selects an option and submits an image or a URL containing an image of the desired item, the system analyzes the image and returns a result to the user. Then, the user optionally configures details for the item and adds it to their wardrobe. The state machine diagram in Figure 3 summarizes this process.

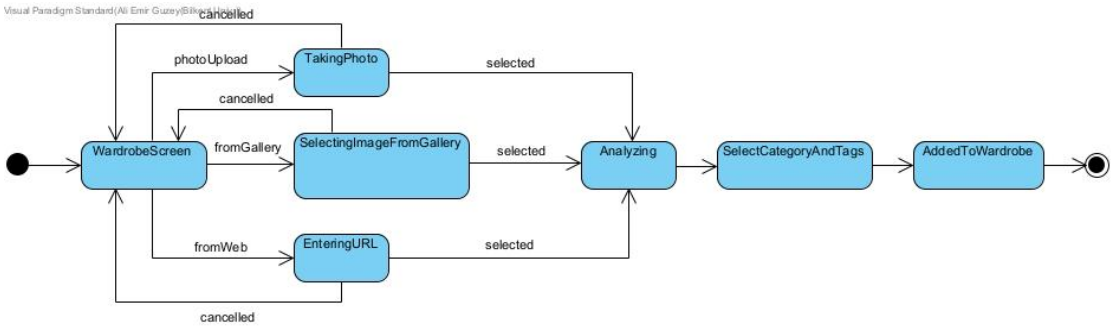

Fig. 3: State Machine Diagram of Uploading New Clothing

There are two ways to create a new outfit. One is to ask the AI engine to create a new one based on a desired category or an event, and the other is to create a new outfit manually by selecting individual items from the wardrobe. When asked the AI engine to create an outfit, it generates a sequence of outfits and the user swipes to view the next outfit until they like one. Lastly, the user optionally configures other details for the outfit, such as entering a name or a date to wear the outfit. Figure 4 depicts these steps in a state machine diagram.

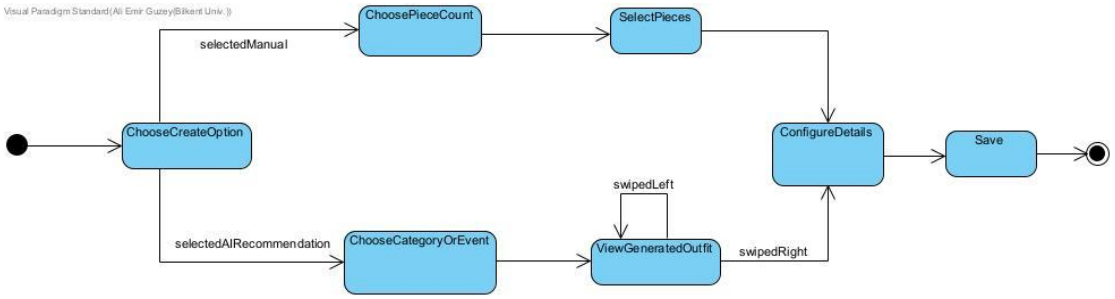

Fig. 4: State Machine Diagram of Creating a New Outfit

3.5.4.2 Activity Models

There are three options for registering a new clothing item to one's wardrobe: from camera, from gallery/camera roll, or from the web. The user selects an option and submits an image or a URL containing an image of the desired item. If the user has selected to register an item through a web link, the Django server performs an additional activity to scrape the webpage linked to the given URL to extract the image. Later, in all options, the image is sent to the AI engine to recognize and categorize the clothing in the image. Then, the system returns a result the the user, and the user optionally configures details for the item and adds it to their wardrobe if they are satisfied with the result. The activity in Figure 5 summarizes this process.

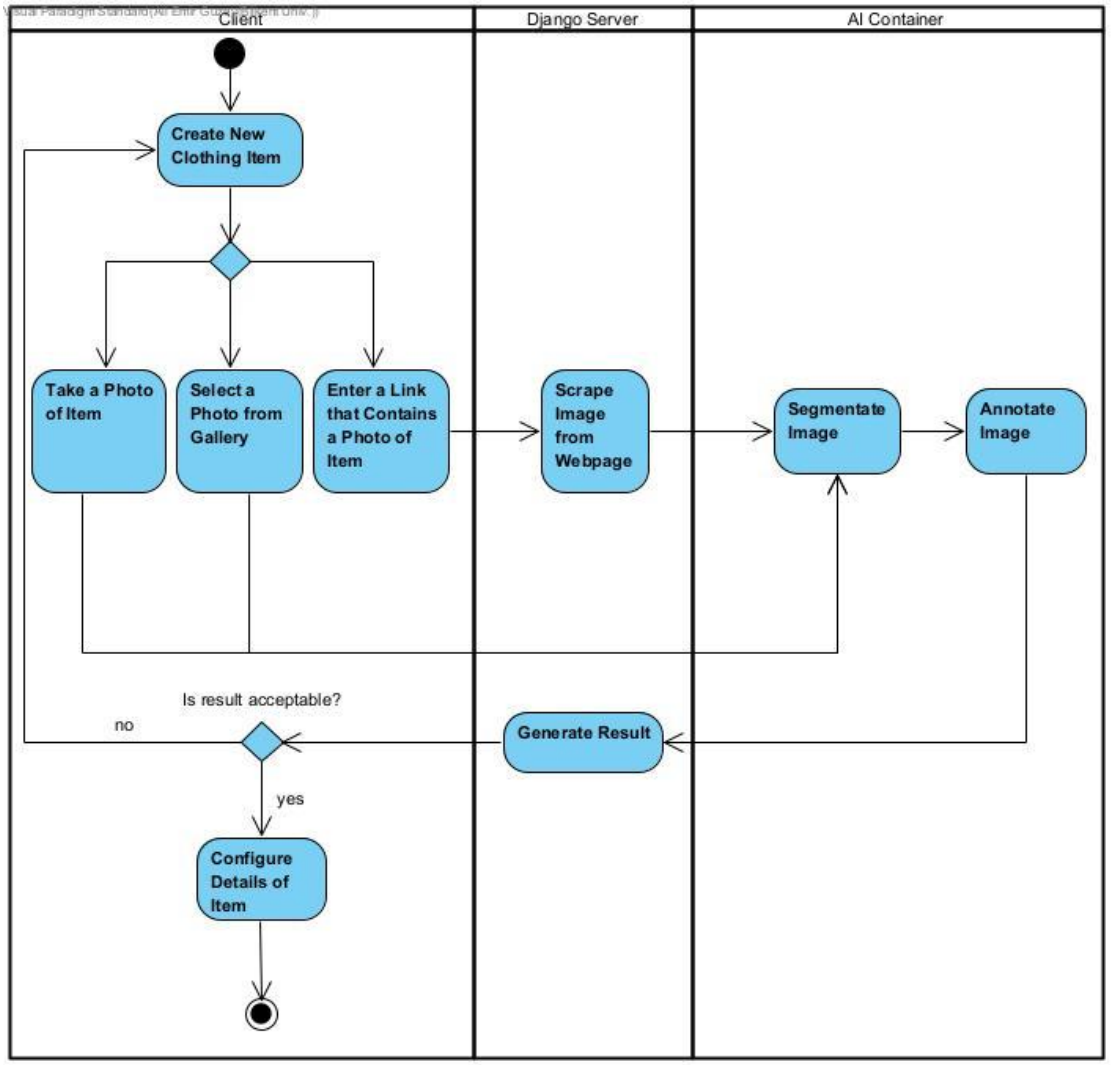

Fig. 5: Activity Diagram of Creating a New Clothing Item

To collaborate on a dress code for an event through Capsule, a user needs to create an event in the application. After creating, they send a link to the participants of the event that redirects other users to the created event in the app. If the event creator allows polling, participants vote in a poll to decide on the dress code that they will follow for the event. Then, having the dress code set, each individual is required to send an outfit from their wardrobe to the event's temporary chatbox. Participants of the event can discuss the outfits through the chat. The activity diagram in Figure 6 shows this process.

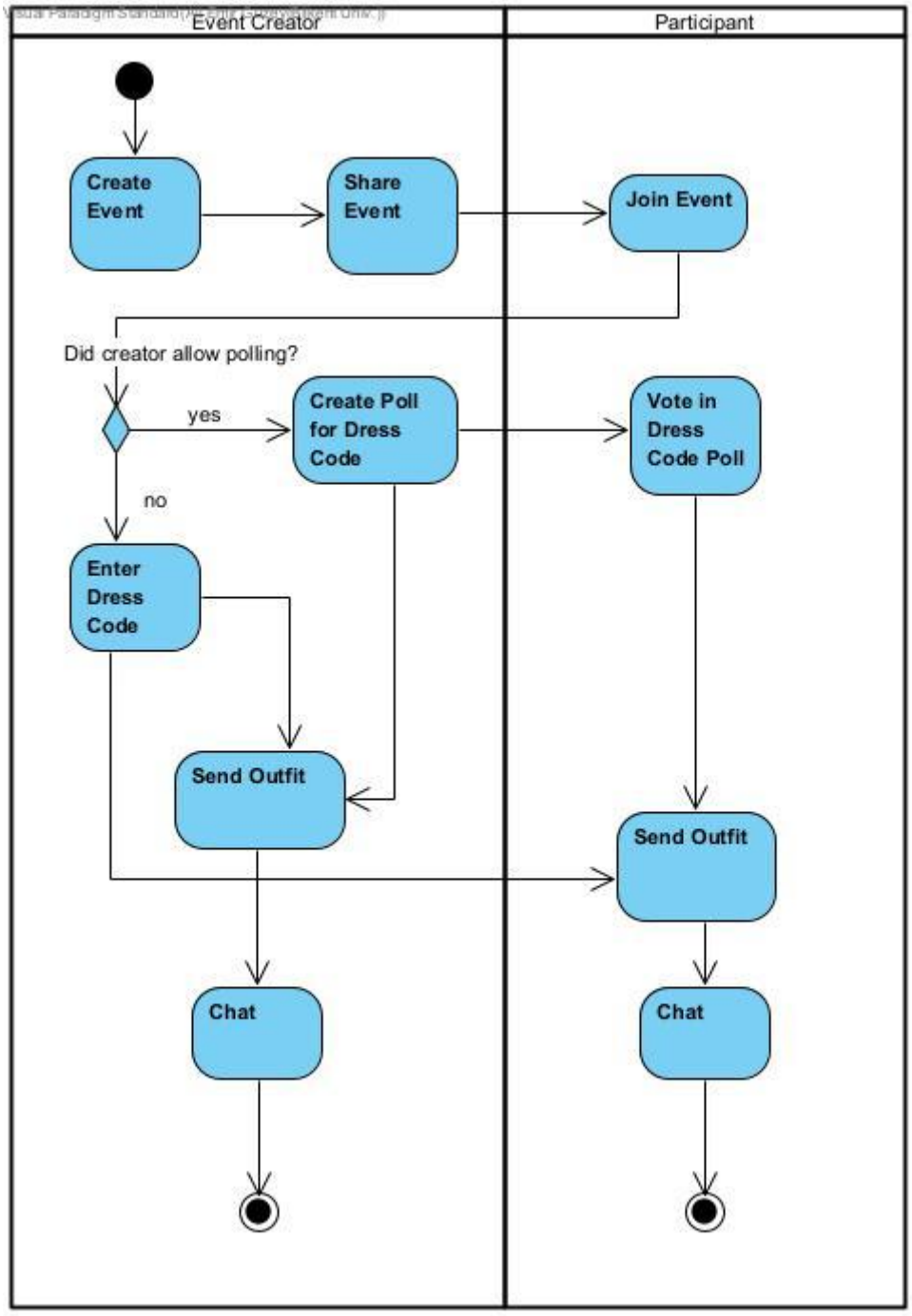

Fig. 6: Activity Diagram of Event Dress Code Collaboration

3.5.4.3 Sequence Model

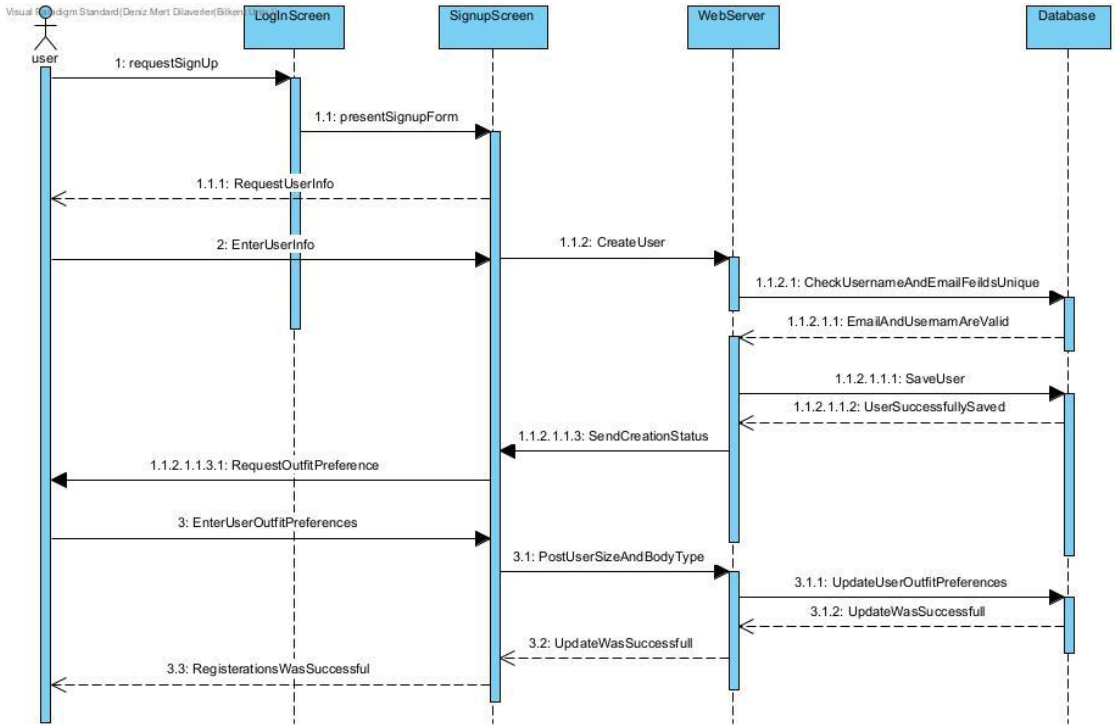

Fig. 7: Sequence diagram for signup use-case

The signup cycle starts by selecting signup on the login page and entering user information like username, email, password, and date of birth. The request is sent to the web server, and after the validations are done, the user is created. Finally, the user enters outfit preferences like size and body type. These preferences are also sent to the webserver, and the database is updated.

3.5.5 User Interface

3.5.5.1 Login Screen

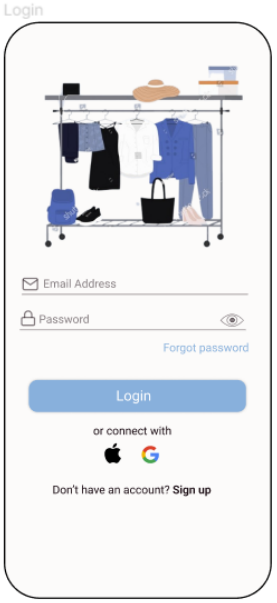

Fig. 8: Login Screen

The user enters their email address and password on this screen to log in. If the user forgets their password, they can press “Forgot Password.” They can also connect with Apple or Google by clicking on their icon. If a user does not have an account, they can register by pressing “sign up.”

3.5.5.2 Sign Up Screen

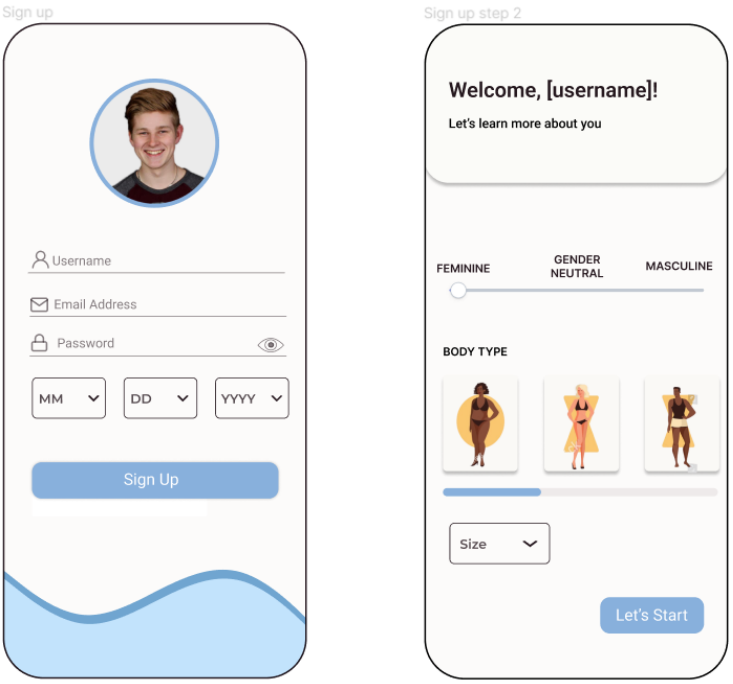

Fig. 9: Sign Up Screen

On this screen, the user enters their email address, password, username, and birth date. When clicking on sign up, the user enters their clothing preference, body type, and sizes.

3.5.5.3 Forgot Password Screen

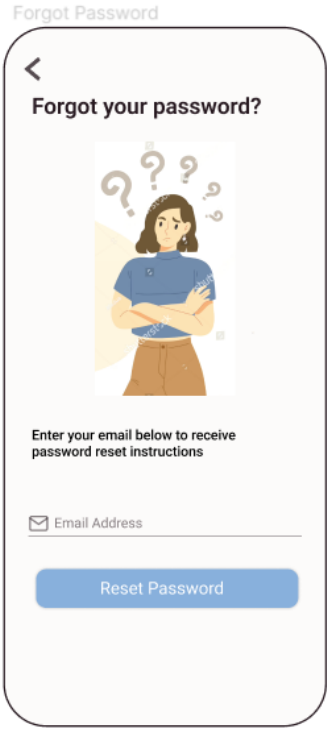

Fig. 10: Forgot Password Screen

On this screen, the user enters their email address to reset their password. When clicked on the button, a reset link will be sent to their email address.

3.5.5.4 View Wardrobe

Fig. 11: View Wardrobe

On this screen, user can view their items. The user can view the favorite items when pressing the favorite icon. When clicked on the filters, the user can filter the items according to color, category, or subcategory. Users can change between wardrobes if they have more than one wardrobe. They can also add a wardrobe if they are a premium user. (an additional add icon will be added near the heart icon in that case)

3.5.5.5 View Profile

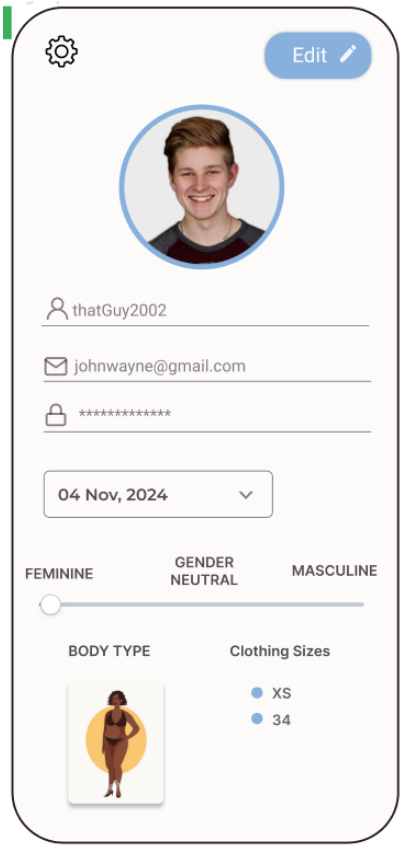

Fig. 12: Profile Screen

On this screen, user can view their profile information. They can click on the settings icon to go to the settings screen. They can also edit their profile information.

3.5.5.6 View Item

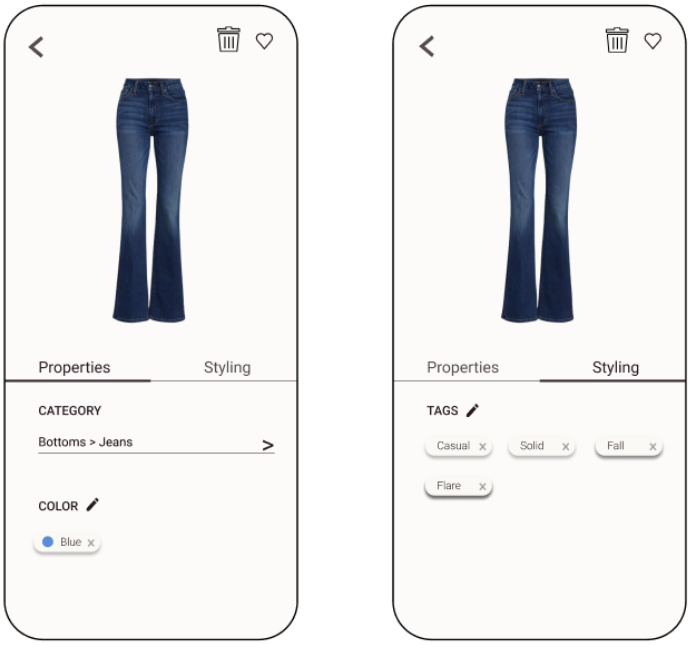

Fig. 13: View Item Screen

On this screen, users can view an item. They can also edit its properties and stylings. They can add/remove an item from favorites. They can also delete an item by clicking on the delete icon.

3.5.5.7 Register Item Menu

Fig. 14: Register Item Menu Options

On this screen, users can press the plus icon to register a new item. They have three options: take a photo, choose from the camera roll, or add a shopping link.

3.5.5.8 Register Item Segmentation Step

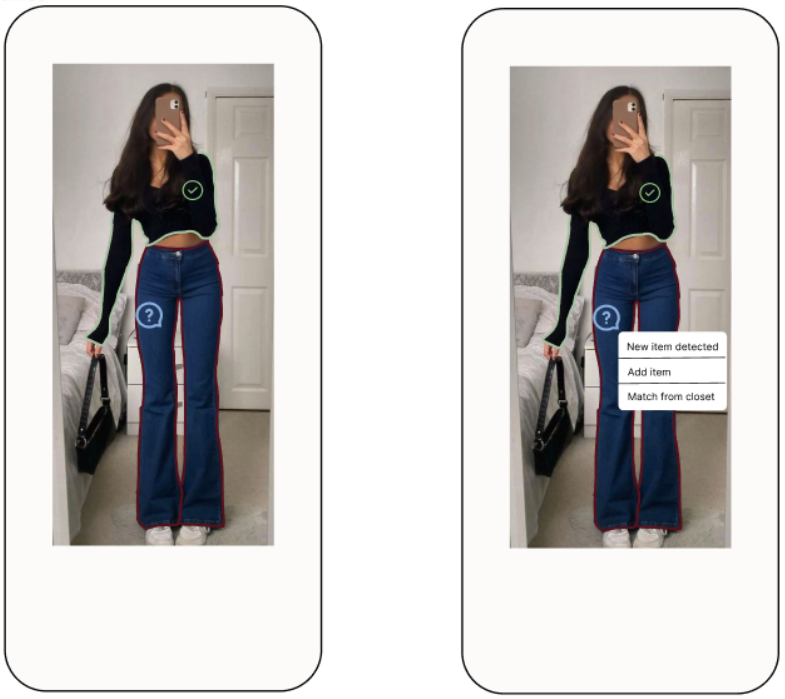

Fig. 15: Segmentation Step

On this screen, after uploading a photo with one of the three options, the mobile app shows items that are already registered and items that are not recognized. User has the option to add the item or match from the closet for both the recognized and unrecognized items.

3.5.5.9 Add Item

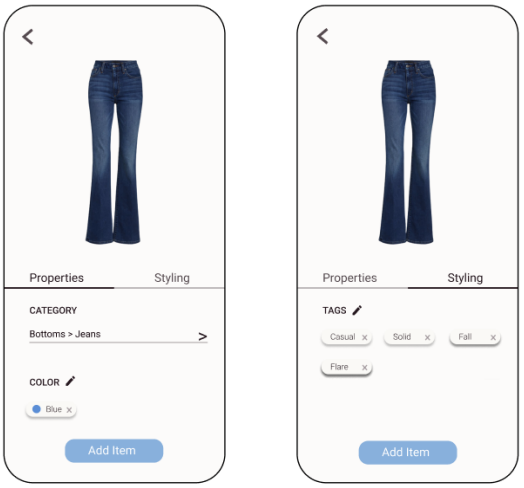

Fig. 16: Add Item Screen

On this screen, users can edit its properties and stylings. When pressed the add item button, it adds it to the wardrobe.

3.5.5.10 Create Outfit Manually

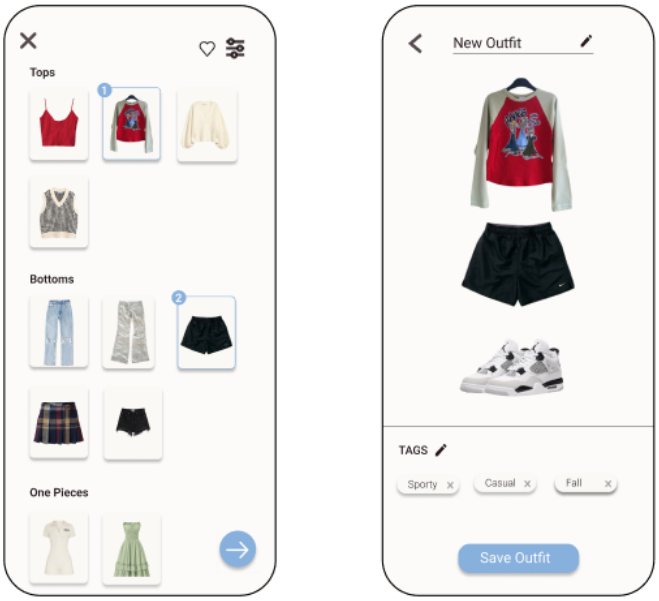

Fig. 17: Create Outfit Manually Screens

On these screens, first, the user selects the outfit length. (One-pieces and shoes, top-bottom and shoes, or an additional coat to both options). After that, the user selects the items. Then, the user can change the outfit name and tags. By pressing “save outfit,” the user saves the outfit.

3.5.5.11 Create Outfit With AI

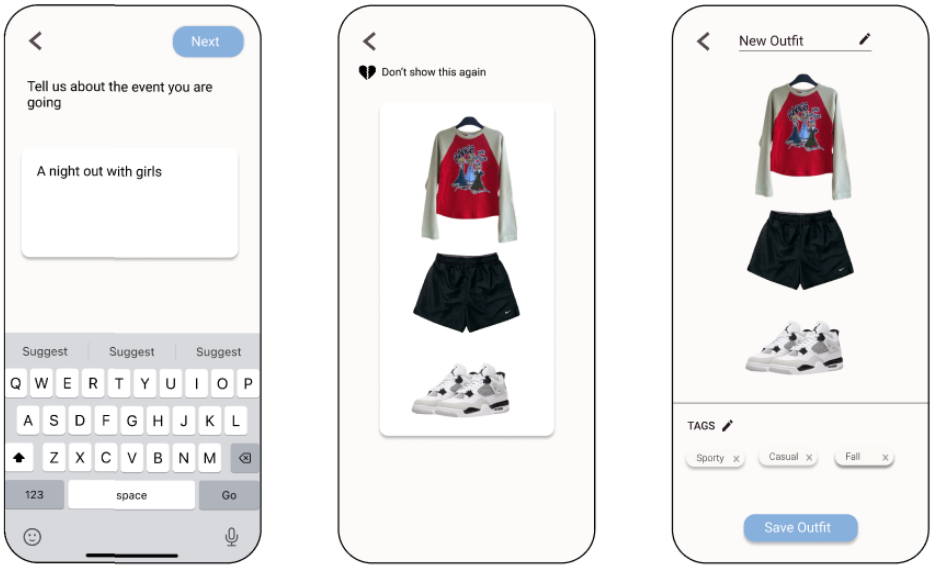

Fig. 18: Create Outfit with AI Screens

On these screens, first, the user must describe the event they are going to. Then, the system will generate an outfit accordingly. After that, the user can swipe left and right to decide on an outfit. If the user wants an outfit not to get recommended again, the user can press the “don’t show this again” button. After deciding on an outfit, the user can edit the outfit name edit tags. When finished, the user can press the “save outfit” button to save the outfit.

3.5.5.12 View Outfits

Fig. 19: Outfits Screen

On this screen, users can view their outfits. When pressed on an outfit, they can edit and view it in more detail. To generate an outfit, they can press the “plus” button.

3.5.5.13 Create Outfit Log and View

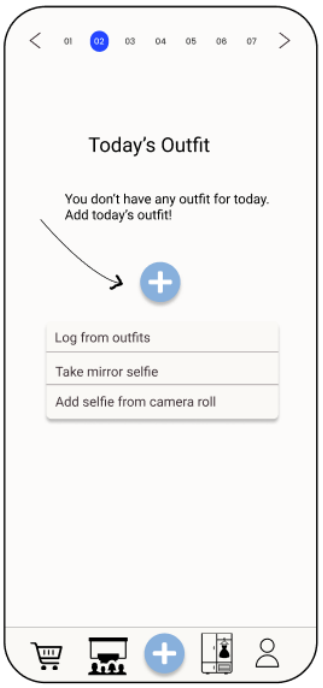

Fig. 20: Outfit Log Screen

On this screen, the user can view outfit logs (if exist) on a selected day. For any day, the user can add a log from existing outfits, take a selfie, or upload a selfie.

3.5.5.14 Creating and Participating in Events

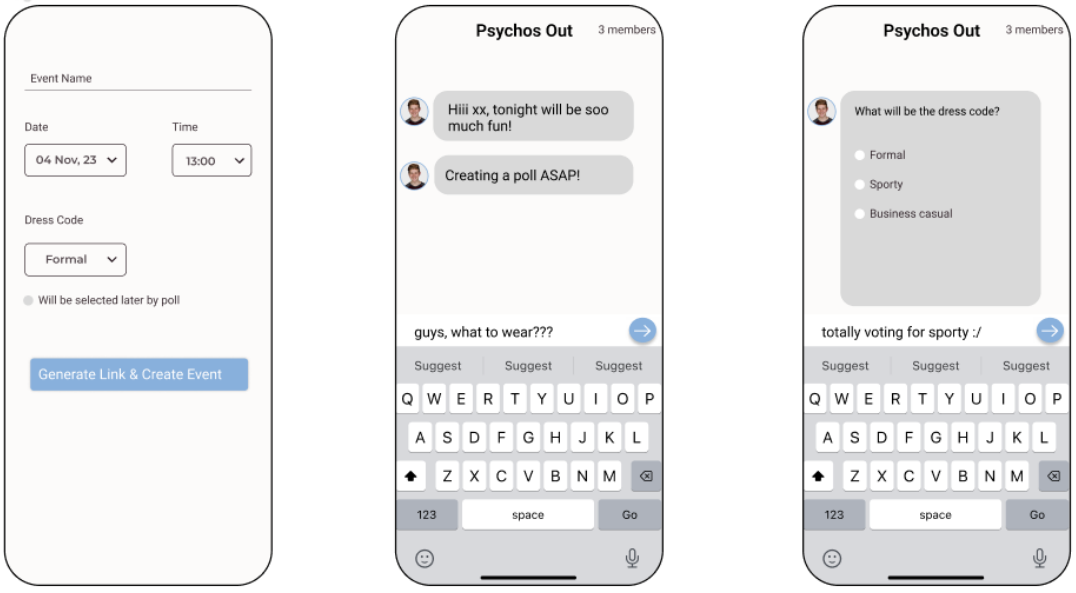

Fig. 21: Event Screens

A user can create an event on the first screen by filling out the required information. The generated link can be shared with people. People can join the event by clicking on the link. When joined, users can chat and share outfits. If the dress code still needs to be decided, they can create a poll and decide on a dress code.

3.5.5.15 View Settings

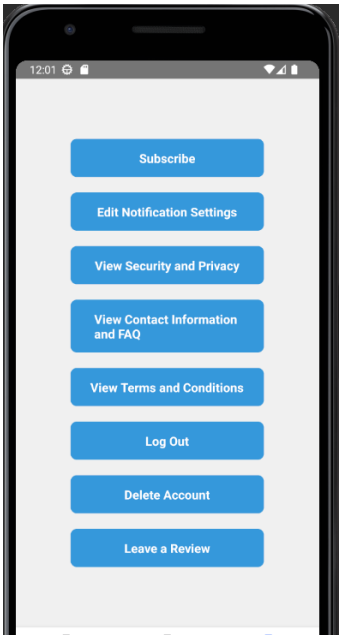

Fig. 22: Settings Screen

The user can subscribe to our app to become a premium user on the settings screen. If the user is already premium, they can cancel a subscription instead. They can edit notification settings, view security and privacy, view contact information and FAQ, and view terms and conditions. They can also log out or delete their account. Finally, they can leave a review on the App Store/Google Play Store.

3.5.5.16 View Analytics

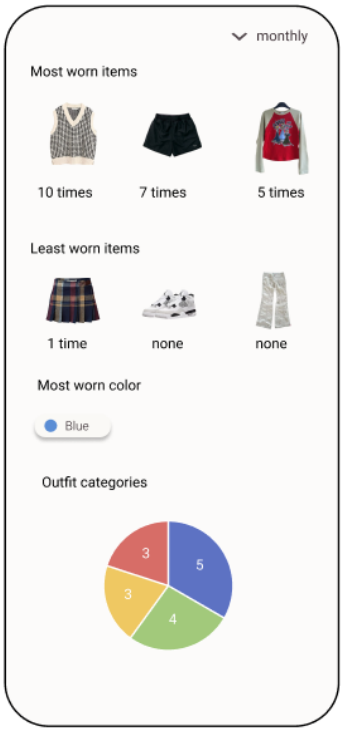

Fig. 23: View Analytics Screen

On this screen, users can view their most worn items, least worn items, most worn color, and the outfit categories distribution as a pie chart.

3.5.5.17 Items Recommendation Screen

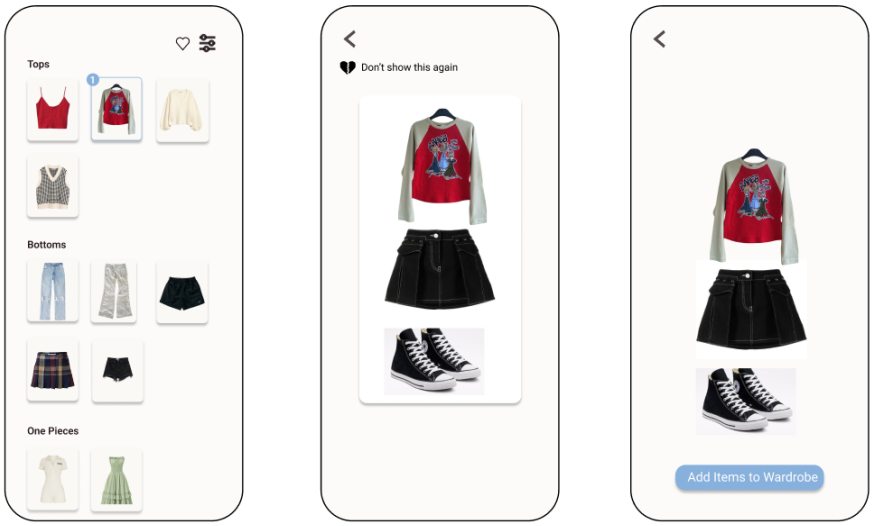

Fig. 24: Item Recommendation Screens

On these screens, the user selects an item piece (or more than one). Then, the AI recommends the remaining items from the partners' clothing pieces. If the user likes an outfit,

they can swipe right. After that, they can press on the items to quickly buy them. To instantly add them to their wardrobe, they can press the “add items to wardrobe” button.

4 Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

In the engineering design of Capsule, we need to consider various factors that may impact our application’s functionality, usability, and sustainability. These factors are discussed in the following subsections

4.1.1 Constraints

4.1.1.1 Public Health and Safety

We must ensure that our application does not compromise the safety and well-being of our users. For example, we should ensure that user data, including images, are handled carefully and that privacy and security measures are in place.

4.1.1.2 Data Security

To prevent unauthorized access to user data, data need to be protected both at transit during client-server communication and at rest on the server side. Necessary authentication and authorization systems need to be implemented, and data should be sent to the client after ensuring the client is the owner of the requested data.

4.1.1.3 Data Privacy and Regulations

Our app should respect the privacy of users’ data and comply with related local regulations, such as the General Data Protection Regulation (GDPR) for Europe and, for example, the localized Turkish version, KVKK. Personally Identifiable Information (PII) of users should not be collected unless required to provide our service, and only a minimal amount of information needs to be collected and stored. In addition to the transparency about which data is collected and how it is used, collected information should be permanently deleted upon the request of a user.

4.1.1.4 Ethical Limitations

Any data collection or processing that may lead to ethical uncertainties should be prevented, and ethical considerations should be in focus throughout the project's development. For example, while users need to send their photos to servers for processing and extraction of clothing-related data, having personal photos of users on the server in a human-readable form would lead to ethical and privacy-related concerns. To overcome this issue, faces can be blurred before being transferred to the server for processing in order to minimize the collected PII, increase user privacy, and resolve ethical concerns..

4.1.1.5 Computational Limitations

As the processing of user data will either be done locally on user’s device or remotely on our servers, the amount of data to process will be limited by the computational capabilities of those machines. Provided service will depend on the time of computation and the virtual space, memory, or disk space spared for the computations. Recognizing that the AI engine will

perform demanding tasks to recognize and categorize user images, this constraint will be significant throughout the development of Capsule.

4.1.2 Standards

4.1.2.1 Sociocultural Factors

Clothing choices can vary greatly across cultures and societies. We need to be sensitive to cultural differences and social norms to provide respectful and appropriate recommendations for a diverse user base.

4.1.2.2 Environmental Impact

The fashion industry has a significant environmental footprint. Capsule aims to reduce unnecessary clothing consumption, but we should also consider the environmental impact of our solution, such as promoting sustainable clothing brands and practices.

4.1.2.3 Economic Factors

By helping users make more informed clothing choices, we aim to reduce unnecessary spending. However, we should be aware of the potential economic impact on the fashion industry and consider strategies for sustainable growth and partnerships.

4.1.2.4 Global Reach

Capsule has the potential to reach a global audience. We must take into account the diverse fashion preferences and needs of users from different parts of the world and adapt our system accordingly.

4.2 Risk and Alternatives

Identifying potential risks and developing alternative plans is crucial for the success of our project. The following subsections discuss some of the risks associated with the development of Capsule and our proposed alternatives.

4.2.1 Privacy and Data Security

Risk: User data breaches or misuse.
Alternative: Implement strong encryption, user consent mechanisms, and regular security reviews.

4.2.2 User Adoption

Risk: Users may not embrace the idea of photographing their outfits daily.
Alternative: Implement a user-friendly interface, provide incentives, and conduct user feedback surveys to improve user experience.

4.2.3 Machine Learning Accuracy

Risk: The ML model may not accurately match outfits.
Alternative: Continuous model training, user feedback, and manual outfit matching options.

4.2.4 Competitive Market

Risk: Competing with established fashion apps.

Alternative: Focus on unique features like collaborative dress code setting and targeted advertising for clothing retailers.

4.2.5 Technological Challenges

Risk: Technical difficulties in implementing computer vision.
Alternative: Collaborate with experts, continuously update technology, and have backup technical solutions in place.

4.3 Project Plan

Table 2: Factors that can affect analysis and design.

	Effect level (0 - None to 10 - Maximum)	Effect
Public health and safety	9	We must prioritize user data security and privacy to ensure the safety and well-being of our users. Implementation of robust security measures is essential.
Sociocultural factors	7	We need to be sensitive to cultural differences and social norms to provide recommendations that are respectful and appropriate for a diverse user base.
Environmental factors	8	Capsule aims to promote sustainable clothing practices. We must consider the environmental impact of our solution, such as endorsing eco-friendly clothing brands and responsible consumption.
Economic factors	5	While helping users make informed clothing choices, we need to be aware of the potential economic impact on the fashion industry and consider strategies for sustainable growth and partnerships.
Global factors	7	Capsule aims to serve a global audience. We must consider the diverse fashion preferences and needs of users from different parts of the world and adapt our system accordingly.

Table 3: Risks

	Likelihood (Low/Medium/High)	Effect on the project	B Plan Summary
Privacy and Data Security	Low	Low	Implement strong encryption, user consent mechanisms, and regular security audits.
User Adoption	Medium	High	Implement a user-friendly interface, provide incentives, and conduct user feedback surveys to improve user experience.
Machine Learning Accuracy	High	High	Continuous model training, user feedback, and manual outfit matching options.
Competitive Market	Low	Medium	Focus on unique features like collaborative dress code setting and targeted advertising for clothing retailers.
Technological Challenges	Low	High	Collaborate with experts, continuously update technology, and have backup technical solutions in place.

Table 4: List of work packages

WP#	Work package title	Leader	Members involved
WP1	Product - UI/UX Design	Ceren Akyar	Deniz Mert Dilaverler
WP2	Management - Report Preparation	Deniz Mert Dilaverler	Ali Emir Guzey, Alp Afyonluoglu, Ceren Akyar, Mehmet Kagan Ilbak
WP3	Product - Website	Ceren Akyar	Ali Emir Güzey
WP4	Product - Mobile Application	Ceren Akyar	Deniz Mert Dilaverler
WP5	Product - Backend API	Deniz Mert Dilaverler	Ali Emir Güzey, Alp Afyonluoglu
WP6	Infrastructure - Machine Learning	Alp Afyonluoglu	Mehmet Kagan Ilbak
WP7	Infrastructure - Cloud	Ali Emir Guzey	Deniz Mert Dilaverler, Mehmet Kagan Ilbak
WP8	Product - Machine Learning Segmentation	Alp Afyonluoglu	Mehmet Kagan Ilbak
WP9	Product - Machine Learning Recommendation	Mehmet Kagan Ilbak	Alp Afyonluoglu
WP10	Infrastructure - DevOps	Ali Emir Guzey	Mehmet Kagan Ilbak
WP11	Infrastructure - Privacy	Mehmet Kagan Ilbak	Deniz Mert Dilaverler

WP 1: Product - UI/UX Design
Start date: *31.10.2023* **End date:** *30.11.2023*

Leader:	*Ceren Akyar*	**Members involved:**	*Deniz Mert Dilaverler*
Objectives: *The goal is creating beautiful user interfaces with the best user experience. UI should be intuitive. User experience should be smooth.*			
Tasks: ***Task 1.1 Create UI mockups*** **:** *Creating UI mockups on Figma.* ***Task 1.2 Create UX designs*** **:** *Designing UX of the app.* ***Task 1.3 Implement primitive UI*** **:** *Implement UI elements such as signup, login, home page, and settings.* ***Task 1.4 Implement critical UI*** **:** *Implement critical UI elements such as wardrobe, recommendation, and cloth addition.*			
Deliverables ***D1.1:*** *UI design on Figma.* ***D1.2:*** *UX design on Figma.* ***D1.3:*** *Primitive UI elements.* ***D1.4:*** *Critical UI elements.*			
WP 2: Management - Report Preparation			
Start date: *31.10.2023* **End date:** *15.05.2024*			
Leader:	Deniz Mert Dilaverler	**Members involved:**	Ali Emir Guzey, Alp Afyonluoglu, Ceren Akyar, Mehmet Kagan Ilbak
Objectives: *This package will allow us to communicate our plans for the application. We will then write reports to solidify our plans.*			
Tasks: ***Task 2.1 Project information form*** **:** *We will use the form to have a clear idea of the application.* ***Task 2.2 Assessment of Innovation form*** **:** *We will consult our innovation expert to have a better understanding of the technical aspects of our application.* ***Task 2.3 Project Specification Document*** **:** *We will have a clear idea of features, limitations, and requirements of our application.* ***Task 2.4 Analysis and Requirements Report*** **:** *We will have an in-depth understanding of our project with this report. We will determine the detailed view of our design with this report.*			
Deliverables ***D2.1:*** *Project information form* ***D2.2:*** *Assessment of Innovation form* ***D2.3:*** *Project Specification Document* ***D2.4:*** *Analysis and Requirements Report*			
WP 3: Product - Website			
Start date: *01.10.2023* **End date:** *15.05.2024*			
Leader:	Ceren Akyar	**Members involved:**	Ali Emir Güzey
Objectives: *This package includes tasks that aim to improve our website over time.*			
Tasks: ***Task 3.1 Create a website with landing page*** **:** *A landing page which briefly describes our product and introduces our team.* ***Task 3.2 Deploy the website with domain*** **:** *This task will allow guests to get an idea for our app as well as meet our team.*			
Deliverables ***D3.1:*** *Frontend of our website.*			

D3.2: *Deployed website which can be accessed with a domain.*			
WP 4: Product - Mobile Application			
Start date: *01.10.2023* **End date:** *15.05.2024*			
Leader:	Ceren Akyar	**Members involved:**	Deniz Mert Dilaverler
Objectives: *The mobile application is our main customer facing product. This package aims to create and improve our product.*			
Tasks: ***Task 4.1 Create the application based on mockups*** **:** *Create a live version of our mobile application based on the previously created mockups.* ***Task 4.2 Connect the mobile application to our backend*****:** *This task will connect our mobile application to our backend. This is essential for the functionality of our application.*			
Deliverables ***D4.1:*** Mobile application ***D4.2:*** *Connected mobile application*			
WP 5: Product - Backend API			
Start date: *01.10.2023* **End date:** *15.05.2024*			
Leader:	Deniz Mert Dilaverler	**Members involved:**	Ali Emir Güzey, Alp Afyonluoglu
Objectives: *This package will allow us to create the backend API of our application. This backend API will be our core server.*			
Tasks: ***Task 5.1 Create a Django application*** **:** *This django application will be our main server.* ***Task 5.2 Write endpoints for authentication*** **:** *This will allow users to authenticate.* ***Task 5.3 Write endpoints for outfits*** **:** *This will allow users to store and create outfits.*			
Deliverables ***D5.1:*** *Django application* ***D5.2:*** *Authentication endpoints* ***D5.3:*** *Outfits endpoints.*			
WP 6: Infrastructure - Machine Learning			
Start date: *31.10.2023* **End date:** *15.12.2023*			
Leader:	Alp Afyonluoglu	**Members involved:**	Mehmet Kagan Ilbak
Objectives: *This package will provide a uniform infrastructure for machine learning tasks. This includes frameworks, tools, and deployment of machine learning tasks.*			
Tasks: ***Task 6.1 Machine learning training pipeline*****:** *This will allow us to train our models across different machines without any issues uniformly.* ***Task 6.2 Machine learning model deployment*****:** *This will allow us to deploy our trained models. The machine learning API will use these models to handle requests.*			
Deliverables ***D6.1:*** *Machine learning pipeline* ***D6.2:*** *Machine learning deployment*			
WP 7: Infrastructure - Cloud			
Start date: *01.01.2024* **End date:** *15.05.2024*			
Leader:	Ali Emir Guzey	**Members involved:**	Deniz Mert Dilaverler, Mehmet Kagan Ilbak
Objectives: *Cloud infrastructure will allow us to create a more systematic deployment environment.*			
Tasks:			

Task 7.1 Product Infrastructure **:** *This will handle our backend and machine learning deployments.*			
Deliverables ***D7.1:*** *Product Infrastructure*			
WP 8: Product - Machine Learning Segmentation			
Start date: *30.09.2023* **End date:** *15.12.2023*			
Leader:	Alp Afyonluoglu	**Members involved:**	Mehmet Kagan Ilbak
Objectives: *This package will introduce our segmentation system. This system will interact with user data to categorize clothes. This data will later be used on product and machine learning systems.*			
Tasks: ***Task 8.1 Research about clothing segmentation systems*** **:** *Literature review will allow us to determine the scope and the complexity of the segmentation system.* ***Task 8.2 Segmentation dataset*****:** *Research about already existing segmentation datasets.* ***Task 8.3 Segmentation system:*** *Create a segmentation system.* ***Task 8.4 Segmentation API:*** *Create a segmentation API which will interact with the product.*			
Deliverables ***D8.1:*** *Segmentation dataset* ***D8.2:*** *Segmentation system* ***D8.3:*** *Segmentation API*			
WP 9: Product - Machine Learning Recommendation			
Start date: *01.02.2024* **End date:** *30.05.2024*			
Leader:	Mehmet Kagan Ilbak	**Members involved:**	Alp Afyonluoglu
Objectives: *This package will introduce our core recommendation system. This will work in sync with the segmentation system.*			
Tasks: ***Task 9.1 Research about clothing recommendation systems*** **:** *Literature review will allow us to determine the scope and the complexity of the recommendation system.* ***Task 9.2 Recommendation dataset*****:** *Research about already existing recommendation datasets. Furthermore, look into the ways of scraping websites for creating up-to-date dataset.* ***Task 9.3 Recommendation system:*** *Create a recommendation system.* ***Task 9.4 Recommendation API:*** *Create a recommendation API which will interact with the product.*			
Deliverables ***D9.1:*** *Recommendation dataset* ***D9.2:*** *Recommendation system* ***D9.3:*** *Recommendation API*			
WP 10: Infrastructure - DevOps			
Start date: *31.10.2023* **End date:** *31.12.2023*			
Leader:	Ali Emir Guzey	**Members involved:**	Mehmet Kagan Ilbak
Objectives: *Creating a development infrastructure. This includes version control pipeline, cloud services and development tools.*			
Tasks: ***Task 10.1 Create a GitHub workflow*** **:** *Creating a GitHub workflow will allow us to move faster with the development. This includes repository, pull request, and branch set up.* ***Task 10.2 Create CI/CD pipeline*** **:** *Creating a CI/CD pipeline will allow us to have easier builds.*			

Deliverables ***D10.1:*** *Fully configured GitHub workflow.* ***D10.2:*** *CI/CD pipeline that meets our needs.*			
WP 11: Infrastructure - Privacy			
Start date: *01.02.2024* **End date:** *01.06.2024*			
Leader:	Mehmet Kagan Ilbak	**Members involved:**	Deniz Mert Dilaverler
Objectives: *Enforcing KVKK and GDPR rules on the entire execution pipeline. This will allow us to create a safer experience for our users.*			
Tasks: ***Task 11.1 Privacy Analysis, Product*** **:** *Analyze the points of data collection. This includes both the frontend and the backend of the application.* ***Task 11.2 Privacy Analysis, ML*** **:** *Analyze how the collected user data is used with machine learning models.* ***Task 11.3 Taxonomy Annotations, Product*** **:** *Annotate the functions which handle user data. This will allow us to spot data collection points.*			
Deliverables ***D11.1:*** *Short document about the data collection on the product.* ***D11.2:*** *Short document about the data usage on machine learning models.*			

4.4 Ensuring Proper Teamwork

Effective teamwork is crucial for the success of our project. To ensure proper teamwork, we will implement the following strategies:

- **Shared Leadership:** All team members will have the opportunity to take on leadership roles in various aspects of the project. Leadership will be rotated to provide an inclusive and collaborative environment.
- **Collaboration Tools:** We will use collaboration and project management tools like GitHub, Jira, or Trello to track individual contributions, project progress, and task assignments.
- **Regular Communication:** We will schedule regular team meetings and maintain open channels of communication to keep all team members informed and engaged.

In addition to the principles above, we will divide the work according to the subjects and every member will be responsible for specific subjects. Details on this are given in Table xx.

4.5 Ethics and Professional Responsibilities

Ethical and professional responsibilities are central to our project. We will uphold the following ethical principles.

4.5.1 User Privacy

We will prioritize user privacy and data security, ensuring that user data is handled with the utmost care and in compliance with relevant regulations.

4.5.2 Transparency

We will be transparent about how our recommendation and advertising algorithms work, ensuring that users understand how their data is used.

4.5.3 Inclusivity

We will design our application to be inclusive, respectful of cultural and social diversity, and promote responsible clothing choices.

4.5.4 Professionalism

All team members will conduct themselves in a professional manner, respecting deadlines, and maintaining a high standard of work.

4.6 Planning for New Knowledge and Learning Strategies

This project requires the use of technologies and methods that are new for some team members in the frontend, backend, and machine learning areas. The project also focuses on a domain not commonly targeted by the software industry, especially during university education, which is clothing. As the need to learn such new information emerged, the team distributed new knowledge areas among members during work allocation to cover and learn various knowledge areas that are required for this project efficiently and effectively. Some areas needed for this project are explained below.

Starting with the machine learning (ML) side, comprehensive research of ML frameworks, libraries, models, and best practices is needed. While some group members have limited prior machine learning experience, no one has experience developing a production-ready model, which is a new area that needs to be explored. Four members started to take the Introduction to Machine Learning course at the beginning of this project's development, meaning the basics of machine learning, as well as related advanced concepts, are new knowledge areas. Since segmentation and recommendation models are needed, related models, such as U-Net and Mask-RCNN for segmentation, will be researched. Various frameworks, such as PyTorch, TensorFlow, and others, in case more efficient ones are found, will be researched. Academic source research will also be conducted to learn more about state-of-the-art ML developments, models, and solutions for segmentation and recommendation engines.

For the backend side, while members are familiar with the Python language, the use of Django will require some members to learn the framework and become familiar with the Django Rest Framework. For an efficient backend structure, image storage and cloud solutions areas, as well as the related best practices, will be researched. Also, as various engines and servers will be containerized, containerization will be another new knowledge area, which will possibly require further research to learn related best practices. For the frontend side, react native will be used. and related development tools, such as the Expo router library, will require research and getting used to the tool.

When the algorithm logic is considered, the structure needs to be efficient and scalable, which is why the time complexity of the algorithm needs to be considered and, when needed, abstract structures, such as design patterns, need to be implemented, making courses such as software engineering and algorithms a requirement. Application of these concepts will probably require

further research during the development; while members are familiar with those concepts, applying the concepts to production may require further analysis of the concepts. Other than the technical side, the solution domain and target audience analysis needs to be done, which is a new area and requires research.

5 Glossary

Capsule Wardrobe: A capsule wardrobe is a minimalist and versatile collection of essential clothing items designed to simplify and enhance one's personal style while promoting sustainable fashion. The goal is to create the maximum number of outfits with the minimum number of items.
GDPR: The EU General Data Protection Regulation
IEEE: The Institute of Electrical and Electronics Engineers
KVKK: Kişisel Verileri Koruma Kanunu
PEP8: Python Enhancement Proposal 8
PII: Personally Identifiable Information

6 References

[1] E. Newbery, “The Average American Spends This Much on Clothes Every Year,” *the ascent*, Jul. 26, 2022. [Online]. Available: https://www.fool.com/the-ascent/personal-finance/articles/the-average-american-spends-this-much-on-clothes-every-year/. [Accessed: Oct. 15, 2023].

[2] E. Penner, “The Ultimate Guide: How to Build a Capsule Wardrobe,” *Modern Minimalism*, 2023. [Online]. Available: https://modernminimalism.com/how-to-build-a-capsule-wardrobe/. [Accessed: Oct. 15, 2023].