Bilkent University

Department of Computer Engineering

CS 491: Senior Design Project

Rhapso

Analysis Report

Group Members:

Işık Özsoy	21703160	isik.ozsoy@ug.bilkent.edu.tr
Defne Betül Çiftci	21802635	betul.ciftci@ug.bilkent.edu.tr
Şebnem Uslu	21802068	sebnem.uslu@ug.bilkent.edu.tr
Melike Fatma Aydoğan	21704043	fatma.aydogan@ug.bilkent.edu.tr
Burak Yetiştiren	21802608	burak.yetistiren@ug.bilkent.edu.tr

Supervisor : *Prof. Dr.* Halil Altay Güvenir

Jury Members : *Asst. Prof. Dr.* Shervin R. Arashloo and *Asst. Prof. Dr.* Hamdi Dibeklioğlu

Innovation Expert: Aras Bilgen

Website URL : https://ivorymask.github.io/cs491-2.github.io/

Analysis Report

November 8, 2021

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.

1. Introduction

In today's world, it is a common practice that many traditional daily activities are supported with tools implemented in engineering domains. Thanks to a wide range of available software, users can perform daily activities more comfortably. Some examples include: *Spotify* [1], by which the users can basically carry around all the music that they would like to listen to throughout the day; this activity would otherwise be more complicated, having in mind that the users would had to carry the music records and the hardware to listen to the music with them, which would be extremely inefficient. *Google Maps* [2], by which the users can navigate easily in the streets of even the most complicated and congested cities; again in the absence of this software, the users would have to use some traditional methods like some physical maps, which are large to carry and do not have any interactive component like Google Maps like telling the traffic, offering alternative routes, suggesting different paths for different means of transportation, etc. There are more examples on a similar basis that we could grow our list on, but we believe that these examples suffice in terms of clarifying our justification. Among explained examples, one other important common point is that all of the examples given include virtualization, that the users can install these software as mobile applications in their mobile devices, and use them wherever they want as long as they have their mobile devices with them.

Having made the previous points, our project *Rhapso*, which is named after a goddess with the meaning "The Stitcher" [3], will be based on easing a particular daily activity, while using the virtualization aspect of mobile applications. As explained, our project will be a mobile app. With this app, we want to implement and provide the functionality of creating a virtual wardrobe using the photos of the clothes of the user. In comparison to a regular wardrobe, our wardrobe will be interactive which will give shopping suggestions in terms of the already existent items in itself. We want to pay attention to the sustainability of the environment, such that with this application we aim to reduce overall clothing consumption. Lastly, in our application, the users will have the opportunity to try on the clothes virtually, so that on occasions that the time is constrained, our application will provide practicality, which we believe will be essential.

Other than the virtualization aspect of the application, one other issue that we are trying to solve with *Rhapso* is the unnecessary consumption of clothing pieces. The statistics show that people bought 60% more clothing garments in 2014 than in 200, while another statistics to note here is that fashion production makes up 10% of humanity's carbon emissions [4]. Exceeding fashion consumption dries up water sources and causes pollution of rivers and streams [4]. These show that consumption of clothing is an ever growing environmental problem. A study on shopping behavior from researchers in University of Michigan state that the participants involved in the study described wanting deliberation or reflection before buying a clothing by asking the shopper what they would use this new clothing for and if they truly needed it [5]. Another way the participants wanted an application to encourage deliberation was through questioning their possessions, reporting if they have similar ones and how many they have of it [5].

Individual activities such as thrifting aim to offer a solution to the problem of pollution of clothing garments, but they do not have a wide impact as they are and they still do not affect customer behavior. One aim of *Rhapso* is to offer another solution to the present over-consumption of clothes. The application aims to do so by:

1. Comparing a new clothing garment to existent ones saved in the closet,
2. Finding similar clothes from online thrift stores to the new clothing such that it would encourage purchase of secondhand clothing instead of a new one,
3. Making the user "try on" the new clothing garment with the clothes they own so that they can see if the new clothes fit in with their closet,

4. Recommending outfits using the clothes inside their closet so that the user can see if they can make combinations of the new clothing with the existing ones.

By doing these tasks, our aim is to have reduced consumption of clothes for customers over time.

In the technical part of the argument, we want to make use of computer vision algorithms. The algorithms will mainly be used in the detection of the cloth from the photo, and while comparing clothes with each other in terms of their similarity. On the other hand, in the part of enabling users to virtually try on clothes, again we will be making use of computer vision algorithms, which for this part will be embraced in *GANs (Generative Adversarial Networks)*.

In this report, we intend to provide a high-level of our designed system. Firstly, similar applications, the differences and missing features will be discussed. Afterwards, a description for the proposed system will be given. We are going to explain Rhapso and its features in detail. Then, functional and non-functional requirements will be explained. Furthermore, we are going to discuss our system models. Scenarios will be generated based on the use-cases, the diagrams will be also provided. Addition to use case diagrams, object and class model and dynamic models such as activity and sequence diagrams will be provided. Following this, there will be screen mock-ups and navigation paths. At the end of the report, a discussion will be provided on the social issues.

2. Current System

There are two similar applications that could be considered for our discussion as current systems. Namely, the applications are TryNDBuy, and Your Closet [6] [7]. First, we want to discuss TryNDBuy. In TryNDBuy, the users can try clothes, which they can choose from a catalog on a 3D model provided by the application. Users also can buy clothes using the application. The main aim of this application is to encourage its users to buy clothing items using the application. The second application we consider is Your Closet. The main ambition in this application is to store the clothing items of a user. The users of the software can add clothing items to their accounts by providing the pictures of the items to the application. Two other applications with the same goal of reducing the consumption of new clothes are Dolap and Gardrops, two of the most popular online secondhand clothing shopping applications in Turkey. These applications are used for cataloging secondhand clothes that users put out and have the purpose of providing a platform for popularizing purchase of secondhand clothing, therefore sharing the same goal of reduction of consumption with Rhapso while not providing any of the benefits of the proposed system.

3. Proposed System

While our proposed system carries some features from both software mentioned in the section 2. Current System, we argue that our application Rhapso will differ from the current system by checking and scoring the similarity between clothing items, giving both suggestions for clothing combinations, and sustainable shopping, and keeping a detailed track of the closet.

3.1. Overview

Rhapso is a tool that uses image processing and machine learning to create a virtual wardrobe, to match outfit as suggestions, and raise awareness to users for existing similar clothes or secondhand options of similar clothes in websites such as Dolap and Gardrop to support sustainability.

The tool is innovative in the sense that no other application that offers a virtual closet can solve the problem of consumption, sustainability, and creation of a virtual fitting room. Rhapso can fit the virtual cloth on the 2D model of the user's choice, combine them to make outfit combinations, and allow users to use a 2D fitting room. Currently, the 2D fitting room is in the process of being made and not successfully managed; whereas, the solution for sustainability and consumption is a new area that we discuss. Therefore, Rhapso aims to solve the long-going problems that are non-managed, or in the process of being managed.

Advanced techniques and technologies will be used for Rhapso. The software will include image processing to scan the photo of the object and create a virtual version of it, machine learning to match clothes in the virtual closet so that outfit suggestions can be made, and a database to keep the virtual closet and outfit combinations permanently accessible. These computer science technologies will be able to make Rhapso for us and show that problems of the non-technological part of life can be solved using high technologies such as GaN.

Rhapso aims for virtualization as discussed before. Users will be able to access their virtual wardrobe from anywhere and add new clothes to their closet as they shop. The software will be implemented as an Android application and future enhancements for iOS users will be discussed.

Tools Features	**Rhapso**	**TryNDBuy**	**Your Closet**	**Dolap**	**Gardrops**
Virtual Closet	✔	✘	✔	✘	✘
Virtual Try-On with 2D Model	✔	✔	✘	✘	✘
Similarity Comparison	✔	✘	✘	✘	✘
Aid in Shopping	✔	✔	✘	✔	✔
Thrifting Suggestions Based on Similarity	✔	✘	✘	✘	✘
Outfit Recommendations	✔	✘	✔	✘	✘
Goal of Reducing Consumption	✔	✘	✘	✔	✔

Tab. 1: Comparison of Rhapso and other applications according to features

3.2. Functional Requirements

3.2.1. For the system

- System should obtain authorization from the user to use the resources of the phone, i.e. camera, etc.
- System should make it possible to provide camera input with an option.
- System should provide an option for registration with a new account, making the user provide a username, an email and a password.
- System should provide an option to change username, email, and password.
- System should provide an option for loginning to the app with a pre-existing account.
- When a new photo of a clothing is provided, the system should distinguish the clothing itself from the background.
- The system should provide an option for the user to add their own outfits to the application by combining the existing clothes in the closet.
- When a new photo of a clothing is provided and when the system has successfully distinguished it from the background, the system should provide options to add the clothing to the wish list, add the clothing to the user’s virtual closet, or compare the clothing with the rest of the clothes in the closet.
- When the user has clicked on a clothing item inside their “closet” or their wish list, the system should provide an option to try on the clothing over the 2D model in the application. The model will be a colored 2D model that will change its shape - i.e. its width, height - based on how the user customizes it.
- When the user has clicked on a clothing item inside their “closet” or their wish list, the system should provide an option to show creations of combinations of the selected clothing with existing clothes inside the closet. The “outfit creator” that creates custom combinations of clothes in the closet should do these combinations to ascribe to certain aesthetical rules, such as how the colors of the clothes the outfit generator combined should match in terms of color theory (considering hue and value of the color of the clothing) and in terms of patterns.
- The system should provide an option to keep or discard the outfit recommendations the outfit creator recommended. If they keep the outfit recommendation, it will be added to the list of their own outfits. If they choose to discard the recommendation, the system will show another recommendation of combinations.
- The system should learn from the user’s choice of outfits, from the clothes they combined as submitted outfits, and from the recommendations of outfits the user decided to keep and discard.

3.2.2. For the users

- Users should be able to accept or reject the permission for the use of the camera. However, as the acceptance of such permissions is essential for the application to work, primary features of the app will be unavailable in the situation of rejection. The rejection will not be taken permanently and the application will continue to ask for permission when the camera feature is tried to be used.
- Users should be able to give photo input to the application through the phone camera.
- Users should be able to register to the application with a username, an email and a password.
- Users should be able to add clothes to and remove clothes from their “closet” or their “wish list” stored in the database.
- Users should be able to get recommendations of new clothes they want to buy from online alternatives of secondhand clothings.
- Users should be able to retrieve their previous information stored in the application later and with different applications if they choose to store their data with a registered account.

- When the user takes the photo of a new clothing piece using the camera feature of the application, the user should be able to see possible combinations of this new clothing piece with the clothes already present in their “closet”.
- When the user takes the photo of a new clothing piece using the camera feature of the application, the user should be able to see a comparison score of the new clothes with the ones already present in their “closet”.
- Users should be able to see possible outfit combinations of clothings in their “Closet” or “Wish List” with the clothes already present in their “closet”.
- Users should be able to virtually try the clothes present in their “closet” as well as a new dress they want to buy by trying them on a body model provided by the application, provided they took photos of each piece of clothing beforehand with the application camera.
- Users should be able to adjust the body measurements of the body provided by the application.
- Users should be able to add new clothing to their “wish list” in the application.
- Users should be able to keep or discard the outfit recommendations created by the “outfit creator” as described in 3.2.1.

3.3. Nonfunctional Requirements

3.3.1. Security

The application must be secure enough with passwords and usernames so that no other outsider could access another account’s information directly. The application must ensure that the camera inputs given to the application will not be accessed by a 3rd party application. The application should ask the user permission for enabling access to camera and photo gallery before the first use of such features. Such features should only be accessible by the application only when the application is running and these features are called by the application itself.

3.3.2. Usability

The application must be supported by Android and supported by different versions of Android phones. The system should have an interface that is easily understandable by users. Its interface design should be usable with different sizes of mobile phones.

3.3.3. Performance

The system needs to access the database for a very large number of users (see 3.3.5. Scalability), for this reason we want the application to perform very fast. The application must be fast enough with its processing of clothing images such that its segmentation and the feature of comparing the newly acquired clothing piece with the ones already saved in the database must take no longer than 1 second. When the user wants to see recommendations of other clothes, it should take no longer than 1 second for the application to search through secondhand clothing websites, otherwise known as online thrift stores. As for the virtual clothing trying-on aspect of the application, the application must take no longer than 1 second to add each piece of clothing to the model the clothing piece is added onto.

3.3.4. Reliability

The application must detect clothes, particularly their structure and their textures if they have anything printed on them, correctly and render new images of the clothes, adjusted to the body model that we will put the clothes on to, as close to real life as possible. The application should also give correct approximations of the closeness of the new clothing piece that is being scanned when it is compared to the clothes already present in the user's "closet", i.e. it should not give too high of a closeness score to two clothes that are similar in no way. These are factors that will determine the user's satisfaction with the application, hence the necessity for reliability of them.

Additionally, the system should notify the user as so in the case of a bad or unintelligible clothing photo, e.g. the photo has no clothing in it, the photos of clothes are not taken from the front view which makes the photo a bad candidate for virtual try-on aspect of it, etc. The system should have reliable failure management. It should not have any errors or exceptions upon release. Stack overruns, memory management problems, and such non-deterministic application failures should be handled by causing the app to rollback, and thus force a restart. The application should have reliable input/output management so that no failures in direct response to a data input or a user input occurs.

Hosting the app' APK on Google Play offers some reliability benefits [8] [9], including:

- Hosting the app on high-uptime Google servers; through this, dealing with server failure problems is cheaper (as we will be leaving the issue mostly to Google services and will not need to buy better hardware for the application)
- Compatibility testing for devices will be automated and done for each mobile device for the application
- Download times are faster than apps hosted in external servers
- Data consumption is reduced when the app is hosted on Google servers

3.3.5. Scalability

The application should have a maximum of 1 second response time, with the general aimed response time being 45 milliseconds. The application should be able to handle 200 requests per second (12,000 per minute) as the count of requests around this value or more would result in a heavy load for smaller servers, and we aim to handle heavy load situations as well.

3.3.6. Accessibility

The application should be available and freely downloadable from the Play Store hosted by Google for easy access for Android phone users.

3.3.7. Extensibility

The application should be easy to manage for future extensions. Although the application will initially be tailored for Android phones, its interface design should be extensible for different mobile OS systems such as iOS. The development environment should allow for new features to be added with ease if the need occurs.

3.4. Pseudo Requirements

3.4.1. Version Control

- Version control will be done by using Git and Github.

3.4.2. Implementation Technologies

- Implementation language for the training of the model for virtual try on aspect of the application will be Python for access to a wide range of frameworks making use of GAN, such as TorchGAN, TF-GAN, etc. This is to write more comprehensible and easily changeable codes for training and evaluation of GAN models we will produce.
- Implementation language for the training of the model used for outfit combinations will be Python, again for its availability of Deep Learning as well as complex mathematical query libraries (e.g. Numpy).
- The application will be designed to be compatible with Android phones.
- Implementation language for Rhapso will be Java. As such, object-oriented design will be our priority.
- Google Research's Google Colaboratory service will be used for training and analysis of the algorithms used for training the machine learning models. More specifically, its GPU machines will be used for accelerating the process of training and testing for the models.
- NVIDIA GPUs will also be used for acceleration of training and testing algorithms for machine learning models. These will be provided individually with computers of group members.

3.5. System Models

3.5.1. Scenarios

Scenario 1

Use Case Name	*login*
Participating Actors	User
Entry Conditions	· There is no user already logged in the application.
Exit Conditions	· User is successfully logged in, or the login failed.
Main Flow of Events	User: 1. Clicks on *login* button. 2. Enters username. 3. Enters password. 4. Clicks on *submit* button.

Scenario 2

Use Case Name	*register*
Participating Actors	User
Entry Conditions	· There is no active account in the application.
Exit Conditions	· The user is registered to the application.
Main Flow of Events	User: 1. Clicks on *register* button. 2. Enters the required fields for identification. 3. Clicks on *submit* button.

Scenario 3

Use Case Name	*giveCameraPermission*
Participating Actors	User
Entry Conditions	· User has not yet given permission to the application for the access to the camera.
Exit Conditions	· The user either gives permission to the application to use the camera, or denies.
Main Flow of Events	User: 1. Clicks on the *allow* button on the notification asking for a permission.

Scenario 4

Use Case Name	*takePhoto*
Participating Actors	User
Entry Conditions	· User is on the *main menu*, *take photo button* is clickable.
Exit Conditions	· User takes a photo successfully.

Main Flow of Events	User: 1. Clicks on the button specified to take a photo by their operating system.

Scenario 5

Use Case Name	*seeCombinations*
Participating Actors	User
Entry Conditions	· User has already taken a photo of a clothing item.
Exit Conditions	· User either chooses a combination or decides to go back from the *combinations menu*.
Main Flow of Events	User: 1. Clicks on the *see combinations button* after s/he has taken a photo of the clothing item. 2. Chooses a combination or decides to exit the menu.

Scenario 6

Use Case Name	*addToWishlist*
Participating Actors	User
Entry Conditions	· User has already taken a photo of a clothing item.
Exit Conditions	· User either adds the item to her/his wishlist or decides to go back from *add to wishlist panel*.
Main Flow of Events	User: 1. Clicks on the *wishlist icon*, which prompts *add to wishlist panel*. 2. Adds the item to the wishlist or goes back to the previous menu.

Scenario 7

Use Case Name	*getSecondhandSuggestions*
Participating Actors	User
Entry Conditions	· User has already taken a photo of a clothing item.
Exit Conditions	· User either clicks on the one of the suggestions or goes back to the previous menu.
Main Flow of Events	User: 1. Clicks on the *get secondhand suggestions* button. 2. Either chooses one of the suggestions by clicking on the *go! button*, which will redirect to the shopping site of that item or goes back to the previous menu.

Scenario 8

Use Case Name	*uploadPhoto*
Participating Actors	User
Entry Conditions	· User has already taken a photo of a clothing item.
Exit Conditions	· The photo uploads successfully.
Main Flow of Events	User: 1. Clicks on the *upload button*.

Scenario 9

Use Case Name	*seeCloset*
Participating Actors	User

Entry Conditions	· User is on the *main menu*.
Exit Conditions	· User clicks on the *close closet icon*.
Main Flow of Events	User: 1. Clicks on the *see closet button* on the main menu.

Scenario 10

Use Case Name	*addClothingItem*
Participating Actors	User
Entry Conditions	· User has already taken a photo of a clothing item.
Exit Conditions	· User successfully adds a clothing item to the closet.
Main Flow of Events	User: 1. Clicks on the add to closet button, which she can see while she is viewing the photo of the clothing item.

Scenario 11

Use Case Name	*removeClothingItem*
Participating Actors	User
Entry Conditions	· User is already viewing her closet.
Exit Conditions	· User successfully removes a clothing item from the closet, or s/he chooses to cancel the removing operation.

Main Flow of Events	User: 1. Clicks on an item s/he wants to remove from the closet. 2. She sees *remove item* and *try on buttons*. 3. She clicks on the *remove item button*. 4. She sees the remove clothing item menu with *cancel* and *remove buttons*. 5. She either removes or keeps the item.

Scenario 12

Use Case Name	*seeWishlist*
Participating Actors	User
Entry Conditions	· User is on the *main menu*.
Exit Conditions	· User clicks on the *close wishlist icon*.
Main Flow of Events	User: 1. Clicks on the *see wishlist button* on the main menu.

Scenario 13

Use Case Name	*tryCloth*
Participating Actors	User
Entry Conditions	· User is already viewing her closet.
Exit Conditions	· User clicks on the *try on button*.

Main Flow of Events	User: 1. Clicks on an item she wants to try on from the closet. 2. She sees *remove item* and *try on buttons*. 3. She clicks on the *try on button*.

Scenario 14

Use Case Name	*adjustBodyMeasures*
Participating Actors	User
Entry Conditions	· User clicks on the *adjust body measures button* in the settings of the application.
Exit Conditions	· User clicks the *save button* in the *adjust body measures menu*.
Main Flow of Events	User: 1. User enters her height. 2. User enters her weight. 3. User saves the information by clicking the *save button*.

Scenario 15

Use Case Name	*authorizeUser*
Participating Actors	System
Entry Conditions	· User has submitted the login information to the system.

Exit Conditions	· The information specified matches the entry in the database and the user is authorized, or the information given was incorrect.
Main Flow of Events	System: 1. Receives the information. 2. Checks if the information matches a record in the database 3. Either logs the user in or prompts the user for a correction.

Scenario 16

Use Case Name	*registerUser*
Participating Actors	System
Entry Conditions	· User has submitted the registration information to the system.
Exit Conditions	· The information entered is compliant with the regulations, and the user is registered; or the information is not compliant with the regulations.
Main Flow of Events	System: 1. Receives the information. 2. Checks if the given information is compliant with the regulations. 3. If the information is compliant with regulations, saves the user to the dataset; if not, prompts the user to enter new information.

Scenario 17

Use Case Name	*segmentPhoto*

Participating Actors	System
Entry Conditions	· User uploads a picture.
Exit Conditions	· The picture is outputted after segmentation.
Main Flow of Events	System: 1. Receives the picture and forwards it to the *segmentation module*. 2. The picture is segmented in the *segmentation module*. 3. The segmented picture is returned for further use.

Scenario 18

Use Case Name	*uploadToCloset*
Participating Actors	System
Entry Conditions	· The segmented picture is provided as an input.
Exit Conditions	· The picture is uploaded to the closet part of the *database*.
Main Flow of Events	System: 1. Receives the picture and uploads it to closet part of the *database*.

Scenario 19

Use Case Name	*appendToWishlist*
Participating Actors	System
Entry Conditions	· The segmented picture is provided as an input.
Exit Conditions	· The picture is uploaded to the wishlist part of the *database*.

Main Flow of Events	System: 1. Receives the picture and uploads it to wishlist part of the *database*.

Scenario 20

Use Case Name	*compareToOtherClothes*
Participating Actors	System
Entry Conditions	· System is provided with a picture.
Exit Conditions	· The comparison of the picture of the clothing item with all the other clothes present in the closet ends.
Main Flow of Events	System: 1. Receives a picture of a clothing item. 2. Compares the item with others one by one. 3. Appends similar items to a list. 4. Returns the list of similar items.

Scenario 21

Use Case Name	*showSecondHandAlternatives*
Participating Actors	System
Entry Conditions	· System is provided with a segmented picture.
Exit Conditions	· System finds all the alternatives available from the online database.

Main Flow of Events	System: 1. Receives a segmented picture of a clothing item. 2. Compares the item one by one with the items available in the online database. 3. Appends the alternatives to a list. 4. Returns the list of alternative clothing items.

Scenario 22

Use Case Name	*accessOnlineDatabase*
Participating Actors	System
Entry Conditions	· The process of seeking alternative options is initiated.
Exit Conditions	· Online database is reached and ready to be used.
Main Flow of Events	System: 1. Receives request to query the database. 2. Reaches the online database containing available clothing items for purchase.

Scenario 23

Use Case Name	*calculateSimilarityScore*
Participating Actors	System
Entry Conditions	· The process to compare two clothing items is initiated.
Exit Conditions	· Similarity score between two clothing items is returned.

Main Flow of Events	System: 1. Forwards the images are to the *similarity module*. 2. Similarity module calculates the similarity of the two clothes and returns a score. 3. Returns the score for further use.

Scenario 24

Use Case Name	*recommendOutfit*
Participating Actors	System
Entry Conditions	· System receives a picture of a clothing item.
Exit Conditions	· The iteration of available clothing items in the closet ends.
Main Flow of Events	System: 1. Iterates through the closet by choosing items one by one. 2. Sends the chosen item with the input item together to the *combination module*. 3. *Combination module* returns possible combinations back. 4. Returns the combinations for further use.

Scenario 25

Use Case Name	*addItem*
Participating Actors	System

Entry Conditions	· An item is received to be added to the *user database*.
Exit Conditions	· The item is added to the *user database*.
Main Flow of Events	System: 1. Accesses the database. 2. Uploads the item.

Scenario 26

Use Case Name	*removeItem*
Participating Actors	System
Entry Conditions	· A query to an item is received for it to be removed to the *user database*.
Exit Conditions	· The item is removed to the *user database*.
Main Flow of Events	System: 1. Accesses the database. 2. Finds the item with given information. 3. Deletes the item.

Scenario 27

Use Case Name	*getItem*
Participating Actors	System
Entry Conditions	· A query to an item is received for it to be returned from the *user database*.
Exit Conditions	· The item is returned from the *user database*.

Main Flow of Events	System: 1. Accesses the database. 2. Finds the item with given information. 3. Returns the item.

Scenario 28

Use Case Name	*showCloset*
Participating Actors	System
Entry Conditions	· System is prompted to show the available items in the closet.
Exit Conditions	· The items in the closet are returned.
Main Flow of Events	System: 1. Accesses the *user database*. 2. Iterates through the closet and appends the items in closet to a list 3. Returns the list for further use.

Scenario 29

Use Case Name	*showWishlist*
Participating Actors	System
Entry Conditions	· System is prompted to show the available items in the wishlist.
Exit Conditions	· The items in the wishlist are returned

Main Flow of Events	System: 1. Accesses the *user database*. 2. Iterates through the closet and appends the items in wishlist to a list 3. Returns the list for further use.

Scenario 30

Use Case Name	*tryOnModel*
Participating Actors	System
Entry Conditions	· System receives a request to try a clothing item on the model.
Exit Conditions	· The image of the model with the clothing item is returned.
Main Flow of Events	System: 1. Sends the image of the clothing item to the *testing module*. 2. *Testing module* processes the image of the clothing item and puts it on the model. 3. *Testing module* returns the image of the model to the system. 4. Returns the image for further use.

3.5.2. Use Case Model

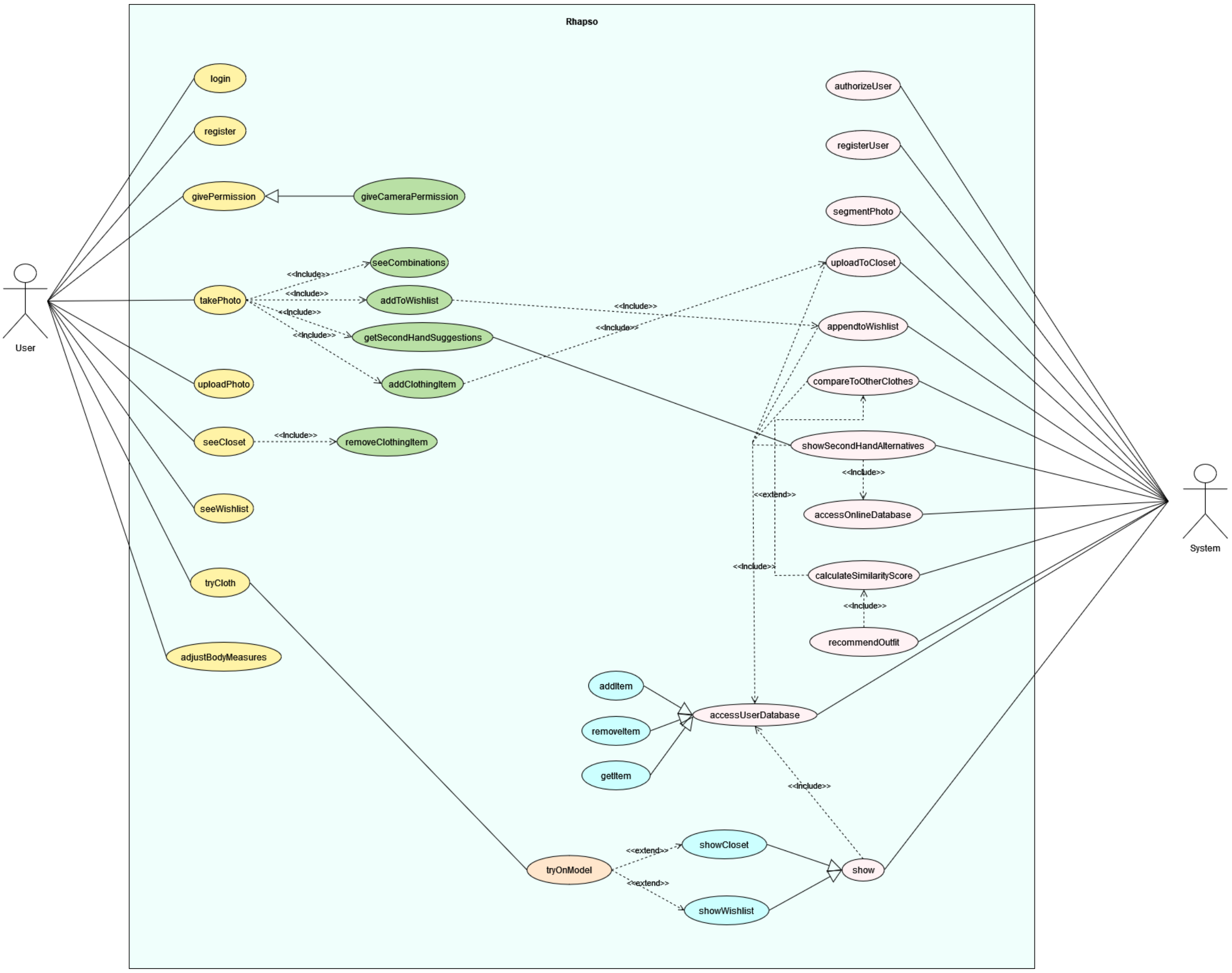


Fig. 1: Use Case Model. A higher quality version of this diagram is provided in the Github page of the project.

3.5.3. Object and Class Model - melike

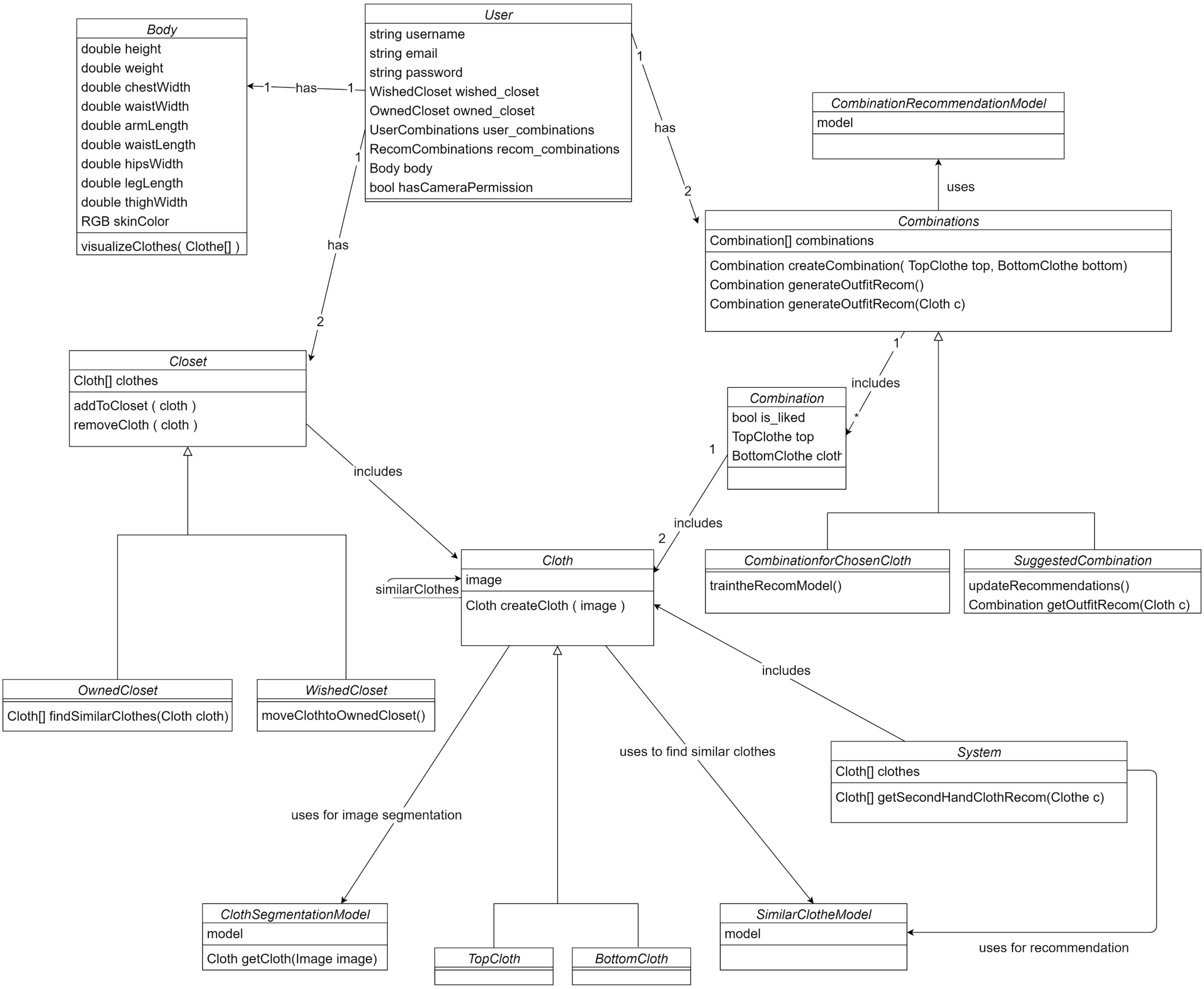


Fig. 2: Object and Class Diagram for Rhapso. A higher quality version of this diagram is provided in the Github page of the project.

3.5.4. Dynamic Models

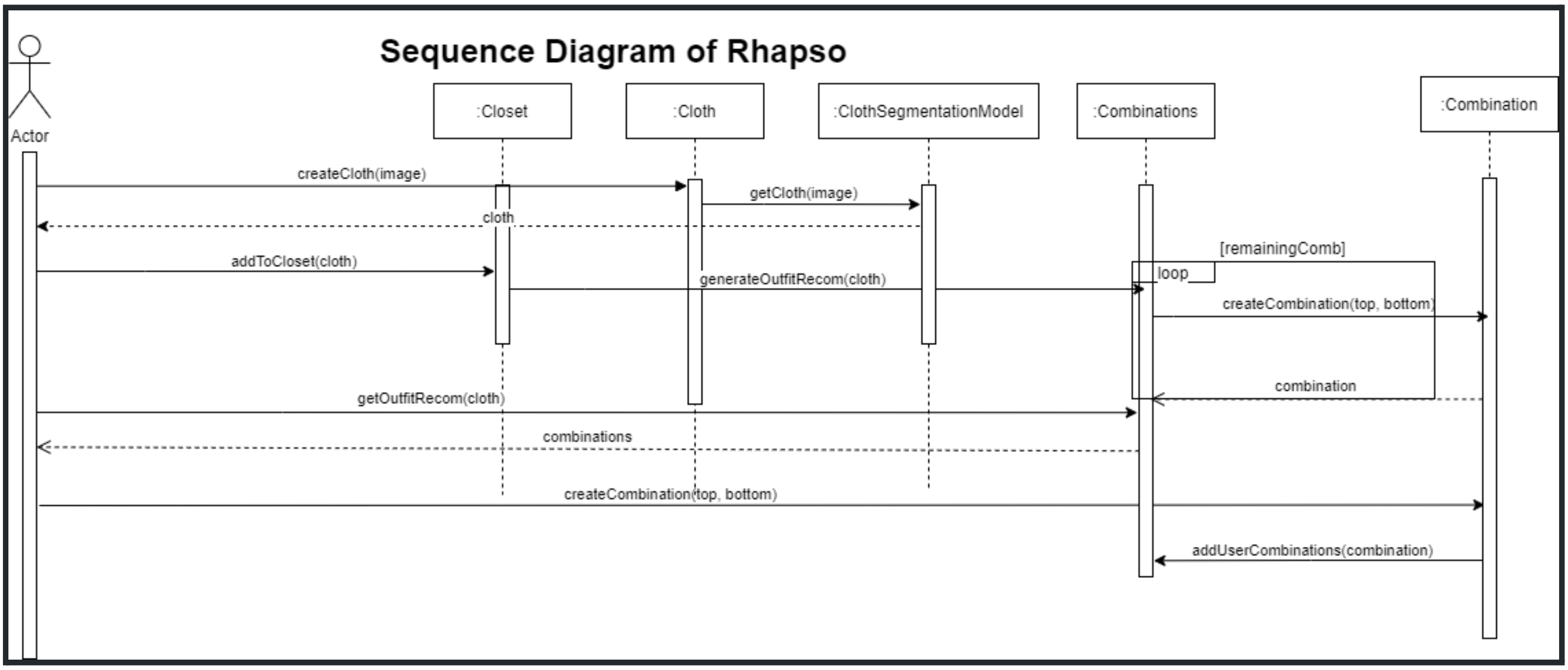


Fig. 3: Sequence Diagram of adding a clothing item, getting an outfit recommendation and adding a combination.

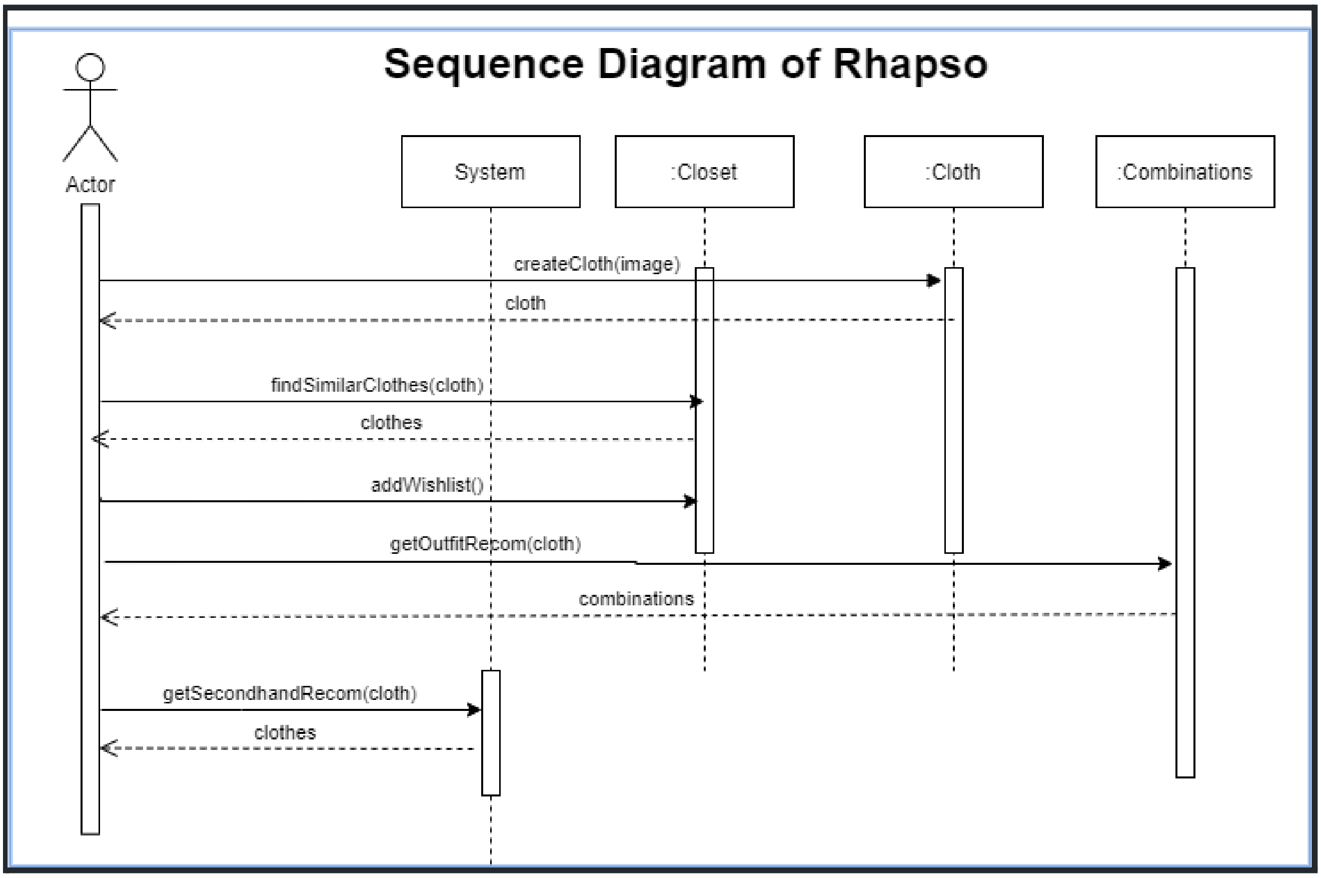


Fig. 4: Sequence Diagram of uploading an image of a clothing item, getting similarity score for it, adding that clothing item to the wishlist and getting suggestions for secondhand shopping.

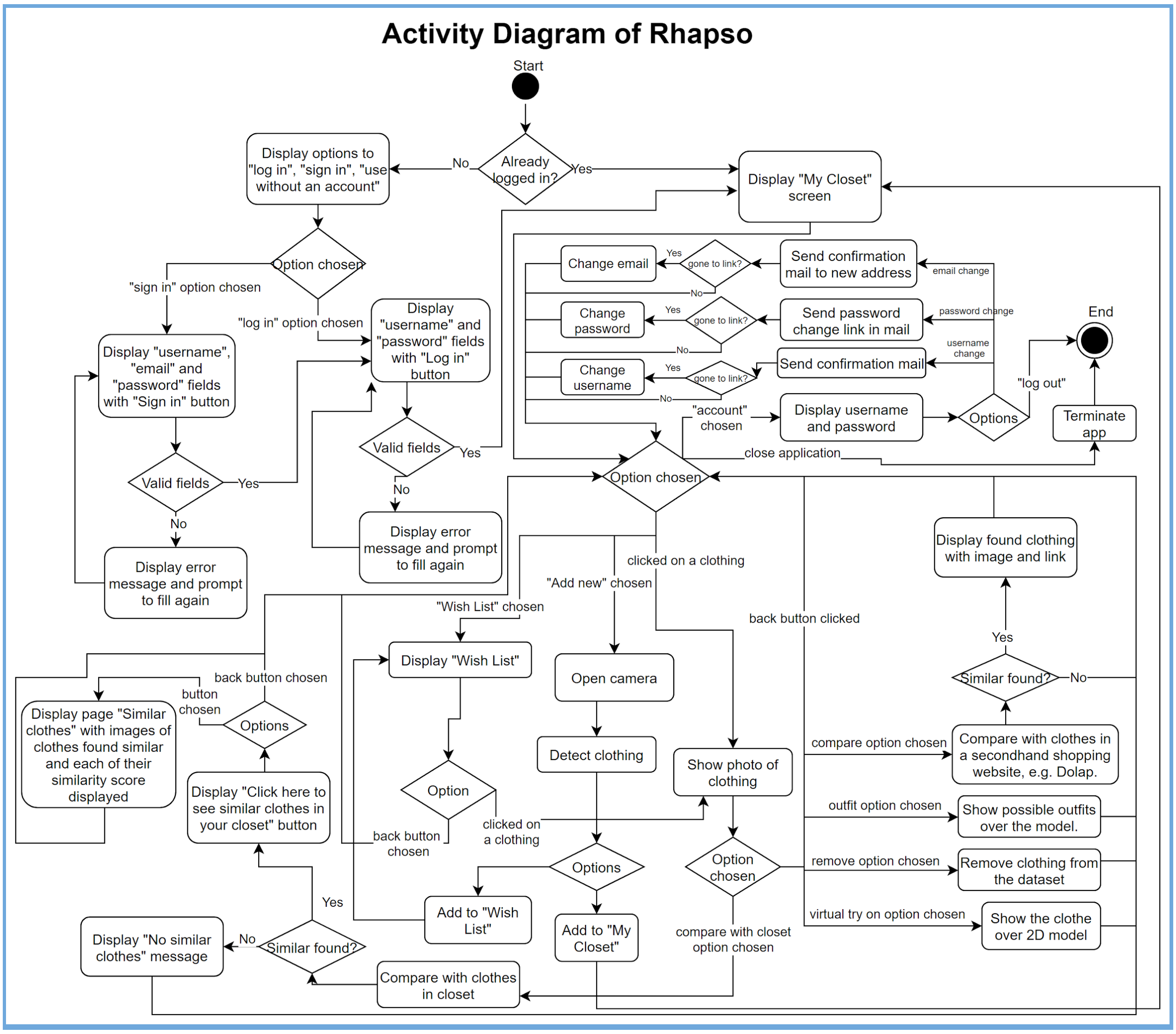

Activity Diagram of Rhapso
Start
Already logged in?
No
Yes
Display options to "log in", "sign in", "use without an account"
Display "My Closet" screen
Option chosen
"sign in" option chosen
"log in" option chosen
Display "username", "email" and "password" fields with "Sign in" button
Display "username" and "password" fields with "Log in" button
Valid fields
Yes
No
Display error message and prompt to fill again
Valid fields
Yes
No
Display error message and prompt to fill again
Change email
Yes
gone to link?
No
Send confirmation mail to new address
email change
Change password
Yes
gone to link?
No
Send password change link in mail
password change
Change username
Yes
gone to link?
No
Send confirmation mail
username change
"account" chosen
Display username and password
Options
"log out"
End
Terminate app
close application
Option chosen
clicked on a clothing
"Add new" chosen
"Wish List" chosen
back button clicked
Display found clothing with image and link
Yes
Similar found?
No
Compare with clothes in a secondhand shopping website, e.g. Dolap.
compare option chosen
Show possible outfits over the model.
outfit option chosen
Remove clothing from the dataset
remove option chosen
Show the clothe over 2D model
virtual try on option chosen
back button chosen
button chosen
Options
Display page "Similar clothes" with images of clothes found similar and each of their similarity score displayed
Display "Click here to see similar clothes in your closet" button
Display "Wish List"
Option
back button chosen
clicked on a clothing
Open camera
Detect clothing
Show photo of clothing
Option chosen
Options
Add to "Wish List"
Add to "My Closet"
compare with closet option chosen
Yes
Display "No similar clothes" message
No
Similar found?
Compare with clothes in closet

Fig. 5: Activity Diagram. A higher quality version of this diagram is provided in the Github page of the project.

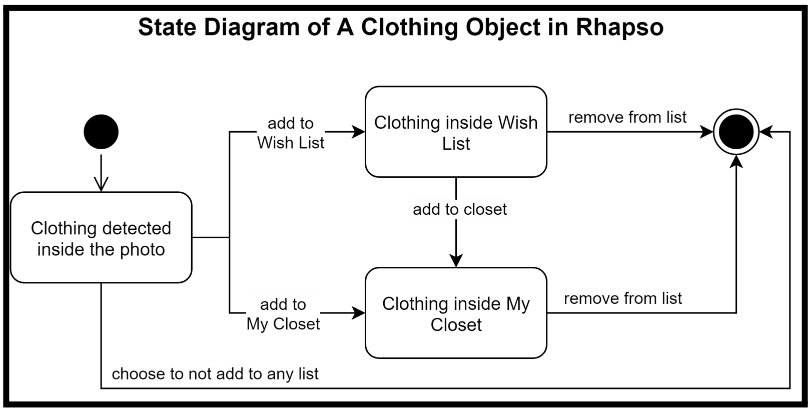
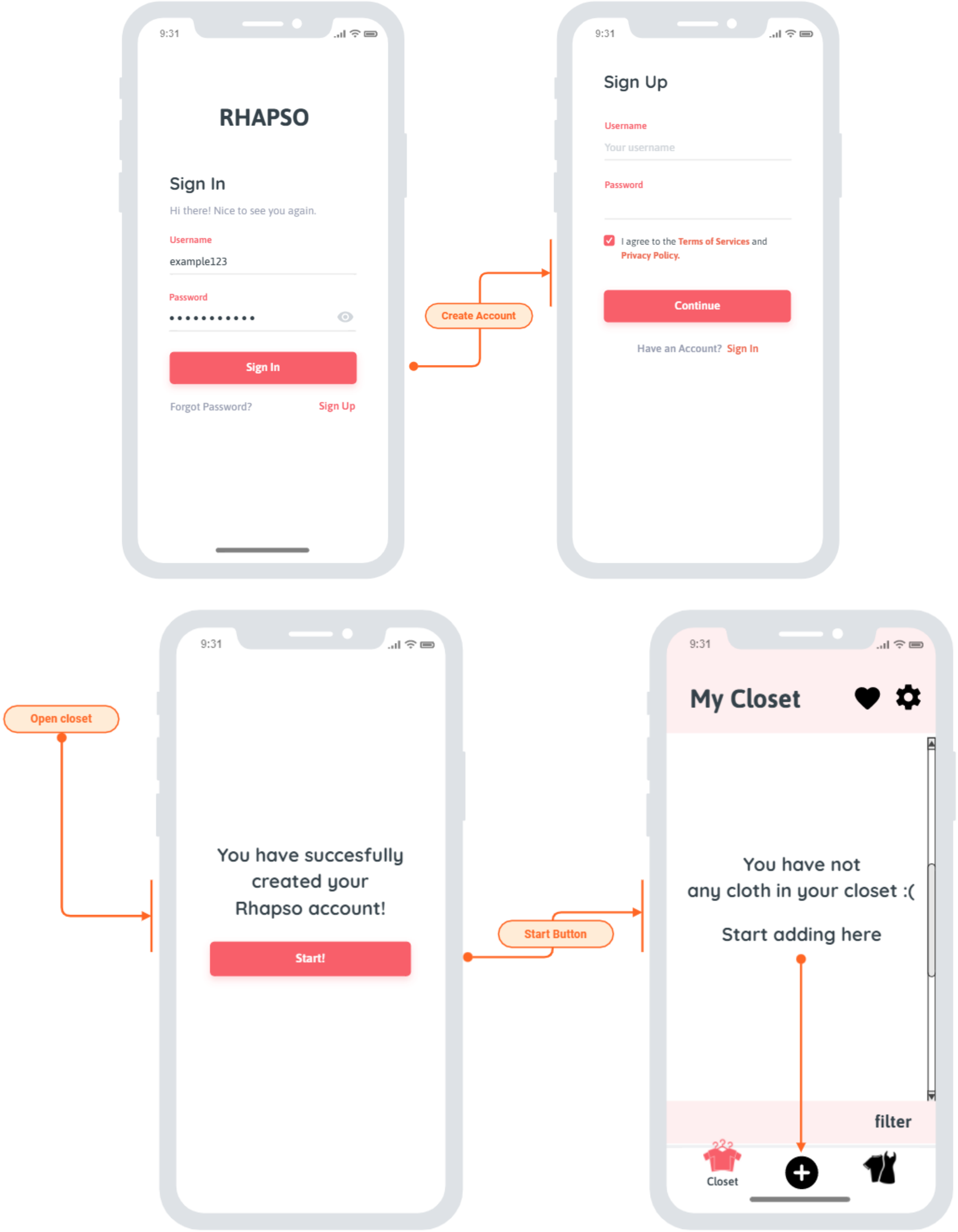

State Diagram of A Clothing Object in Rhapso
Clothing detected inside the photo
add to Wish List
Clothing inside Wish List
remove from list
add to closet
add to My Closet
Clothing inside My Closet
remove from list
choose to not add to any list

Fig. 6: State Diagram of A Clothing Object. A higher quality version of this diagram is provided in the Github page of the project.

3.5.5. User Interface - Navigational Paths and Screen Mock-ups



9:31

RHAPSO

Sign In

Hi there! Nice to see you again.

Username

example123

Password

••••••••••••

Sign In

Forgot Password? Sign Up

9:31

Sign Up

Username

Your username

Password

☑ I agree to the Terms of Services and Privacy Policy.

Continue

Have an Account? Sign In

9:31

You have succesfully created your Rhapso account!

Start!

9:31

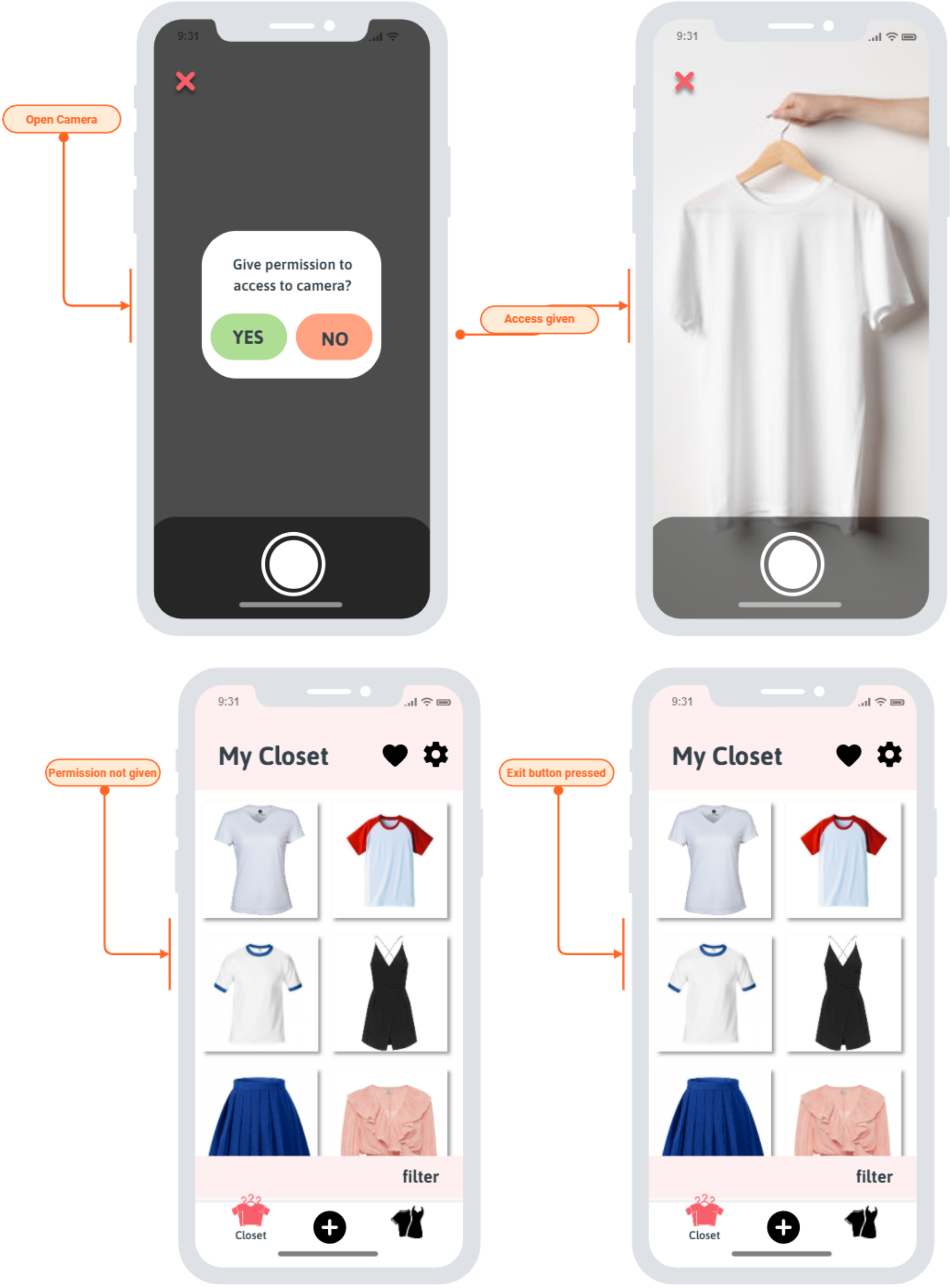
My Closet

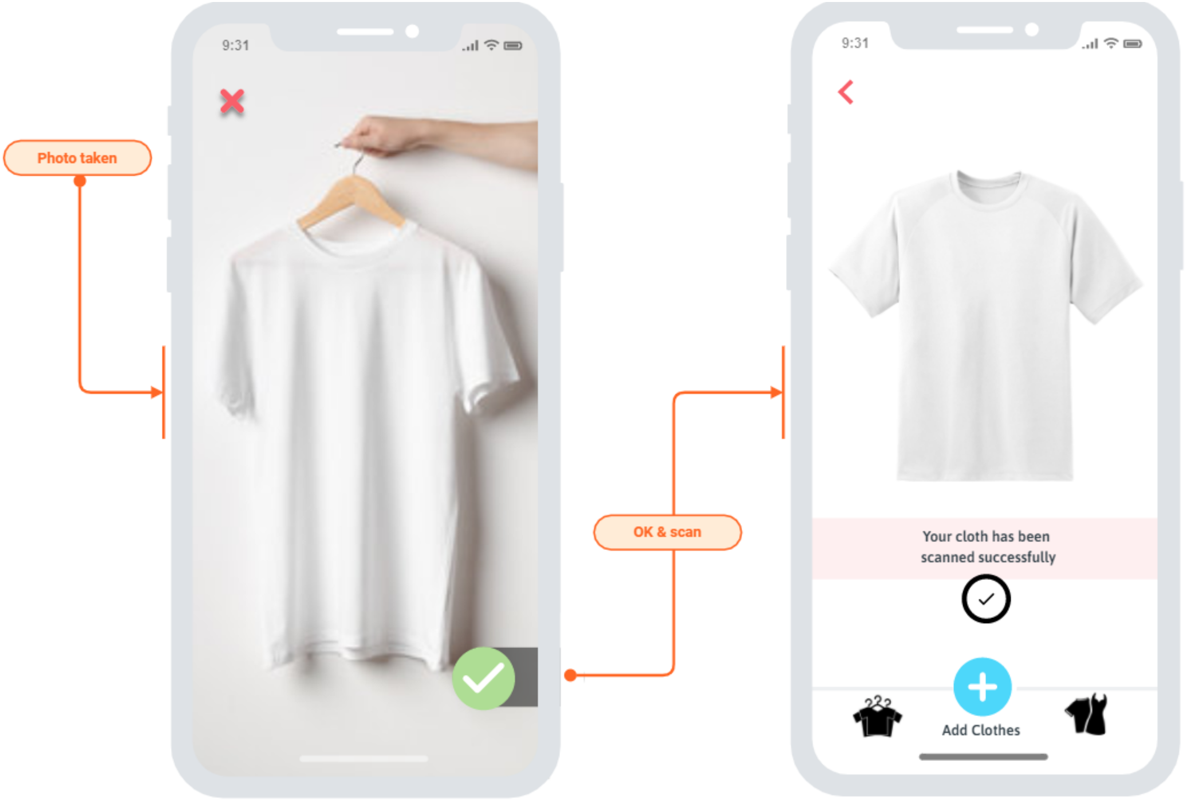
You have not any cloth in your closet :(

Start adding here

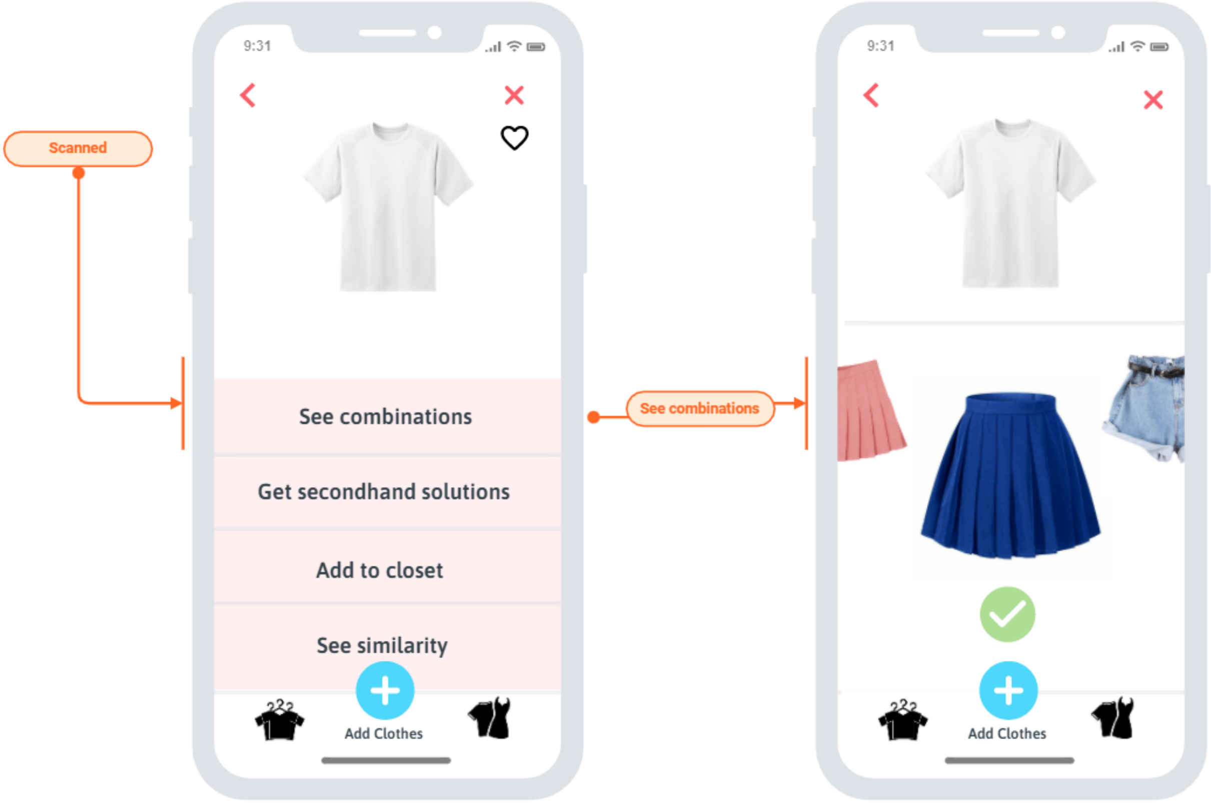
filter

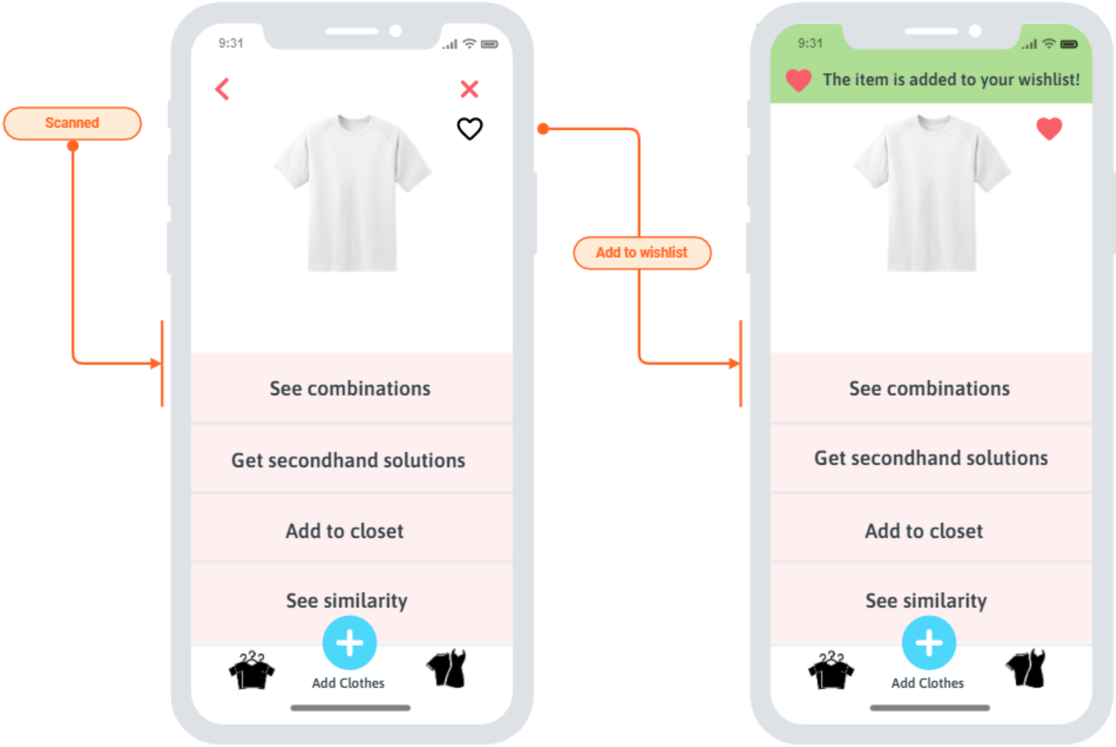
Closet


Open Camera
9:31
Give permission to access to camera?
YES
NO
Access given
9:31
Permission not given
9:31
My Closet
filter
Closet
Exit button pressed
9:31
My Closet
filter
Closet

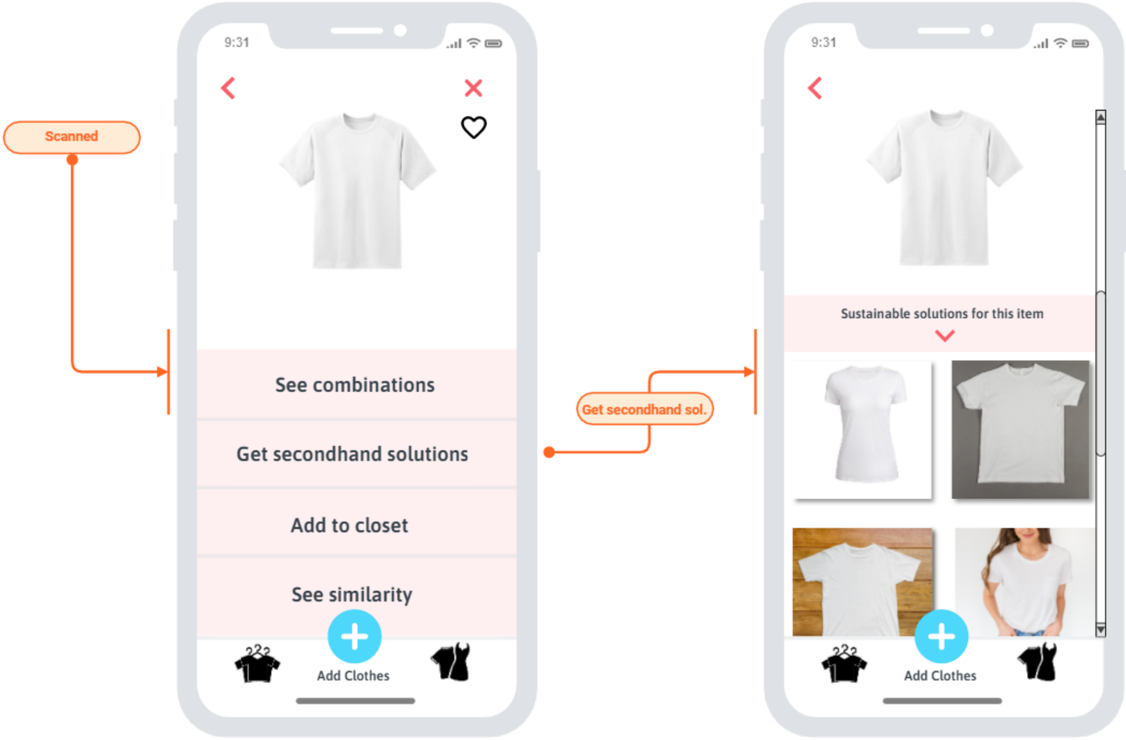


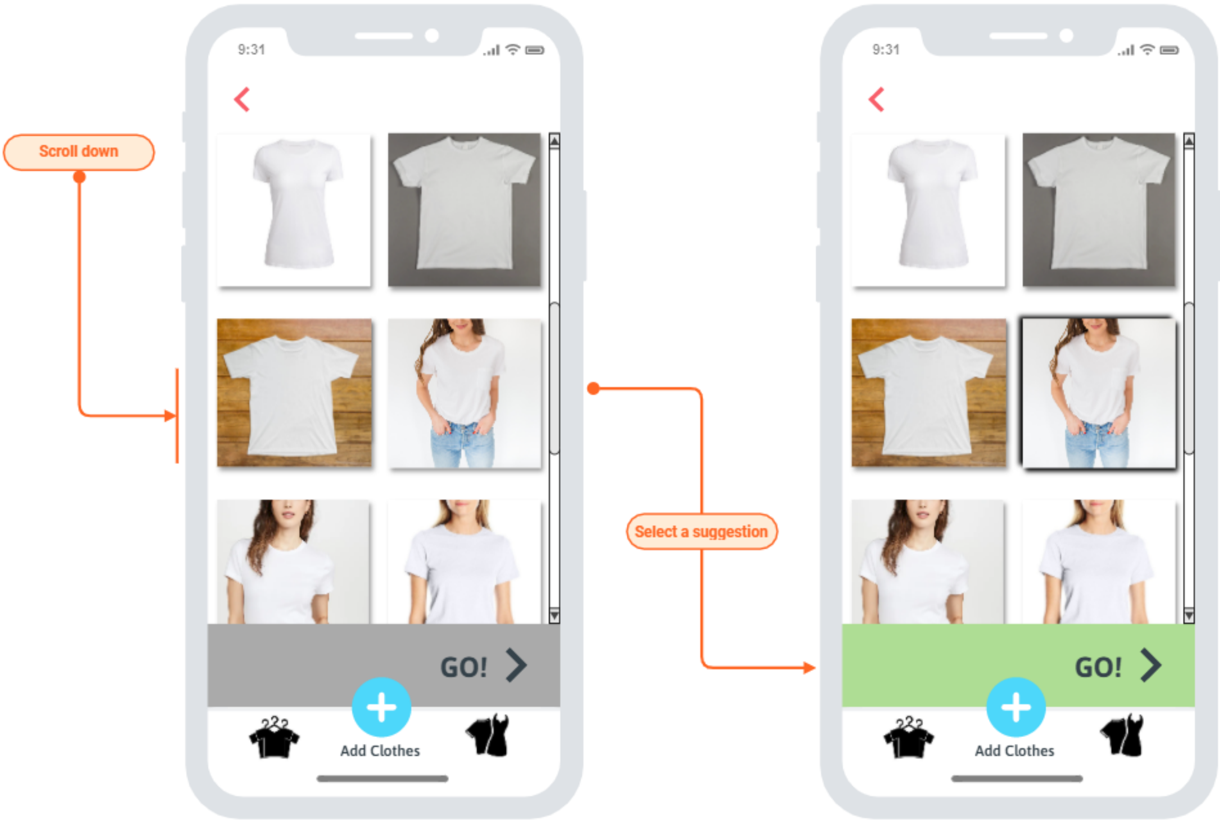
Your cloth has been scanned successfully

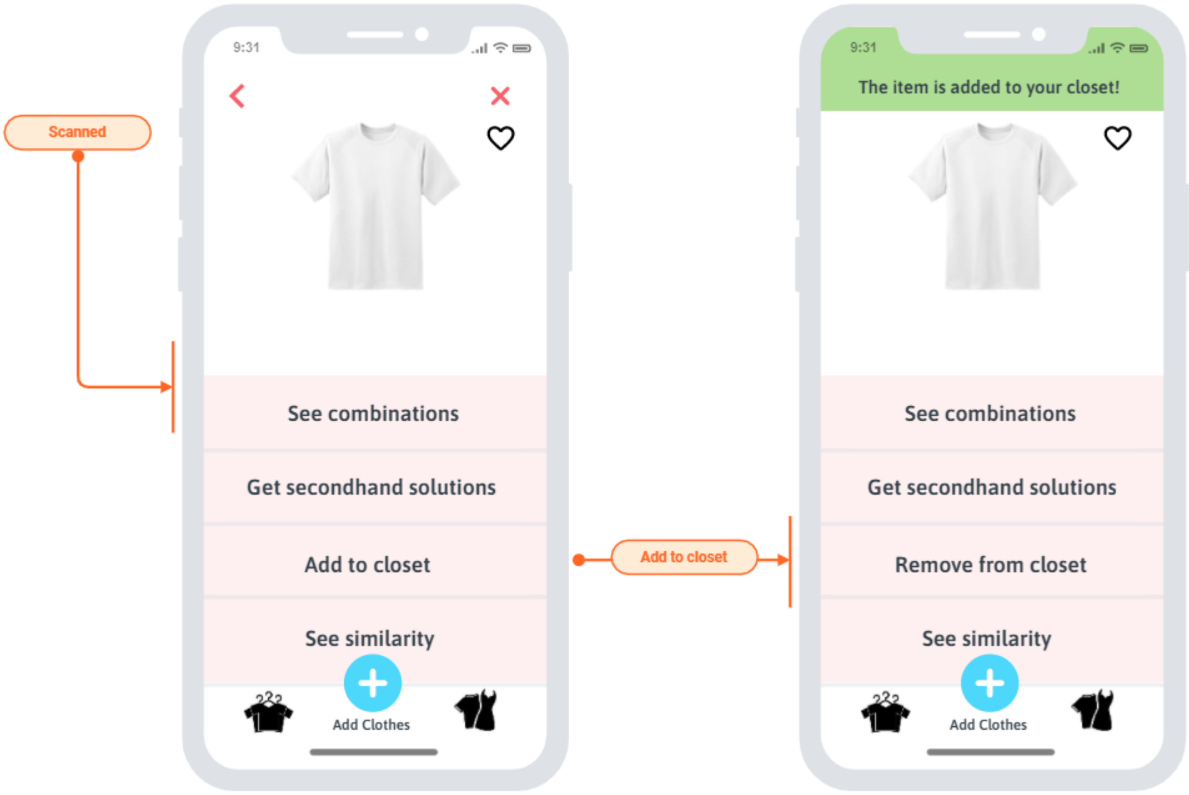




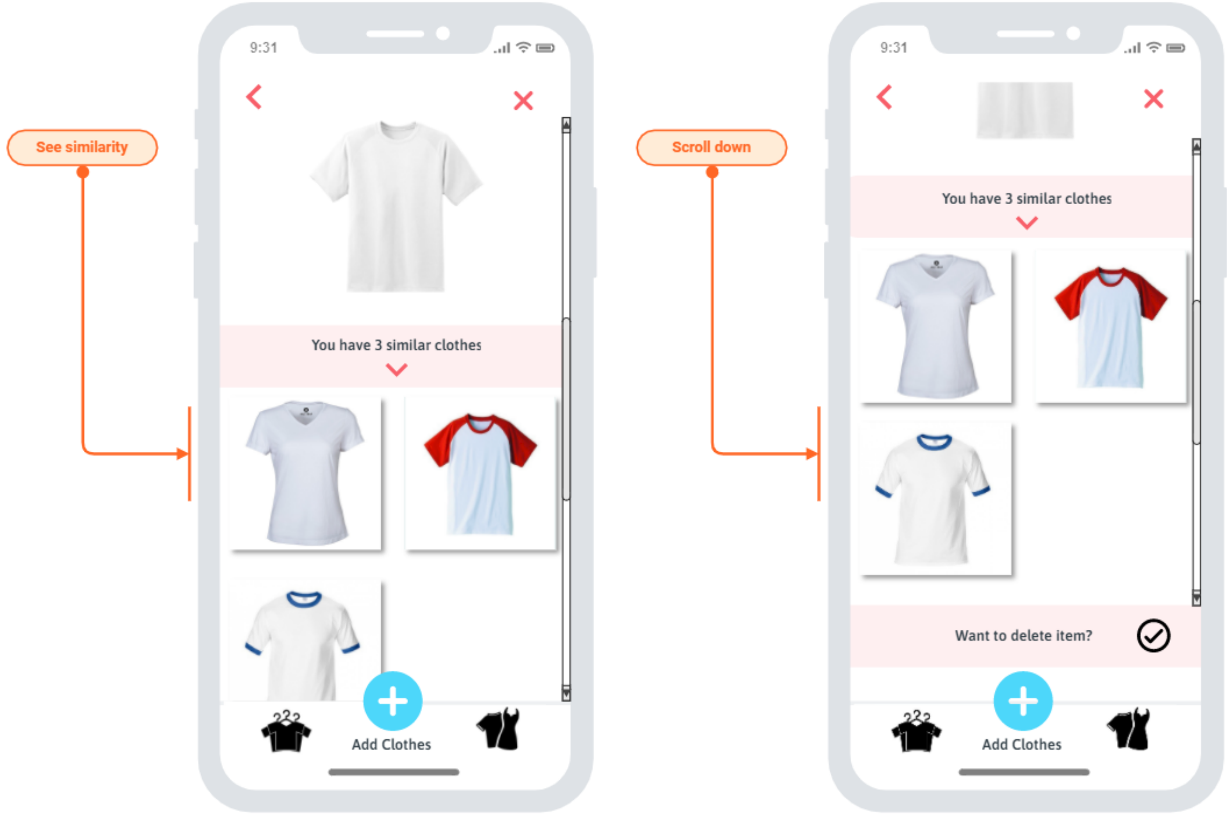
The item is added to your wishlist!

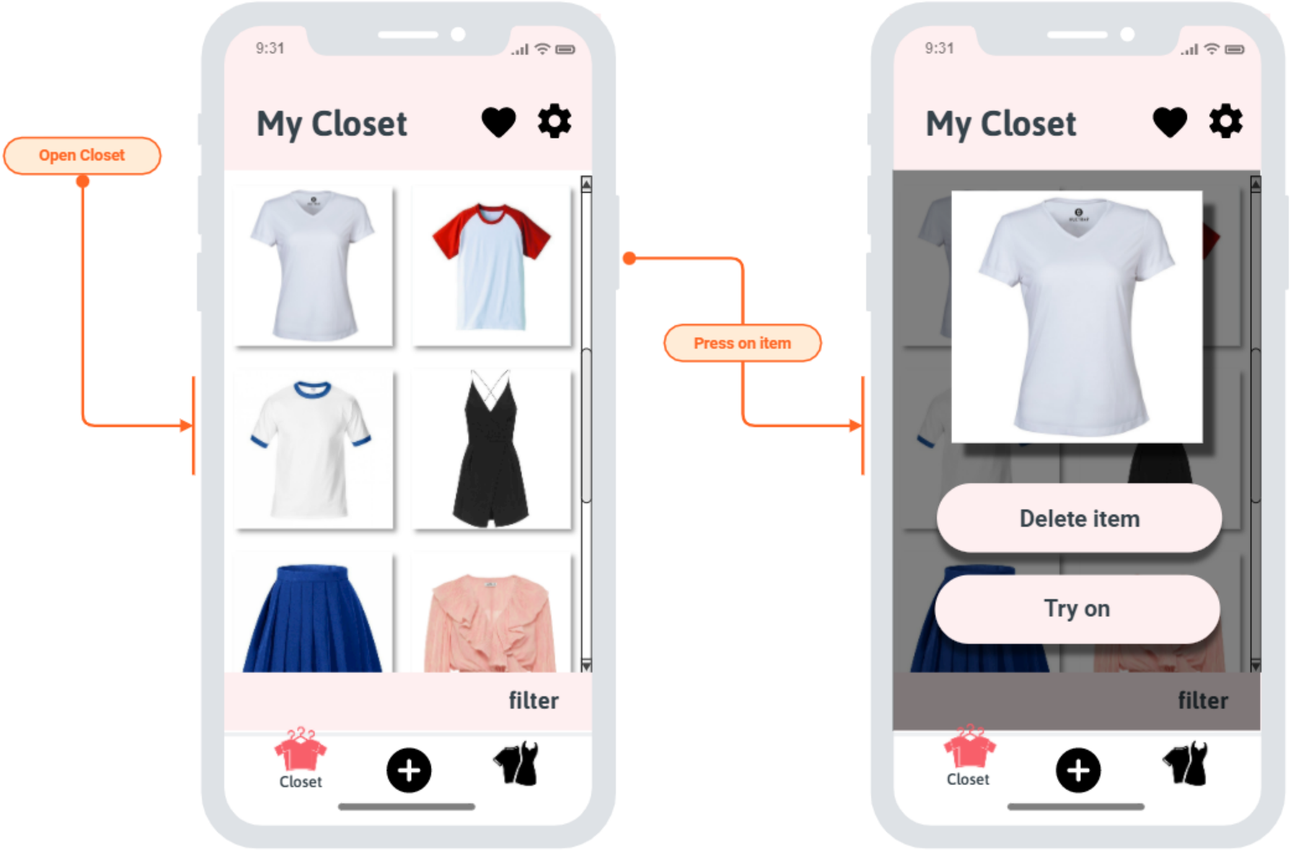


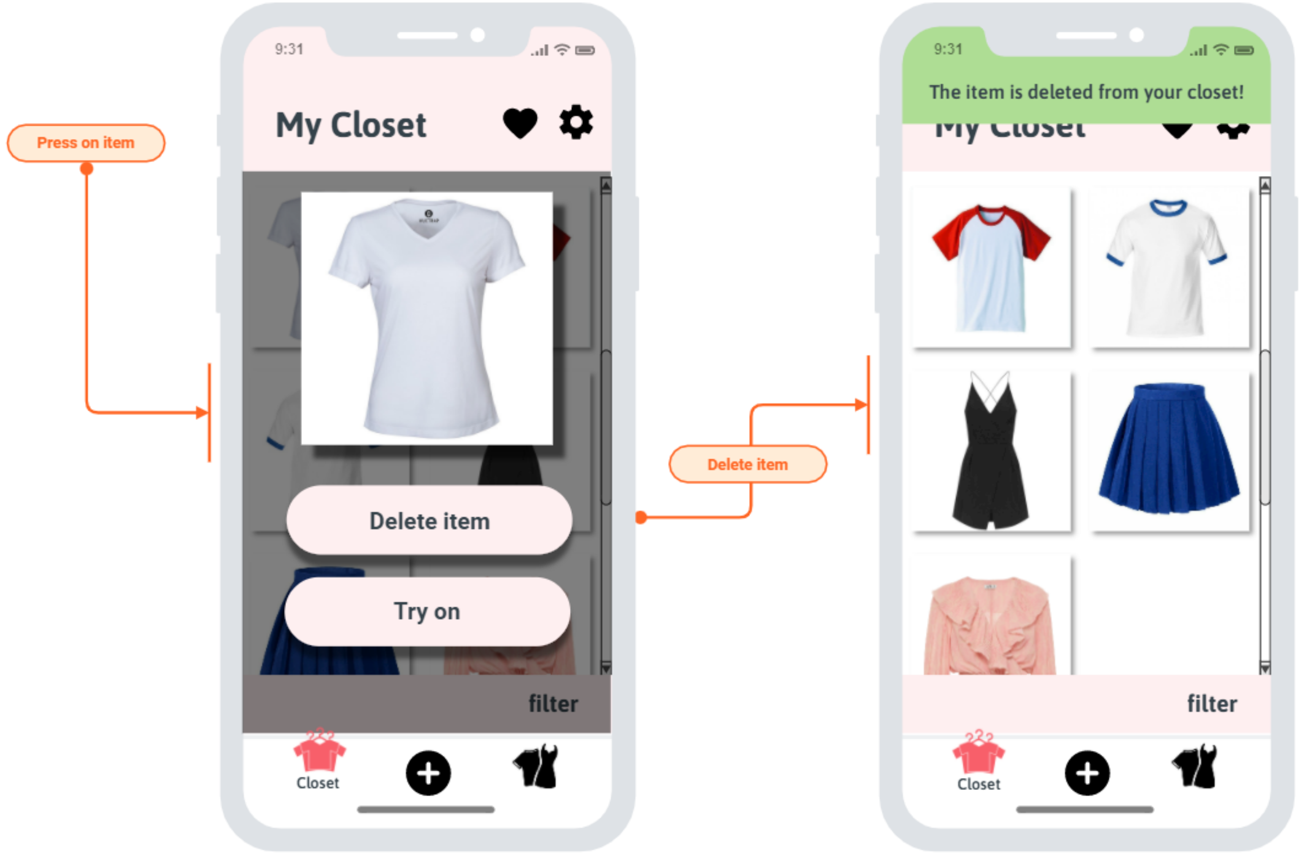




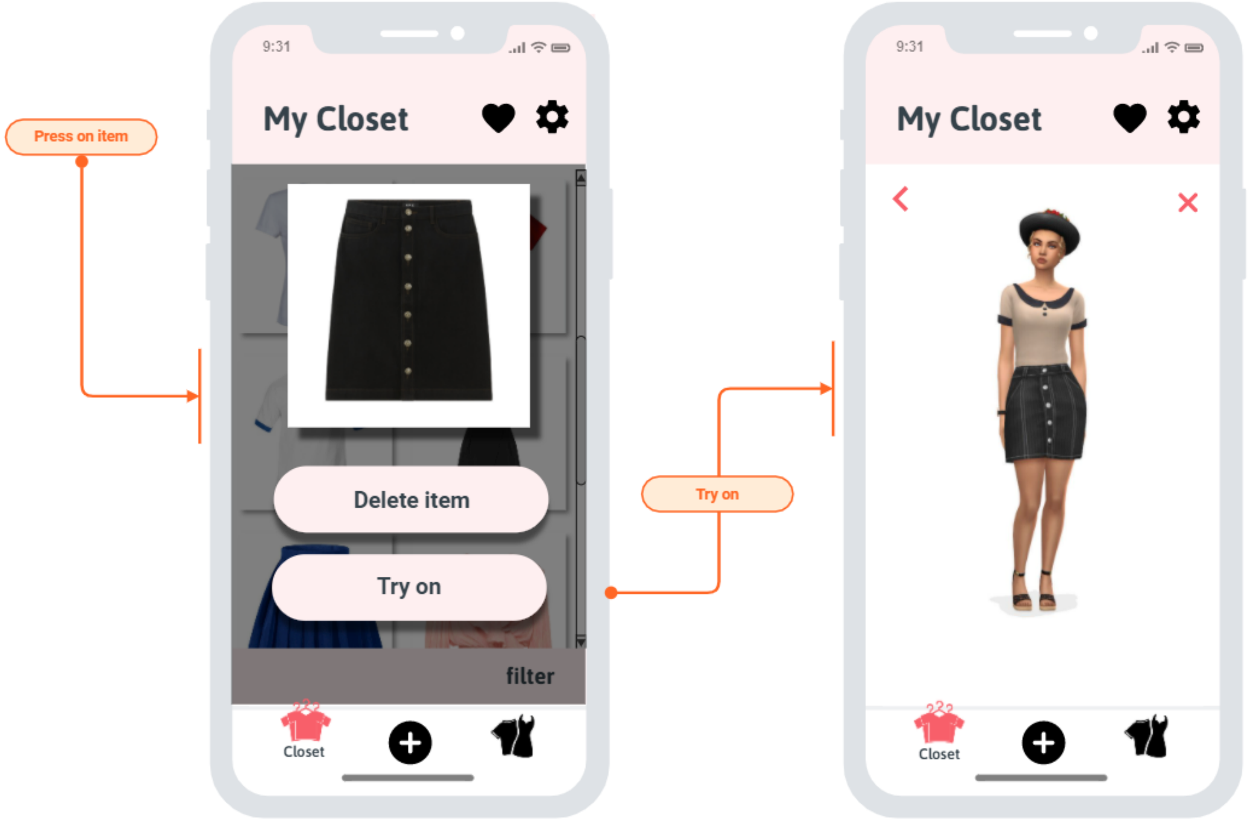
The item is added to your closet!


See similarity
9:31
You have 3 similar clothes
Add Clothes
Scroll down
9:31
You have 3 similar clothes
Want to delete item?
Add Clothes

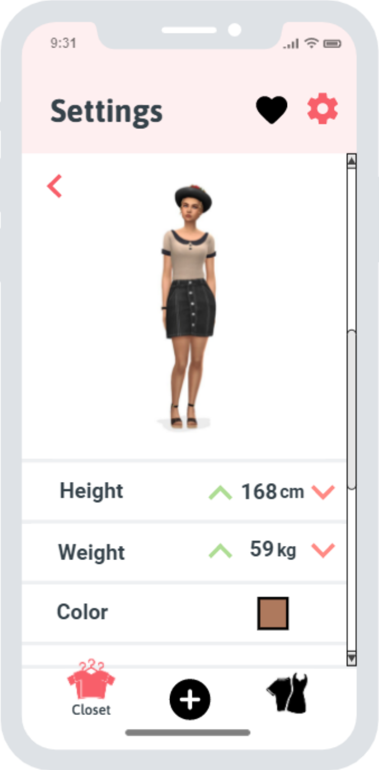

Open Closet
9:31
My Closet
filter
Closet
Press on item
9:31
My Closet
Delete item
Try on
filter
Closet



The item is deleted from your closet!



Settings



9:31

Settings

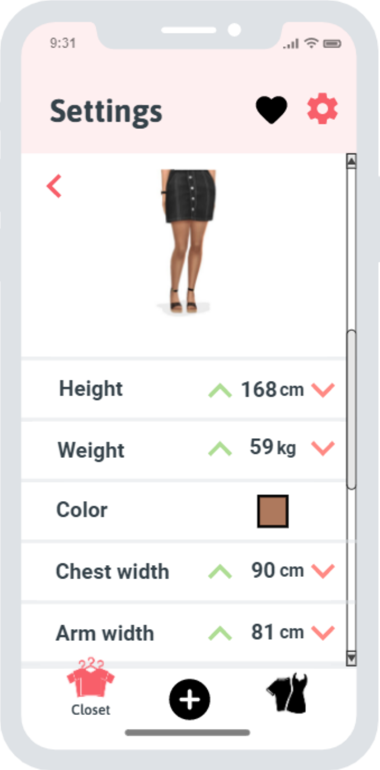
Height 168 cm

Weight 59 kg

Color

Closet

Scroll down



9:31

Settings

Height 168 cm

Weight 59 kg

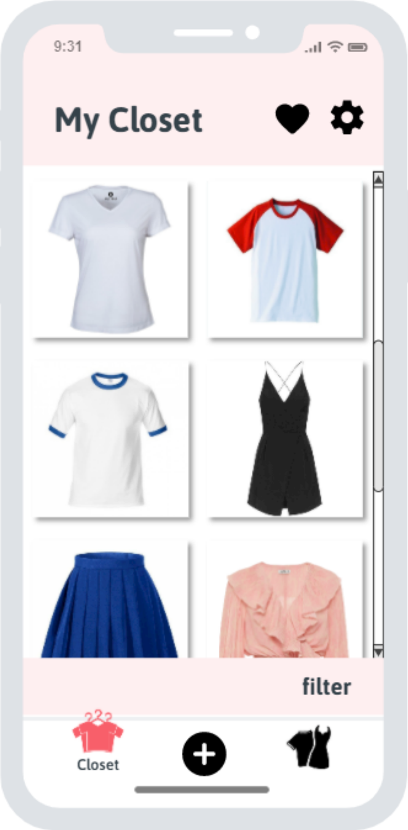
Color

Chest width 90 cm

Arm width 81 cm

Closet

Open closet



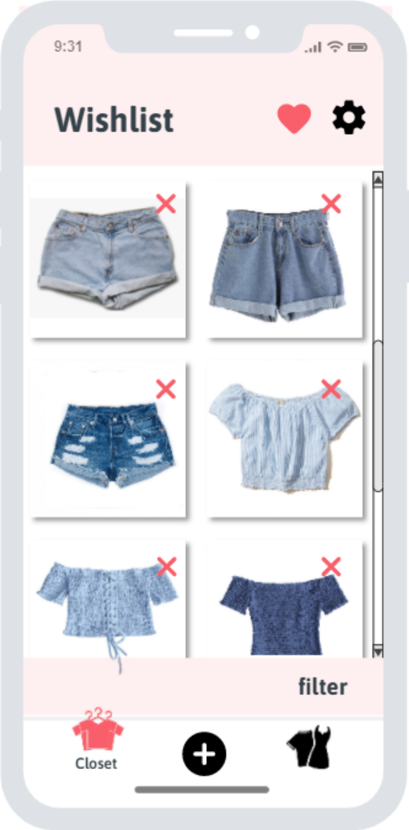
9:31

My Closet

filter

Closet

Wishlist Button

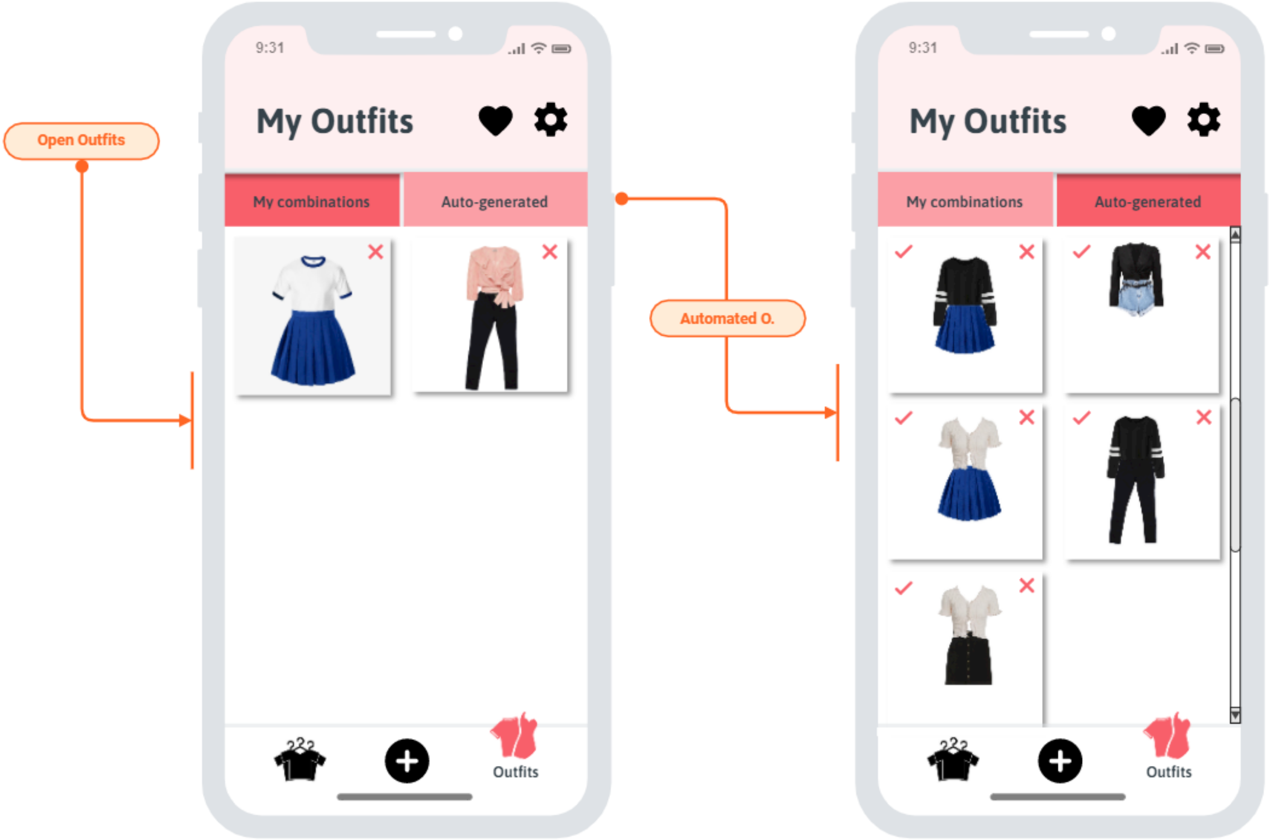


9:31

Wishlist

filter

Closet



4. Other Analysis Elements

4.1. Consideration of Various Factors in Engineering Design

There are various factors that will be considered during the development of Rhapso. Those factors will be limiting us at some points.

4.1.1. Social Factors

We have considered the factors such as age, economic status, religion, race. Based on the age, the preferences of the people changes so that our recommendation algorithm will be affected by this. Furthermore, the way people use Rhapso will also change. For example, a teenager will use Rhapso mostly for outfit recommendations but an adult will use it not to buy similar clothes. In terms of religion, people belong to different religions and based on their religious orientation, their outfit will change accordingly. Rhapso will be suggesting combines based on the existing clothes so that the combines will reflect the religious orientation of the people. Likewise, according to the economic status, people with lower income may desire to buy the thing they really need. From this point of view, they will be mostly interested in the similarity score feature.

4.1.2. Economic Factors

There are economic benefits provided by Rhapso. The main aim of Rhapso is to reduce the amount of clothing shopping. There are two main reasons that cause over-shopping. One of them is people could not find outfits, many people complain that they have nothing to wear. As a result, they generally decide to go shopping, however, Rhapso will have a feature that recommends outfits by using the clothes in the virtual gardrobe. Therefore, even if the people cannot decide on what to wear, Rhapso will help them by generating suggestions and this will decrease the possibility of desire to buy new clothes. Another main reason is people may forget the clothes they have. For example, in winter, the summer clothes are put in the bags, and generally, there are discounts on summer clothes during winter. Many people use that opportunity to buy cheaper things, however, they may not remember their existing clothes and buy very similar clothes. Rhapso provides another feature called similarity score which generates a similarity score for the given clothes based on the ones in the virtual closet. Therefore, people may not prefer similar clothes, and save that money.

4.1.3. Environmental Factors

There is not any environmental harm that is caused by Rhapso however, unnecessary shopping may result in environmental damage in the long term. Especially with the fast fashion in which manufacturing of cloth became cheaper and easier, through new materials like polyester and nylon, the unnecessary production of clothes made by those poor quality materials increases. Following this trend, people are buying very similar clothes over and over. To prevent this to some extent, we will develop another model for similarity check. The model will produce a similarity score by comparing an uploaded cloth with the existing ones. As a result, the user will be aware if s/he has a similar dress or not.

4.1.4. Global Factors

In our discussion, what is meant by global factors is the rules, norms and some regulations that countries have an agreement on. Some of the standards are General Data Privacy Regulation (GDPR) [10], Google Play Terms of Service [11] and other global agreements between the developers, users and governments.It is significant that we limit ourselves to these factors since those are the opinions of the public who will be using our product. Thus, considering those factors during development is important.

4.1.5. Cultural Factors

As mentioned before, models that produce outfit suggestions will be retrained by using the data uploaded by the users. For example, a new recommendation will be generated by using the previous outfit pair data. There will be data from different countries, cultures which may cause a bias in our models. For example, in some countries, people may prefer wearing more colorful clothes, while others prefer darker colors. To prevent such a cultural bias incorporated into our model, we will train our model with equally distributed data.

4.2. Risks and Alternatives

Rhapso contains several implementation risks that might be significant in the development phase. Being aware of those risks and working on the back-up plans are must for us.The potential risks are mentioned and possible alternatives are explained below.

4.2.1. Unsatisfactory Server Performance

Rhapso will be using Cloud service which requires servers. Since our models which are used to detect the clothes and to generate AI-based recommendations are difficult to run and having short response time is important, the server performance is substantial. If we face any situation in which the server performance is unsatisfactory, we will find other server options

4.2.2. High Server Costs

Another possible risk is the high cost of servers that exceeds our budget. We are not given any budget for the project which means we are going to cover the financial expenses out of our pocket. Therefore, if we face such a problem, we are going to optimize our models/algorithms and in the worst case try to run them in the local machine. In order to have extra financial sources, we might add advertisements to our application. It is also important to keep in mind that there is a trade-off between server performance and the server cost.

4.2.3. Discrepancy Among Group Members

It is possible that team members cannot agree on some decisions so that discrepancy occurs within the team. In those situations, we are going to follow the democratic rules and accept the decision of the majority.

4.2.4. Low Performance

The similarity check may require too much time. That feature should generate a similarity score as quickly as possible so that the user can benefit during the shopping. Otherwise, it may not serve its purpose as desired. Moreover, the virtual body will be adjustable in terms of the body measures so that when the user wants to see the combination on it, the photos of the clothes should be adjusted to be fit into the model. This should be done again as quickly as possible for a better user experience.

Risk Description	Likelihood	Effect on the Project	Action
Unsatisfactory Server Performance	Medium	Poor user experience	Adopting new servers
High Server Cost	Medium	Necessity of optimization	Reducing the need of servers and optimization for certain features
Discrepancy Among Group Members	Medium	Decreases the efficiency of working and motivation among team members	Accepting the decision of the majority
Low Performance	High	Increases the response time, as decreases the user experience	Developing models with better performance

Table. 2: Risks and Alternatives

4.3. Project Plan

We can divide our project into seven main parts.

First part is the Project Specifications part. In this workload, we decided on the constraints of, professional & ethical issues related to and functional & non-functional requirements of the project. Also, we searched similar technologies.

Second part is the Analysis Report. In this workload, we update the requirements that we decided on the Project Specifications and create the models of our projects which are Use Case, Objects & Class, Dynamic models. Finally, we create user interfaces.

Third part is the High Level Design Report. In this workload, we explain our proposed software architecture, subsystem decomposition, hardware/software mapping, data management, accesses & security, global software control and subsystem services.

Fourth part is Demo. To show our basic features, we will focus on our machine learning models with the basic GUI. These models are the segmentation model for detecting a cloth in a photo, the combination recommendation model, the secondhand clothes recommendation model and the model that finds similar clothes. Furthermore, we will start to implement the body which we will show the clothes on.

Fifth Part is the Low Level Design Report. We will decide Object Design Trade-offs, prepare Interface Documentation guidelines and explain our packages, and class interfaces.

Sixth Part is the Final Report. We will finalize our architecture, document our implementation, explain our tests and explain our maintenance plan.

Seventh and the last part is the Final Demo. This part is given as the last part but we will be working on it through the second semester. For this package, we will implement our database, integration tests, visualization of clothes on body, backend of getting second hand clothes from shopping websites according to our similar clothes model, and we will train our models for more accurate results.

For the work packages which are listed below, we assigned two people for each of them and one of the members will lead the task. For the next tasks, this division might be updated. Also, some deadlines of the tasks are not announced so that they are given as “To Be Decided (TBD)”.

Work Package	Leader	Team Members	Deadline	Deliverables
1.1	Şebnem Uslu	Şebnem Uslu, Burak Yetiştiren	Oct 11, 2021	Project Specifications Report
1.2	Melike Aydoğan	Melike Aydoğan, Şebnem Uslu	Oct 11, 2021	
1.3	Defne Betül Çiftci	Defne Betül Çiftci, Melike Aydoğan	Oct 11, 2021	
1.4	Işık Özsoy	Işık Özsoy, Defne Betül Çiftci	Oct 11, 2021	
1.5	Burak Yetiştiren	Burak Yetiştiren, Işık Özsoy	Oct 11, 2021	
2.1	Defne Betül Çiftci	Defne Betül Çiftci, Şebnem Uslu	Nov 8, 2021	Analysis Report
2.2	Burak Yetiştiren	Burak Yetiştiren, Defne Betül Çiftci	Nov 8, 2021	
2.3	Melike Aydoğan	Melike Aydoğan, Işık Özsoy	Nov 8, 2021	
2.4	Işık Özsoy	Işık Özsoy, Melike Aydoğan	Nov 8, 2021	

2.5	Şebnem Uslu	Şebnem Uslu, Burak Yetiştiren	Nov 8, 2021	
3.1	Şebnem Uslu	Şebnem Uslu, Işık Özsoy	Dec 24, 2021	High Level Design Project
3.2	Işık Özsoy	Işık Özsoy, Defne Betül Çiftci	Dec 24, 2021	
3.3	Burak Yetiştiren	Burak Yetiştiren, Şebnem Uslu	Dec 24, 2021	
3.4	Melike Aydoğan	Melike Aydoğan, Şebnem Uslu	Dec 24, 2021	
3.5	Melike Aydoğan	Melike Aydoğan, Burak Yetiştiren	Dec 24, 2021	
3.6	Defne Betül Çiftci	Defne Betül Çiftci, Burak Yetiştiren	Dec 24, 2021	
3.7	Işık Özsoy	Işık Özsoy, Melike Aydoğan	Dec 24, 2021	
4.1	Şebnem Uslu	Şebnem Uslu, Işık Özsoy	Dec 24, 2021	Demo
4.2	Melike Aydoğan	Melike Aydoğan, Defne Betül Çiftci	Dec 24, 2021	
4.3	Işık Özsoy	Işık Özsoy, Burak Yetiştiren	Dec 24, 2021	
4.4	Burak Yetiştiren	Burak Yetiştiren, Şebnem Uslu	Dec 24, 2021	
4.5	Defne Betül Çiftci	Defne Betül Çiftci, Melike Aydoğan	Dec 24, 2021	
5.1	Burak Yetiştiren	Burak Yetiştiren, Melike Aydoğan	TBD	Low-Level Design Report
5.2	Şebnem Uslu	Şebnem Uslu, Işık Özsoy	TBD	
5.3	Işık Özsoy	Işık Özsoy, Melike Aydoğan	TBD	
5.4	Defne Betül Çiftci	Defne Betül Çiftci, Burak Yetiştiren	TBD	
6.1	Defne Betül Çiftci	Defne Betül Çiftci, Şebnem Uslu	TBD	Final Report
6.2	Burak Yetiştiren	Burak Yetiştiren, Melike Aydoğan	TBD	
6.3	Şebnem Uslu	Şebnem Uslu, Işık Özsoy	TBD	
6.4	Melike Aydoğan	Melike Aydoğan, Burak Yetiştiren	TBD	
7.1	Melike Aydoğan	Melike Aydoğan, Işık Özsoy	TBD	Final Demo
7.2	Işık Özsoy	Işık Özsoy, Burak Yetiştiren	TBD	
7.3	Defne Betül Çiftci	Defne Betül Çiftci, Melike Aydoğan	TBD	
7.4	Şebnem Uslu	Şebnem Uslu, Defne Betül Çiftci	TBD	
7.5	Burak Yetiştiren	Burak Yetiştiren, Şebnem Uslu	TBD	

Tab. 3: Distribution of work packages through the member

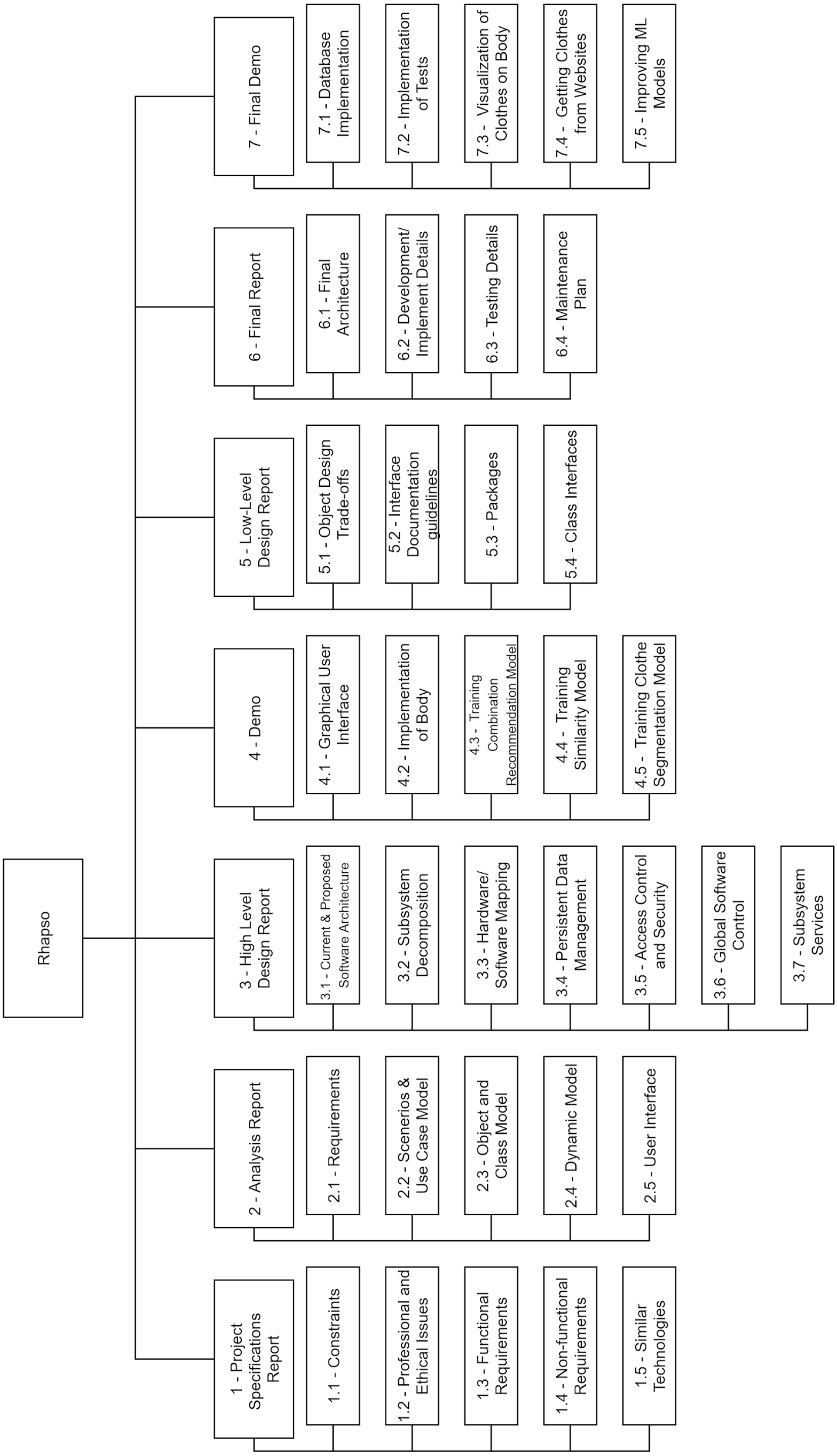

Rhapso
1 - Project Specifications Report
1.1 - Constraints
1.2 - Professional and Ethical Issues
1.3 - Functional Requirements
1.4 - Non-functional Requirements
1.5 - Similar Technologies
2 - Analysis Report
2.1 - Requirements
2.2 - Scenerios & Use Case Model
2.3 - Object and Class Model
2.4 - Dynamic Model
2.5 - User Interface
3 - High Level Design Report
3.1 - Current & Proposed Software Architecture
3.2 - Subsystem Decomposition
3.3 - Hardware/ Software Mapping
3.4 - Persistent Data Management
3.5 - Access Control and Security
3.6 - Global Software Control
3.7 - Subsystem Services
4 - Demo
4.1 - Graphical User Interface
4.2 - Implementation of Body
4.3 - Training Combination Recommendation Model
4.4 - Training Similarity Model
4.5 - Training Clothe Segmentation Model
5 - Low-Level Design Report
5.1 - Object Design Trade-offs
5.2 - Interface Documentation guidelines
5.3 - Packages
5.4 - Class Interfaces
6 - Final Report
6.1 - Final Architecture
6.2 - Development/ Implement Details
6.3 - Testing Details
6.4 - Maintenance Plan
7 - Final Demo
7.1 - Database Implementation
7.2 - Implementation of Tests
7.3 - Visualization of Clothes on Body
7.4 - Getting Clothes from Websites
7.5 - Improving ML Models

Fig. 7: Work Packages

4.4. Ensuring Proper Teamwork

For the proper teamwork, we are planning regular meetings. In each meeting we are reporting what we did after the last meeting and tasks we could not finish. Then, according to the left tasks we are distributing these and deciding the new meeting's date and the deadline of the tasks.

Also, necessary decisions are noted during the meetings and shared with team members with the task division. To share these items, we are using discord to create different channels just related to one topic. With this, we can find necessary items more easily without confusion.

Moreover, for proper teamwork, we respect each others' lecture related tasks and exams. That is the reason why we are deciding the next meeting during the last meeting to avoid workload.

4.5. Ethics and Professional Responsibilities

In this section, we want to discuss some possible impacts that our project may have on different contexts, and inversely the possible impacts of these contexts on our project. We found that the most relevant contexts worth discussing that reside in our project's scope are: **Societal**, **Global**, **Cultural**, **Environmental**, and **Economic** contexts.

We want to initiate our discussion with the Global context. With our team, we set our main ambition to virtualize the wardrobes of our users across the world; therefore, we are well aware of the extended comprehension of our project. We believe that the act of storing clothing items in some place is not limited by any group, territory, or entity. Thus the problem we want to deal with is global. By virtualizing some aspects in human life, we want to provide various functionality, which will act as an extension to activities related to the real-life wardrobes. Popular instances of our methodology could be seen in many applications available on the market today, like Instagram [12], for which one can argue that the application has replaced the traditional photo albums, and added many functionalities like photo filters, "like" functionality, instant messaging, et cetera. In line with this idea, we also want to enhance another real-life component with our virtual solution provided with functionality, which is not possible or likely to be executed in real-life scenarios. Having said aspects of our problem in mind, as we have discussed in the beginning of this section, we believe that the scope of our problem extends to a global context, and therefore our project has the potential to both have a global impact while also the global context will have an impact on our project.

We want to advance our discussion with the Cultural context. This is one of the contexts we found indispensable to discuss, seeing that clothing is eminently correlated with culture, which should be a key point not to neglect while realizing our design. As discussed above, culture reflects clothes, and considering the global characteristic of our project, we comprehended that one of our main burdens was to include different cultures in our design by realizing clothing items that have an essence of indigeneity. In other words, we do not want to overlook or diversify any culture and have the same impact or performance on every culture. On a further note, we believe that, by accessing different cultures across the globe, the reflective impact of the different cultures on our project would further ameliorate our product and widen our scope.

The following context is one of the most essential segments to be considered for the scope of our project, which is the Environmental context. We give the utmost importance to this topic, as one of our goals to achieve with our project is to vouch for an increase of sustainability of the environment. We are well aware that production in the clothing industry has a negative impact on an increase in carbon emissions and pollution. In fact, the studies show that 10% of humanity's carbon emissions are caused by the fashion industry single-handedly [13]. Another study shows that the fashion industry is the second most polluting industry, only after the oil industry [14]. Therefore, our group is aware of the magnitude of the problem, therefore giving the effort to incorporate the effect to increase the level of sustainability of the environment, in the fashion industry, by trying to enhance customer preferences. While touching upon the impact that our system will attempt to have on the environmental contexts, we also want to discuss possible impacts of our system or the implementation of our system on the environment. Previously, we have described that our project will include many models with different state-of-the-art architectures introduced by computer vision and machine learning communities. In our implementation, we want to train said models with data we found online. Here we have three possible concerns that reside in the environmental context. First, while training models in our system, we will consume a considerable amount of electricity, which is unfavorable in terms of sustainability of energy, and sustainability of the environment. In fact, an example from the machine learning community include the implementation of GPT-3, state-of-the-art technology used for text generation [15]; in the training of this model, 27,648 kWh were used over nine days, where the average electrical power usage of a US household is 10,649 kWh annually [16]. On the other hand, whilst production of electricity, carbon emissions usually play an important role. For instance, the electricity was produced in plants like thermal power stations; hence in our previous argument, we also touched upon the sustainability of the environment. Secondly, as explained previously, we want to make use of many datasets available online. This means that we have to consider the possible data center energy usage. This also applies to our third concern: our use of data centers to store different information of each of our users. We stated these concerns because we feel the responsibility of the environment and want to optimize the possible hazards that our project may have for the environment.

For the next aspect of our discussion, we want to consider the Economic context. While this context is not as momentous as the previous contexts we have discussed, we nevertheless want to contribute to our discussion with this context. As we are completing our project as a capstone project and trying to satisfy our course requirements, our priority is to assure the obligations that are required from us by our department. Hence any intention to attain economic benefit from our project is not a priority by our side. On the other hand, for the implementation for our project, some of the tools we want to make use of are requiring payment, which could be considered as an economic constraint that will apply on us. For instance, we want to make use of Google's Colab tool [17] for our high-performance computing tasks. Having in mind the volume of our tasks, we need to purchase a Colab Pro+ subscription of the service, which costs $49.99 per month. Similar constraints may apply on different tools, but for the sake of simplicity, and considering the meager relevance of the argument, we do not want to discuss further about economic contexts.

The last context we found worthy of discussing is the societal context. In this context, we have one main point to discuss, which is our users' data privacy. It should be well understood by now that we will be collecting a considerable amount of personal data, for better functionality of our project, of course, with the consent of our users. On the other hand, having the consent of our users should not mean that we will not be securing the sensitive information collected. We are well aware of the importance of the problem and will give the utmost importance to the security of user information.

4.6. Planning for New Knowledge and Learning Strategies

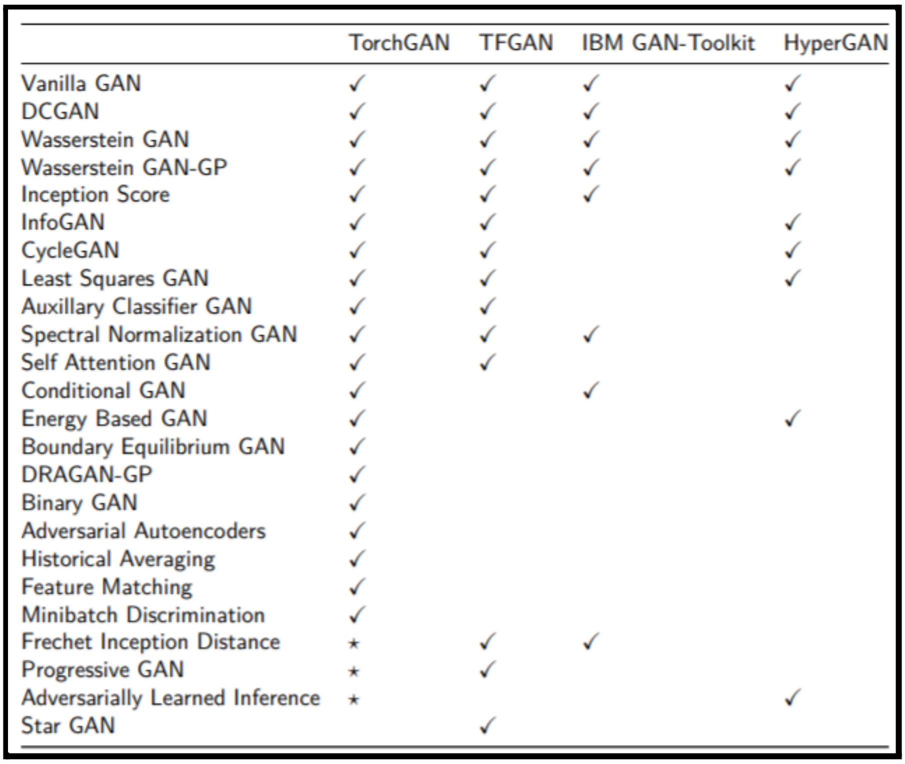
Although, as 4th year Computer Science students we would consider ourselves to have some experience relating to some issues, the problems we have chosen to deal with in this project are a combination of topics we have previous knowledge on though with need of refreshments and completely new topics with interest in them being present. As we have taken the course CS102 Algorithms and Programming II through which we have learnt Java and, as per course requirements, taken part in a group project which requires the use of Java in which most of the group members have developed an Android mobile application. As such, this kind of project is not new to the group members, although refreshments about mobile application development are needed; also there have been a surge of developments in mobile application technologies and frameworks when compared to the developments of four years ago - so one expectation of group members is to deliver the application by making use of these developments. In the course CS353 Database Systems, we have taken part in a project about database management where we were prompted to manage databases for a website of our creation. As for learnings about artificial intelligence and machine learning, our group members either have already taken courses related to machine learning and computer vision (such as CS464 Introduction to Machine Learning) or are taking them throughout the duration of this project; some of the group members have individual experiences pertaining to data science and machine learning as well, through involvement in individual projects, contests, or internships and such forms of experiences. We need to refresh our minds about the things we have learnt throughout the years, and be open to learning about various new technologies.

Main topics of focus for our group in this process will be as follows:

- Knowledge of Python and Java
    - This includes knowledge of frameworks and libraries relating to the technologies that we will use, such as TorchGAN framework, NumPy library
- Neural networks, deep learning, and how to apply deep learning to a model for predictive results
- GANs
- Knowledge of mobile application development, from interface creation to backend
- Knowledge of database structures
    - how to alter and manipulate databases, how to make them secure

Our one main source of learning for this project will be academic papers investigating the use of the technologies we are planning to use (such as personalized outfit generation through deep learning, virtual try-on aspect of the application with GAN, and such other machine learning oriented technologies if such need arises). One other source of learning will be the internet and the over-abundance of tutorials pertaining to different aspects of this application.

As for the development of a virtual try-on making use of GAN technology, we have found the paper for VITON (Virtual Try-On) [18] that is developed for trying out different clothes on 2D models, and we plan to analyse how the neural network for this specific aim was built. An improvement to the results of this paper, named as CP-VTON, will also be analysed to further deepen our understanding of how the networks were built and how they could be made to perform better [19]. These papers were accepted by CVPR 2018 and EECV 2018 respectively, so we can have faith in their results. As for how we will build these networks in a coded demonstration, we plan to utilize TorchGAN for which we will use the official documentation page for this framework for reference [20]. As for the reason we have decided on TorchGAN, the below table shows a comparison of usable GAN models for different frameworks. We have chosen TorchGAN because of its flexibility in this regard, as it allows for more GAN models to be used and changed if necessary by our team when our understanding of the framework gets better.



	TorchGAN	TFGAN	IBM GAN-Toolkit	HyperGAN
Vanilla GAN	✓	✓	✓	✓
DCGAN	✓	✓	✓	✓
Wasserstein GAN	✓	✓	✓	✓
Wasserstein GAN-GP	✓	✓	✓	✓
Inception Score	✓	✓	✓	
InfoGAN	✓	✓		✓
CycleGAN	✓	✓		✓
Least Squares GAN	✓	✓		✓
Auxillary Classifier GAN	✓	✓		
Spectral Normalization GAN	✓	✓	✓	
Self Attention GAN	✓	✓		
Conditional GAN	✓		✓	
Energy Based GAN	✓			✓
Boundary Equilibrium GAN	✓			
DRAGAN-GP	✓			
Binary GAN	✓			
Adversarial Autoencoders	✓			
Historical Averaging	✓			
Feature Matching	✓			
Minibatch Discrimination	✓			
Frechet Inception Distance	⋆	✓	✓	
Progressive GAN	⋆	✓		
Adversarially Learned Inference	⋆			✓
Star GAN		✓		

Tab. 4: Comparison of usable GAN models and existing GAN frameworks. [21]

As for the neural networks that we will build for the outfit recommendation aspect of our application, which requires a model that generates outfit recommendations for the user based on outfits submitted by the user and can change these recommendations over time as more outfits are submitted, we have looked into sources for models dealing with personalized outfit recommendation systems. Of these, the paper about how "Dida" developed to be used in the shopping website Alibaba, which is a personalized outfit generator for users of the website, was built will be of particular use for our project as it deals with a personalized outfit recommendation system using a neural network structure [22]. Another source of potential use is a tutorial article in which the problem of personalized outfit recommendation system was dealt with passing outfits to the model directly and its aim is to generate outfit outputs that "contain some aspects of the input but with serendipity to pleasantly surprise users," to quote [23].

The developer website [24] provided by Google services will be particularly useful, specifically the tutorials pertaining to the development of an application.

The last source of ours during this development process will be people: The expertise of group members, and constructive feedback of other experts. As we, as group members, have worked on different projects and have taken different approaches to the same problems throughout the years, we believe that communicating throughout the entirety of this project will be highly beneficial for the development to be smoother and easier. Until now, we have taken constructive feedback from our talks with our project advisor Prof. Dr. Halil Altay Güvenir and from our presentation with , and we are planning to communicate effectively with our advisor and we are open to being directed to different experts of various fields of study.

We will be learning during the process of development through our experimentations and about new learning topics if such need arises, and after the publication of the app to the end user, we will continue to work on various feedbacks about our application.

5. Glossary

TorchGAN: A PyTorch based Python framework for writing comprehensible codes for training and evaluation of GANs.

TF-GAN: A Tensorflow based Python framework for writing comprehensible codes for training and evaluation of GANs.

PyTorch: A Python library that provides automatic differentiation for building and training neural networks through structures which are n-dimensional Tensors (see “Tensorflow”) that can run on GPUs. [25]

Tensorflow: A Python open-source framework for building and training machine learning and artificial intelligence structures, or other predictive workloads. Tensorflow handles data sets that are in the form of nodes; the edges that connect those nodes create what are known as Tensors, representing matrices or multidimensional vectors. [26]

Numpy: NumPy is a Python library for scientific computing that provides a multi-dimensional array object, various derived objects such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Machine Learning: An algorithm that alters itself over time based on the data that it is exposed to.

Neural Network: Machine learning structures that aim to mimic the human brain that are comprised of connected nodes that have “weights” of their own which are altered as necessitated and pass on data to each other. [27]

Deep Learning: A subset of machine learning that defines the dependation of the algorithm onto a neural network structure with three or more layers. [28]

Computer Vision: Subset of artificial intelligence that defines a specialization in deriving meaningful information from visual inputs. [29]

GAN: An abbreviation for Generative Adversarial Network. This is a network model that is designed with two components: generator that creates data, and discriminator that determines the validity of the generated data. Its aim is to generate data that is similar to the training data [30].

GPU: An abbreviation for Graphics Processing Unit. A GPU is an electronic circuit that is designed specifically to speed up the production of graphical units in a frame buffer to provide outputs to a display unit. [31]

Git: Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

GitHub: GitHub is a provider of Internet hosting for software development and version control using Git.

6. References

[1] “Listening is everything,” *Spotify*. [Online]. Available: https://www.spotify.com/us/. [Accessed: 11-Oct-2021].

[2] *Google maps*. [Online]. Available: https://maps.google.com/. [Accessed: 11-Oct-2021].

[3] Wolfreys, J. (2009). In *Glossalalia: An alphabet of critical keywords* (pp. 358). essay, Edinburgh Univ. Press. [Online]. Available: https://books.google.com.tr/books?id=9b1UZF3pzWkC&printsec=frontcover&redir_esc=y#v=onepage&q=rhapso&f=false. [Accessed: 21-Oct-2021]

[4] M. McFall-Johnsen, “The fashion industry emits more carbon than international flights and maritime shipping combined. Here are the biggest ways it impacts the planet.,” *Business Insider*, 21-Oct-2019. [Online]. Available: https://www.businessinsider.com/fast-fashion-environmental-impact-pollution-emissions-waste-water-2019-10#in-europe-fashion-companies-went-from-an-average-offering-of-two-collections-per-year-in-2000-to-five-in-2011-3. [Accessed: 11-Oct-2021].

[5] C. Moser, S. Y. Schoenebeck, and P. Resnick, “Impulse buying,” *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.

[6] “Tryndbuy,” *Brand*. [Online]. Available: https://tryndbuy.com/. [Accessed: 15-Nov-2021].

[7] “YourCloset,” *Closet Organizer & Smart Fashion App for Android*. [Online]. Available: https://www.yourclosetapp.com/. [Accessed: 15-Nov-2021].

[8] “Hosted private apps - managed google play help,” *Google*. [Online]. Available: https://support.google.com/googleplay/work/answer/6145197?hl=en#zippy=%2Creliability-and-service. [Accessed: 25-Oct-2021].

[9] “Deploy apps to enterprises using Google Play : android developers,” *Android Developers*. [Online]. Available: https://developer.android.com/distribute/google-play/work. [Accessed: 25-Oct-2021].

[10] “General data privacy regulation.” https://eugdpr.org/. [Accessed: 6- Nov- 2019].

[11] “Google play terms of service.” https://play.google.com/about/play-terms/ index.html. [Accessed: 6- Nov- 2019].

[12] “Instagram,” *Instagram*. [Online]. Available: https://www.instagram.com. [Accessed: 15-Nov-2021]

[13] M. McFall-Johnsen, “Is fashion bad for the environment?,” *World Economic Forum*, 31-Jan-2020 [Online]. Available: https://www.weforum.org/agenda/2020/01/fashion-industry-carbon-unsustainable-environment-pollution/. [Accessed: 15-Nov-2021]

[14] G. Subramanian, “The true environmental cost of fashion: What can we do?,” *ET Retail*, 01-Apr-2019 [Online]. Available: https://retail.economictimes.indiatimes.com/re-tales/the-true-environmental-cost-of-fashion-what-can-we-do/3499. [Accessed: 15-Nov-2021]

[15] Contributors to Wikimedia projects, “GPT-3,” *Wikipedia*, 08-Nov-2021. [Online]. Available: https://en.wikipedia.org/wiki/GPT-3. [Accessed: 15-Nov-2021]

[16] M. Labbe, “Energy consumption of AI poses environmental problems,” *TechTarget*, 26-Aug-2021 [Online]. Available: https://searchenterpriseai.techtarget.com/feature/Energy-consumption-of-AI-poses-environmental-problems

[17] “Google Colaboratory.” [Online]. Available: https://colab.research.google.com/. [Accessed: 15-Nov-2021]

[18] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, “Viton: An image-based Virtual Try-On Network,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. [Online]. Available doi: 10.1109/cvpr.2018.00787

[19] B. Wang, H. Zheng, X. Liang, Y. Chen, L. Lin, and M. Yang, “Toward characteristic-preserving image-based virtual try-on network,” *Computer Vision – ECCV 2018*, pp. 607–623, 2018. [Online]. Available doi: 10.1007/978-3-030-01261-8_36

[20] “Torchgan¶,” torchgan. [Online]. Available: https://torchgan.readthedocs.io/. [Accessed: 29-Oct-2021].

[21] A. Pal and A. Das, “Torchgan: A flexible framework for gan training and evaluation,” *Journal of Open Source Software*, vol. 6, no. 66, p. 2606, 2021. [Online]. Available doi: 10.21105/joss.02606

[22] W. Chen, P. Huang, J. Xu, X. Guo, C. Guo, F. Sun, C. Li, A. Pfadler, H. Zhao, and B. Zhao, “Pog: Personalized outfit generation for fashion recommendation at Alibaba Ifashion,” *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019. [Online]. Available doi: 10.1145/3292500.3330652

[23] K. X. Thia, “Building a personalized real-time fashion collection recommender,” Medium, 01-Apr-2020. [Online]. Available: https://towardsdatascience.com/building-a-personalized-real-time-fashion-collection-recommender-22dc90c150cb. [Accessed: 29-Oct-2021].

[24] “Android developers,” *Android Developers*. [Online]. Available: https://developer.android.com/. [Accessed: 29-Oct-2021].

[25] “Pytorch,” *PyTorch*. [Online]. Available: https://pytorch.org/. [Accessed: 29-Oct-2021].

[26] “Tensorflow,” *TensorFlow*. [Online]. Available: https://www.tensorflow.org/. [Accessed: 29-Oct-2021].

[27] By: IBM Cloud Education, “What are neural networks?,” *IBM*. [Online]. Available: https://www.ibm.com/cloud/learn/neural-networks. [Accessed: 29-Oct-2021].

[28] By: IBM Cloud Education, “What is deep learning?,” *IBM*. [Online]. Available: https://www.ibm.com/cloud/learn/deep-learning. [Accessed: 29-Oct-2021].

[29] By: IBM Cloud Education, “What is computer vision?,” *IBM*. [Online]. Available: https://www.ibm.com/cloud/learn/computer-vision. [Accessed: 29-Oct-2021].

[30] “Machine learning glossary | google developers,” *Google*. [Online]. Available: https://developers.google.com/machine-learning/glossary?hl=en#generative-adversarial-network-gan. [Accessed: 25-Oct-2021].

[31] “Graphics Processing Unit,” *Wikipedia*, 21-Oct-2021. [Online]. Available: https://en.wikipedia.org/wiki/Graphics_processing_unit. [Accessed: 25-Oct-2021].