Bilkent University

Department of Computer Engineering

CS 491 - Senior Design Project I

T2327 - Capsule

Project Specification Document

Ali Emir Güzey - Ceren Akyar - Deniz Mert Dilaverler - Mehmet Kağan İlbak - Alp Afyonluoğlu

Supervisor: Asst. Prof. Hamdi Dibeklioğlu

Course Instructors: Dr. Atakan Erdem & Mert Bıçakçı

November 1, 2023

Table of Contents

1. Introduction..3
1.1. Description..3
1.2. High-Level System Architecture & Components of Proposed Solution..4
1.2.1. Mobile Interface..4
1.2.2. Django Web Server..5
1.2.3. PostgreSQL Database..5
1.2.4. Labeling Engine..5
1.2.5. Recommendation Engine..5
1.2.6. Matching Engine..5
1.2.7. AWS S3 Bucket..5
1.2.8. Google Vision API..5
1.3. Constraints..6
1.3.1. Economic Constraints..6
1.3.2. Ethical Constraints..6
1.3.3 Political Constraints..6
1.4. Professional and Ethical Issues..6
1.4.1 Privacy and Data Security:..6
1.4.2 Algorithmic Bias and Fairness:..7
1.4.3 Informed Decision-Making:..7
1.4.4 User Consent for Advertisement Targeting:..7
1.4.5 Collaborative Decision-Making:..7
1.4.6 Image Processing and Consent:..7
1.4.7 Selling Recommendations:..7
1.4.8 Intellectual Property and Copyright:..7
1.4.9 User Profiling:..8
1.5 Standards..8
1.5.1 Reports..8
1.5.2 Codebase..8
1.5.3 Commit Messages..8
1.5.4 Issue Tracker..8
2. Design Requirements..9
2.1. Functional Requirements..9
2.1.1. User Profile Creation and Management..9
2.1.2. Event Creation and Management..9
2.1.3. Event Discussion and Dress Code Setting..9
2.1.4. Event Notifications..9
2.1.5. Privacy Controls..9
2.1.6. Recommendation and Suggestion Integration..9
2.1.7. Real-time Chat and Messaging..10

2.1.8. Event Reminders........ 10
2.1.9. Outfit Recommendation........ 10
2.1.10. Personalization........ 10
2.1.11. Built-in Access to Items From Brands........ 10
2.1.12. Image Recognition........ 10
2.1.13. Wardrobe Tracking........ 11
2.1.14. Wardrobe Processing........ 11
2.1.15. User Analytics........ 11
2.1.15. Building Outfits Manually........ 11
2.1.16 Outfit Management........ 11
2.1.17 Signup and Login........ 11
2.1.18 Reset Password........ 12
2.1.19 Selecting Outfits For a Day........ 12
2.1.20 Viewing/Managing Wardrobe........ 12
2.1.21 View Advertised Outfits........ 12
2.2. Non-Functional Requirements........ 13
2.2.1. Usability........ 13
2.2.2. Reliability........ 13
2.2.3. Performance........ 13
2.2.4. Supportability........ 13
2.2.5. Scalability........ 13
3. Feasibility Discussions........ 13
3.1. Market & Competitive Analysis........ 14
3.1.1. Target Audience........ 14
3.1.2. Competitive Landscape........ 14
3.2. Academic Analysis........ 16
3.3. Risks........ 17
3.4. Business Plan........ 18
Acronyms and Abbreviations........ 19
References........ 20

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.

1.1. Description

Capsule is a mobile wardrobe solution powered by computer vision (CV) and machine learning (ML). With Capsule, users can track when and what they wear just by taking casual photos. and get personalized outfit recommendations. At its core, Capsule provides a matching model that generates personalized outfit pairings and is trained with outfits created by real-life stylists working for well-known clothing companies. The model combines learnt outfit patterns with user preferences to generate personalized results. By tracking user’s clothing preferences, not only will we be able to recommend outfits but also show them a report on their wardrobe utilization. Our knowledge of users’ outfit preferences will also enable us to direct clothing retailers to their preferred demographics via advertisements and/or affiliate marketing.

In addition to its core feature, tracking wardrobe usage and providing outfit recommendations, 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 one is going out to a fast food restaurant or going out for drinks with their friends, they can collaboratively decide on outfits to wear through Capsule.

Our vision is to create a mobile application to track and advise users’ wardrobe usage. In order to achieve this vision, extensive use of CV and ML is required as the system will 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.

1.2. High-Level System Architecture & Components of Proposed Solution

Capsule follows two architectural styles: layered architecture and microservices architecture. User Interface, Cloud-Hosted Backend, and Data Storage System are the three layers that form the application. The ML engines for outfit recommendation, cloth labeling, and matching form a microservices architecture inside the backend layer. Figure 1 summarizes the high-level architecture of Capsule in a diagram.

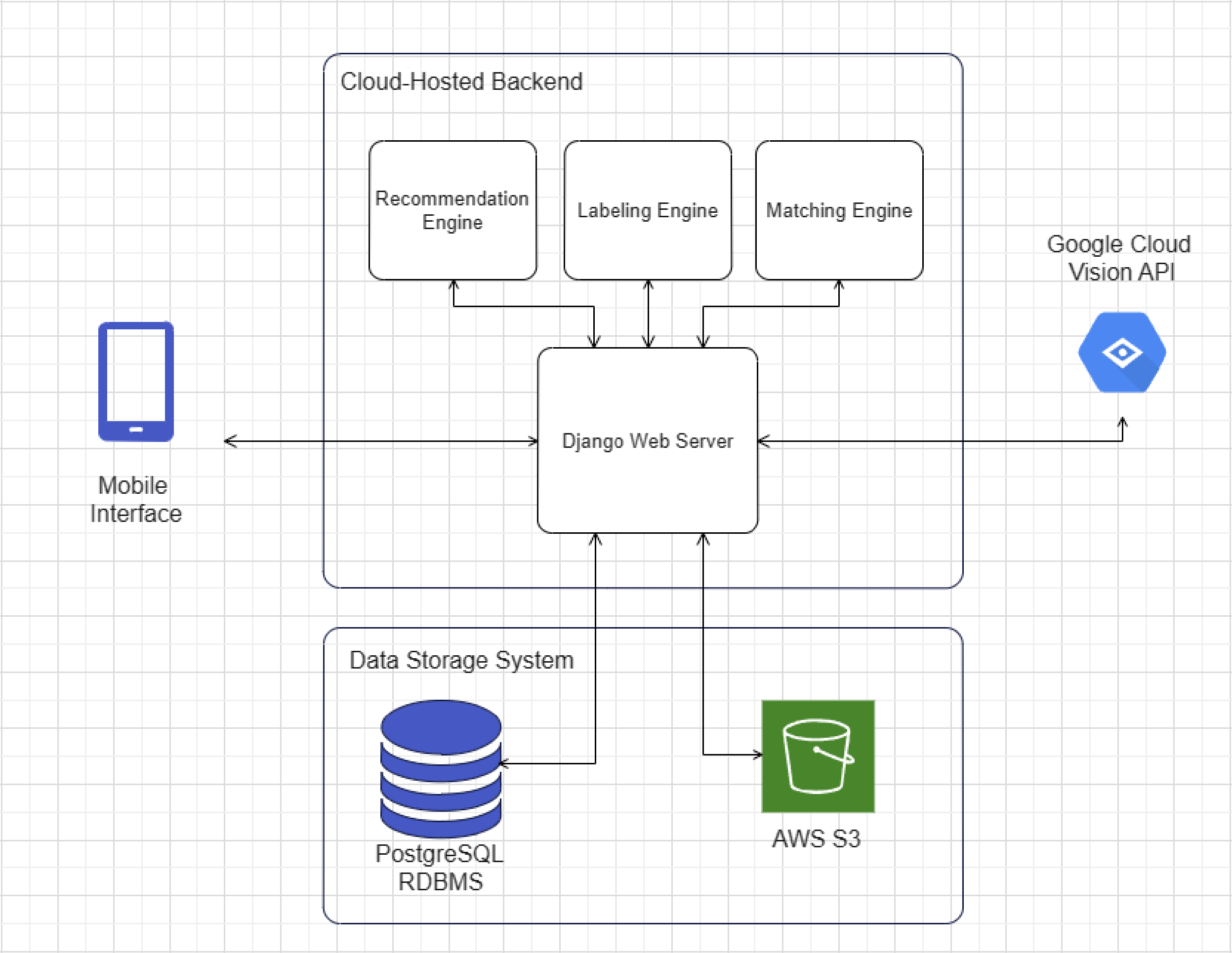

Figure 1: Capsule High-Level System Architecture Diagram

1.2.1. Mobile Interface

Capsule will be a cross-platform mobile app built on React-Native for its ability to build cross-platform applications from a single code base. The mobile interface will be the only interface the user will be interacting with the platform. It is where the user creates their account and manages outfits, pieces of clothing, and events.

1.2.2. Django Web Server

The Django web server will be the main business layer component that interacts with the external engines, APIs, and the database. As far as the Mobile Interface is concerned, the Django web server is the only option for communication. All the logic about Authorization and Authentication, users, friends, and events will be handled here. This will also be used as a proxy to interact with the ML engines.

1.2.3. PostgreSQL Database

As the primary database, a PostgreSQL relational database will be used because of the team's familiarity with the system and due to the economical reasons as it is an open-source system. Unless there is a reason not to store data in a relational database, all data will be stored here.

1.2.4. Labeling Engine

The Labeling Engine is the ML model that we will produce to label pieces of clothing to be used by other engines. These labels are needed for creating semantic data from images that are more easily processable.

1.2.5. Recommendation Engine

The Recommendation Engine will recommend outfits according to user preference and the outfits the user has in stock. This engine uses the Labeling Engine’s results and reccomends clothes and outfits to the end-user. This engine is made of multiple models, which includes the core matching model that matches multiple clothing items to create outfit pairings, as well as the personalization model, which shapes the recommendations based on previous actions and preferences of the user.

1.2.6. Matching Engine

The Matching Engine is for matching a picture of an outfit with one in the user’s wardrobe. This engine uses image recognition to match two pictures of a single clothing.

1.2.7. AWS S3 Bucket

An AWS S3 Bucket will be used for storing and organizing the clothing images uploaded by the users.

1.2.8. Google Vision API

Google Vision API will be used for segmenting and selecting pieces of clothing from the image uploaded by a user. By using Google Vision API, we are relieved from segmenting the pictures ourselves, giving us more time to develop the other technologies.

1.3. Constraints

1.3.1. Economic Constraints

The application should be accessible to the target audience with no initial fees. This would allow the application to grow faster within the target user base. However, necessary partnerships must be made with the fashion industry to cover operational costs. The majority of the cost will be from storing user content and running the machine learning models. As an initiative with no funding, it is crucial for us to keep everything as low-cost as possible.

1.3.2. Ethical Constraints

As we are working with user image data it is crucial for us to keep these data secure and private. We also must be transparent about what data we are collecting and receive permission from the user to collect these data. Also, the algorithms we use shouldn't have any bias or discriminate against any groups.

1.3.3 Political Constraints

The application must respect the corresponding data privacy law within the countries it operates. Initially, the application must follow KVKK in Turkey. For compliance with the GDPR the user must be able to delete their account with all the stored data according to the "Right to be forgotten" principle.

1.4. Professional and Ethical Issues

The application operates on user-uploaded content. Such content includes photos that are used to both create a virtual wardrobe of the user and generate outfit recommendations. For instance, it is essential to filter ineligible photos which may or may not be intended to be uploaded. Such photos would result in ethical and professional issues.

1.4.1 Privacy and Data Security:

Users of the application will upload their sensitive data to the application's system. Therefore, it is essential to keep this data secure during the transport and storage of it.

Users of the application will be informed about the scope and the extent of the collected data. Users will also be informed about the potential processing of their data to enhance the application's performance. Furthermore, users will be notified and required to accept new conditions if they arise.

1.4.2 Algorithmic Bias and Fairness:

The initial machine learning model will be trained on existing datasets. However, the algorithm will change over time to provide better results for users. As a result, certain biases such as age or gender may occur due to the majority in the dataset. Therefore, we will try techniques proposed by machine learning researchers to minimize such biases.

1.4.3 Informed Decision-Making:

Users will be informed about how their selections will be used to improve the system. One such usage is during the generated outfit selection, which will be used to enhance generated outfits in the future.

1.4.4 User Consent for Advertisement Targeting:

Advertising Preferences: When directing clothing retailers to preferred demographics, user consent should be explicitly obtained, and users should have control over the extent to which they receive targeted advertisements. Respect for user preferences and data autonomy is vital.

1.4.5 Collaborative Decision-Making:

Consent and Control: The collaborative feature for setting dress codes should allow users to opt in or out, ensuring that individuals maintain control over their clothing choices. Consent and respect for personal preferences are critical in these collaborative decisions.

1.4.6 Image Processing and Consent:

The application will process user-uploaded images to enhance the overall recommendation performance of the system. Users must be informed about such processing.

1.4.7 Selling Recommendations:

Users must be informed about commissions or fees when they decide to sell clothes that have not been worn in a long time. Furthermore, such selling recommendations must be made carefully to avoid confusion.

1.4.8 Intellectual Property and Copyright:

Users must be informed about the intellectual property rights of their content if they decide to upload. It must be clear who has the rights to user-uploaded content.

1.4.9 User Profiling:

Users should know how their choices affect the promoted content that is shown to them. Promoted content includes brand deals and advertisements.

1.5 Standards

Since Capsule is worked on by a team of 5 people it is important to have standards to keep consistent documentation and a consistent codebase.

1.5.1 Reports

In our reports, we plan on using UML 2.5.1 for our object-oriented design models [3]. These include Class Diagrams, Usecase Diagrams, State Diagrams, Sequence Diagrams, and Activity Diagrams. For the elicitation of requirements, we are planning on using the IEEE 830 standard for documenting our requirements [4]. Finally, we will be citing all our references in the IEEE citation format [5].

1.5.2 Codebase

As with reports, it is important to standardize our codebase for consistency and increased readability. Hence, for our Python code, we will be using the 'Blackened' style applied by Black Formater which is PEP 8 compliant [6]. For our Typescript code, we will be using Google's Typescript style guide [7].

1.5.3 Commit Messages

We will be using GitHub for issue-tracking and repository hosting. Having a consistent commit message convention is important for looking back at these commits. Hence we will be using the following convention for our commit messages:

<Issue Type> CAP-<Issue-Id>: [Commit message starting with an imperative verb (without a suffix)]

1.5.4 Issue Tracker

All changes must be merged to the main branch through a pull-request reviewed by another team member. Adding commits to the main branch directly is forbidden. We are adding this standard for eliminating destructive commits on the main branch.

2. Design Requirements

In this section, we outline the essential criteria and specifications that will guide the development of Capsule.

2.1. Functional Requirements

This subsection defines the core capabilities and functionalities of Capsule. These functional requirements encompass the application's ability to track wardrobe usage, generate outfit recommendations based on user preferences, and facilitate collaborative dress code decisions.

2.1.1. User Profile Creation and Management

- Users must be able to create and manage their profiles, including adding personal information, profile pictures, and wardrobe details.

2.1.2. Event Creation and Management

- Users must be able to create events and specify event details such as name, date, time, and location.
- Event creators can invite friends to events by creating a link to the event.
- Event host can kick participants
- Users can leave event

2.1.3. Event Discussion and Dress Code Setting

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

2.1.4. Event Notifications

- Users should receive notifications for event updates and event discussions.

2.1.5. Privacy Controls

- Users should have the ability to control the visibility of their activities and wardrobe details.
- Users can set privacy preferences for their profile, events, and dress code discussions.

2.1.6. Recommendation and Suggestion Integration

- Social interactions, such as discussing dress codes, should inform the recommendation system for suggesting outfits to users attending events.

2.1.7. Real-time Chat and Messaging

- Users attending the same event should be able to chat and exchange messages in real-time.

2.1.8. Event Reminders

- Users should be able to receive reminders for upcoming events they've scheduled.

2.1.9. Outfit Recommendation

- The system must be able to combine clothes to form personalized outfit recommendations that can be accessed by the user within the app.
- Outfit items, which are clothes, must be primarily from the user's wardrobe, and the algorithm must be able to provide extra outfit item suggestions by using the sponsored items or the items fetched from known clothes retailers' internet sites.
- Either a completely algorithm-generated outfit must be suggested, or the user must be able to enter specific items that they would like to wear at the moment, and the system must be able to provide suggestions that go well with the selected item(s).

2.1.10. Personalization

- Outfit recommendations must be personalized based on the user's behavior and personal information, which may include the user's age, gender, body type, and behaviors learned via machine learning models.
- Outfit recommendations must also be personalized based on the instance for which the recommendation will be made, which may include weather, dress code, and user's preferences regarding what type of clothes to wear at that specific time.
- For behavior-analysis-led personalization, some features must be tracked, which may include the type of clothes worn around the current time.

2.1.11. Built-in Access to Items From Brands

- The system must be able to fetch and parse data from targeted websites by using a custom web crawler.
- Clothing-related data must be automatically classified and labeled to make the fetched content accessible for the recommendation system.
- Fetched clothes must be usable for direct outfit item suggestion to users, and outfit combinations must be used to train the outfit recommendation model.

2.1.12. Image Recognition

- If an item cannot be recognized or is not present in the database, the user must be able to take photos of an item, and it must be automatically labeled by the system before adding it to the database.
- Photos of the user that are either shared in the app or uploaded as mirror-selfie must be processed and matched with items available in the user's wardrobe to track the last-worn information of items, as well as how long the item is used for.

2.1.13. Wardrobe Tracking

- The system must be able to track and estimate outfit items that are available for users to use at any given time.
- The system must add outfit items to the wardrobe as the user takes photos of clothes. Other than taking clothes photos one by one, the system must also be able to detect clothes worn by the user in their photos.

2.1.14. Wardrobe Processing

- The system must be able to make suggestions to make use of the least-worn clothes. The suggestions may include creating outfit combinations which include that specific item or offering to sell the item.
- The system must be able to track and detect when a cloth is worn out too much and suggest similar new clothes from the internet or sponsors.
- The system must also be able to make personalized suggestions for new clothes the user may prefer to buy.

2.1.15. User Analytics

- The system must be able to show weekly and monthly wardrobe reports, which will have a digest of the following information. These reports will be shown to the user like a pop-up event.
  - Most/Least worn clothes
  - Most worn colors
  - View the pie chart of outfit categories in wardrobe
  - New outfits / New clothes added to wardrobe (3 examples most and count of items (weekly))
  - Wardrobe utilization percentage (weekly/monthly)

2.1.15. Building Outfits Manually

- In addition to system-recommended outfits, users must be able to manually create outfits by using the clothes already available in their wardrobes.
- User must build an outfit by selecting a clothing item for each item type.

2.1.16 Outfit Management

- Users should be able to save either manually or automatically created outfits and access them later.
- Users should be able to add outfit properties optionally, such as names and tags.

2.1.17 Signup and Login

- Users should be able to authenticate via Google, Apple, or email and password.
- Users should be able to verify their email address by entering a verification code sent to their email address by the system.

- During the signup process, users should be able to enter information such as full name, date of birth, and username, upload profile picture, and select their style preference, which includes masculine, neutral, or feminine, to shape their profile.

2.1.18 Reset Password

- In case of forgetting their password, users should be able to reset their password by entering their email address and using the link mailed to them to set a new password.

2.1.19 Selecting Outfits For a Day

- Select “Today” or a day from the calendar view
- The user can select an outfit
    - Select a saved outfit from the built outfits in the wardrobe
    - Build an outfit for the day
    - Take a mirror selfie to match worn outfits to the ones in the system
- User can view selected outfits from the calendar view and be reminded of their daily outfit

2.1.20 Viewing/Managing Wardrobe

- User selects between outfits and clothes
- Filtering clothes by color, favorited clothes and type (top, overwear, bottom, full-body, shoes)
- Adding clothing or outfit to favorites
- See outfit/clothing properties
    - ML-recommended
        - Colors
        - Tags
        - Seasons
        - Texture
    - Usage counts
    - Last worn date
    - Favorited or not
    - Create an outfit with this clothing item
- Delete 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
- Add outfit/clothing to wardrobe

2.1.21 View Advertised Outfits

- User sees advertised pieces of clothing once in a certain amount for outfit generation.
- User can also trigger shopping recommendation from a single clothing piece
    - App shows the user recommended shopping pieces

2.2. 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.

2.2.1. 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.

2.2.2. 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%.

2.2.3. 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.

2.2.4. 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.

2.2.5. 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. Feasibility Discussions

In this section, we explore the practicality and viability of implementing Capsule. This section assesses the technical, economic, and operational feasibility of the project, considering the resources, infrastructure, and budgetary constraints.

3.1. Market & Competitive Analysis

This subsection delves into examining market dynamics and the competitive landscape for Capsule.

3.1.1. Target Audience

The target audience demographics are ages 15-55. The gender distribution is mainly weighted towards women, even though the main target is fashion-conscious individuals from both genders. The target audience is selected with respect to the surveys. According to the surveys, a larger percentage of women opt for online shopping than men, and the ages 15-55 are concerned about what to wear and have a desire to buy clothes often [8]. Our target demographic has varying income levels but a significant presence of middle to high-income users. The target audiences' consumer behavior may vary.

On the one hand, there are regular online shoppers for items of clothing that are comfortable with various online payment methods and online shopping applications. They are inclined to go online shopping and explore a wide range of clothes, and thus, suitable for our "clothing recommendation from brands" feature. On the other hand, there are followers of the minimalism trend that aims to create a sustainable capsule wardrobe. Our outfit recommendation feature, followed by the outfit tracker that tells the user which clothes are less worn, is perfect for this audience.

In the near future, we plan to publish our application to the Turkish market. According to TÜİK, there are approximately 85 million Turkish people [9]. Our target audience (women of 15-55 ages) is nearly 32% of Turkey. If we assume 80% of them are active smartphone users and only 20% are interested in fashion, there are approximately 5.44 million potential users. Assuming our application only captures 5% of the users, there are 272,000 potential users.

3.1.2. Competitive Landscape

Currently, there are a few outfit recommender mobile applications, but not a lot where users can create outfits manually by taking photos of clothing items. Some examples are Style DNA, Vestai, Skap, and Acloset - AI Fashion Assistant. Although the application users can create new outfits, we believe it does not eliminate the main problem of "recommending what to wear to the user." Hence, we choose our main competitors as outfit recommender mobile applications. Table 1 contains a comparison between Capsule and its possible competitors on the market.

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

There are currently three main outfit recommender mobile applications: Whering, Pronti, and Pureple. One possible competitor to Capsule may be Wearing, which also has the outfit recommendation feature in their application. However, their recommendation system seems broken and utterly randomized. According to an article by Rule, "... the randomized wardrobe shuffle is as good as pulling things out of your wardrobe with your eyes closed." [10]. Another possible competitor may be Pronti, another smart wardrobe mobile app that had much hype on TikTok and, thus, lots of downloads. However, their app currently seems broken since both their image processing and outfit recommendation features do not work. Pureple didn't meet the mark when compared to other smart wardrobe solutions we've reviewed. They opted for a subscription model, which, in our view, may not offer good value for users. Using the app without a subscription can be a bit frustrating due to the frequent 5 to 30-second advertisements that pop up. Also, not any smart wardrobe mobile app was widely introduced to the Turkish market previously. Hence, there is a huge market gap in the Turkish market, which we believe we can fill. Also, since there is no proper outfit recommendation system that our target audience demands, we believe there will be a demand for our product on the market.

3.2. Academic Analysis

There are various academic sources that can be used during the development of Capsule. Topics of computer vision and machine learning must be thoroughly understood to develop the crucial components of the system. These components' functionalities include clothing detection and extraction from uploaded images, labeling of the detected clothing pieces, matching of different clothing items to generate outfits, and learning users' clothing preferences based on worn clothes. The following is a selection of the academic papers scanned during the first iteration of the literature review, and the list is expected to evolve as the development process continues.

- **Hierarchical Fashion Graph Network for Personalized Outfit Recommendation** [11]
  This paper introduces a framework called Hierarchical Fashion Graph Network (HFGN) that combines fashion compatibility modeling and personalized outfit recommendation. HFGN creates a hierarchical structure to capture relationships among users, items, and outfits. It employs graph neural network techniques to aggregate item information into outfit representations and enhance user representations using their historical outfits [11].
- **Personalized Outfit Recommendation With Learnable Anchors** [12]
  This paper presents a novel approach for personalized outfit recommendation that addresses diverse user preferences. It employs a stacked self-attention mechanism to capture complex item interactions within outfits and creates a condensed representation for each outfit. Users are characterized using a set of learnable latent vectors, or "anchors," that represent their preferred outfits. Additionally, general anchors are learned to capture shared preferences among all users [12].
- **FashionNet: Personalized Outfit Recommendation with Deep Neural Network** [13]
  This paper introduces FashionNet, a system designed for recommending outfits that match users' individual fashion preferences. It recommends sets of coordinated clothing items such as shirts, skirts, and shoes. It comprises two main components: a feature network for extracting item features using deep convolutional networks and a matching network that computes the compatibility of items using a multi-layer fully-connected network. To achieve personalized recommendations, the paper proposes a two-stage training strategy, fine-tuning a general compatibility model to incorporate personal preferences [13].
- **OutfitNet: Fashion Outfit Recommendation with Attention-Based Multiple Instance Learning** [14]
  This paper proposes OutfitNet, a two-stage framework. In the first stage, a Fashion Item Relevancy network (FIR) learns item compatibility and generates relevancy embeddings. The second stage, an Outfit Preference network (OP), captures users' fashion preferences based on visual data. OutfitNet processes multiple fashion items within an outfit, modeling item compatibility, user preferences for each item, and user attention to different outfit components using an attention mechanism [14].
- **Personalized fashion outfit generation with user coordination preference learning** [15]

This introduces the Template-guided Outfit Generation (TOG) framework, which combines user coordination preferences, user-item interactions, and outfit compatibility modeling. By incorporating coordination knowledge in fashion, the paper defines category combinations as templates and models the relationship between users and these templates. This approach enables the simultaneous modeling of personal preferences and outfit generation [15].

- **Example-based object detection in images by components** [16]
  This paper introduces a general framework for object detection in static images based on components, demonstrated through the example of locating people in cluttered scenes. The system employs four separate example-based detectors to identify different body components (head, legs, left arm, and right arm) and then combines their results to classify a pattern as a "person" or "nonperson." [16]
- **Object detection using image reconstruction with PCA** [17]
  This paper presents an object detection system for pedestrian detection in still images without prior knowledge about the image. The system utilizes a two-stage approach, where an initial classifier examines locations at different scales, followed by a heuristic-based step to eliminate false detections. The classifier employs Principal Component Analysis (PCA) separately for positive and negative examples. When classifying a new pattern, it projects it into both sets of PCs and assigns it to the class with the smallest reconstruction error. The system can detect frontal and rear views of pedestrians and often identifies side views despite not being specifically trained for them [17].

3.3. Risks

Since we will work with massive data about items of clothing and user preferences, there can be some legal and compliance risks. We need to be aware of laws and regulations about data analysis, such as KVKK and GDPR. There are also financial risks, which is one of the main problems since we currently lack funding. For instance, our development and marketing costs may exceed initial estimates.

There is the risk of user acquisition and user behavior risks in general. Acquiring users through advertising and marketing can be costly, and we may fail to target the right audience with our ads. Hence, we should have a solid and well-thought-out marketing plan that balances the cost with effectiveness. Also, users may abandon our app after initial use or may not use the application regularly. Hence, a notification system that may catch users' attention but does not feel forced is currently planned to avoid the risk.

There are also technical risks. We may not implement the optimal recommendation algorithm that takes all the parameters into account like the weather, user preferences, body type, and occasion. To avoid these risks, we have to come up with the right action plans. Also, fashion trends are constantly changing. As Capsule, our application must not rely heavily on current fashion trends, and we must keep up with the pace of change, maintaining our outfit recommendation algorithms occasionally.

Other than the recommendation-related machine learning models, computer vision models also lead to risks caused by the possibility of generating low-accuracy outputs. In addition to the difficulty of extracting an object from an image and automatically labeling the object's features, the models need to be efficient, effective, and highly accurate so that the outputs should be usable even for low-quality, complex images. The initial research about image segmentation suggests the use of various networks, such as Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN) [18]. While promising solutions seem to exist that currently overcome image processing issues, coming up with models that satisfy the performance and accuracy constraints forms risks.

3.4. Business Plan

Since the target audience is mainly Turkish people with various incomes, we plan to make Capsule's features free. We plan on gaining profit primarily from affiliate marketing and possible sponsors. We plan to combine our "tracking and reporting the less worn clothes" feature with a sponsored application that allows users to sell their less worn clothes, such as Dolap [19] or Letgo [20], with just a button click. This will benefit the users since they will clear space from their wardrobe, and will benefit the sponsors since their mobile application will be promoted and selected by the users.

With our "recommend me a clothing piece" feature, we plan to collaborate with clothing brands that will also be our sponsors. While recommending a clothing piece, the user can select to get recommendations from the sponsored clothing brand's clothes. Also, on the outfit recommendation feature, one in ten outfits recommended to the user will include a sponsored brand's clothing piece. When clicked on the sponsored clothing item, it will direct the user to the brand's store page. Hence, they will promote their clothes to a target audience matching ours and can enhance their sales.

Finally, although both premium and default users have access to all features, we plan on adding a paid subscription system to gain profit. Premium users, users who subscribed to Capsule, will have an unlimited closet size, whilst default users will have a closet size of 50 clothes. Also, premium users will have the chance to close recommendation of sponsored items that comes across the users on the outfit recommendation feature.

Acronyms and Abbreviations

KVKK: Kişisel Verileri Koruma Kurumu
GDPR: The EU General Data Protection Regulation
IEEE: The Institute of Electrical and Electronics Engineers
PEP8: Python Enhancement Proposal 8

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].

[3] “About the unified modeling language specification version 2.5.1,” OMG Standards Development Organization, https://www.omg.org/spec/UML/2.5.1/About-UML/ (accessed Nov. 16, 2023).

[4] 830-1998 - IEEE recommended practice for software requirements, https://ieeexplore.ieee.org/document/720574 (accessed Nov. 16, 2023).

[5] “Guides: Citation styles: APA, MLA, Chicago, Turabian, IEEE: IEEE style,” IEEE Style - Citation Styles: APA, MLA, Chicago, Turabian, IEEE - Guides at University of Pittsburgh, https://pitt.libguides.com/citationhelp/ieee (accessed Nov. 16, 2023).

[6] Psf, “PSF/Black: The uncompromising Python code formatter,” GitHub, https://github.com/psf/black (accessed Nov. 16, 2023).

[7] Google typescript style guide, https://google.github.io/styleguide/tsguide.html (accessed Nov. 16, 2023).

[8] A. Brezinskaya et al., “Men vs. women. who runs the online shopping world?,” *BelVG Blog*, https://belvg.com/blog/men-vs-women-who-runs-the-online-shopping-world.html. [Accessed: Oct. 29, 2023].

[9] “Dünya Nüfus Günü, 2023,” *TÜİK*, Jul. 6, 2023. [Online]. Available: https://data.tuik.gov.tr/Bulten/Index?p=Dunya-Nufus-Gunu-2023-49688. [Accessed: Oct. 15, 2023].

[10] D. Rule, “The Best Wardrobe Apps,”*INDYX*, Jan. 2, 2023. [Online]. Available: https://www.myindyx.com/blog/the-best-wardrobe-apps. [Accessed: Oct. 15, 2023].

[11] X. Li, X. Wang, X. He, L. Chen, J. Xiao, and T.-S. Chua, “Hierarchical Fashion Graph Network for Personalized Outfit Recommendation,” in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, in SIGIR ’20. New York, NY, USA: Association for Computing Machinery, Jul. 2020, pp. 159–168. doi: 10.1145/3397271.3401080. [Accessed: Oct. 28, 2023]

[12] Z. Lu, Y. Hu, Y. Chen, B. Zeng, “Personalized Outfit Recommendation with Learnable Anchors,” *IEEE Xplore*, 2021. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/papers/Lu_Personalized_Outfit_Recommendation_With_Learnable_Anchors_CVPR_2021_paper.pdf. [Accessed: Oct. 15, 2023].

[13] T. He and Y. Hu, “FashionNet: Personalized Outfit Recommendation with Deep Neural Network.” arXiv, Oct. 04, 2018. doi: 10.48550/arXiv.1810.02443. [Accessed: Oct. 28, 2023].

[14] Y. Lin, M. Moosaei, and H. Yang, “OutfitNet: Fashion Outfit Recommendation with Attention-Based Multiple Instance Learning,” in Proceedings of The Web Conference 2020, in WWW ’20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 77–87. doi: 10.1145/3366423.3380096.

[15] Y. Ding, P. Y. Mok, Y. Ma, and Y. Bin, “Personalized fashion outfit generation with user coordination preference learning,” Information Processing & Management, vol. 60, no. 5, p. 103434, Sep. 2023, doi: 10.1016/j.ipm.2023.103434.

[16] A. Mohan, C. Papageorgiou, and T. Poggio, “Example-based object detection in images by components,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 4, pp. 349–361, Apr. 2001, doi: 10.1109/34.917571.

[17] L. Malagón-Borja and O. Fuentes, “Object detection using image reconstruction with PCA,” Image and Vision Computing, vol. 27, no. 1, pp. 2–9, Jan. 2009, doi: 10.1016/j.imavis.2007.03.004.

[18] A. Abdullayeva, “Clothes and Color Extraction with Generative Adversarial Network,” Towards Data Science, Sep. 11, 2019. [Online]. Available: https://towardsdatascience.com/clothes-and-color-extraction-with-generative-adversarial-network-80ba117e17e6. [Accessed: Oct. 15, 2023].

[19] *Dolap*, 2023. [Online]. Available: https://dolap.com/. [Accessed: Oct. 15, 2023].

[20] *Letgo*, 2023. [Online]. Available: https://www.letgo.com/. [Accessed: Oct. 15, 2023].