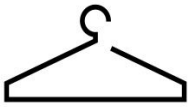
Bilkent University

Department of Computer Engineering

Senior Design Project

Project name: Dressy



Dressy.

*Your own virtual wardrobe

High Level Design Report

Asuman Aydın, Doğukan Köse, Fatih Çakır, Muhammed Musab Okşaş, Şeyma Aybüke Ertekin

Supervisor: Uğur Güdükbay

Website Link: https://dress-y.github.io/

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 **3**
1.1. Purpose of the System 4
1.2. Design Goals 4
1.2.1 Usability 4
1.2.2 Maintainability 4
1.2.3 Privacy 4
1.2.4 Reliability 5
1.3. Definitions, Acronyms, and Abbreviations 5
1.4. Overview 5

2. Current Software Architecture **6**
2.1. Human Pose Estimation - DensePose 6

3. Proposed Software Architecture **7**
3.1. Overview 7
3.2. Subsystem Decomposition 8
3.3. Hardware/Software Mapping 9
3.4. Persistent Data Management 10
3.5. Access, Control, and Security 11
3.6. Global Software Control 11
3.7. Boundary Conditions 12

4. Subsystem Services **14**
4.1. Presentation Layer 15
4.2. Application Layer 16
4.3. Data Layer 17

5. Consideration of Various Factors in Engineering Design **18**

6. Teamwork Details **19**
6.1 Contributing and functioning effectively on the team 19
6.2 Helping creating a collaborative and inclusive environment 19
6.3 Taking lead role and sharing leadership on the team 20

7. References **21**

1. Introduction

The preference of online shopping is a huge trend in today's digital world because of the lack of time, the convenience of online shopping, the advantage to have access to other users' experiences, the existence of cost choices and richness in the product variety. By the beginning of 2020, 69% of Americans had an online shopping experience. Also, 25% of them were regular online shoppers who bought at least one item each month. Statistics also demonstrate that 47% of online shoppers buy clothing items. The online shopping trend is not limited to America. If we look at the worldwide statistics, the rate of online shopping in 2018 is 47.3%. While 72% of women had preferred to shop items online while 40% of these items, the largest percentage, were clothing. With the effect of the coronavirus, this preference has become even clearer. People have begun to prefer to buy most of their needs online rather than visiting crowded shopping malls and trying out clothes that they do not know who had worn previously. [1] However, shopping for clothes online has a downside. Shoppers do not have the advantage of trying clothes and therefore, it is difficult to make good choices in terms of looks and size of clothing.

Figure 1. Statistics on presentation of increase on online shopping after Covid-19

The purpose of our senior design project is to assist people who choose to buy their clothes online. We want to create an application that provides a virtual fitting room for them. Thereby, they can try the clothes they have chosen and see the clothes on themselves.

With this project specifications report, we aim to explain the description of our application, current systems, requirements, diagrams about our senior project, Dressy. Also, we will provide mockups of screens of our app.

1.1. Purpose of the System

Dressy is designed to bring people together with shopping malls online, using today's technological developments. Thanks to the application, it will provide a great opportunity in quarantine conditions and will become an entertainment door for people. Its biggest aim is to prevent confusion in the store and to enable people to take their clothes from their homes with the benefits of technology.

1.2. Design Goals

1.2.1 Usability

The usability of the features that we stated to be in the application is one of the most important design goals for Dressy. The user base we intend to use can be limited to anyone who can shop. We can understand that this limitation is also a wide limit for the user for us. In this respect, usability is both a great goal and a big problem. To apply this feature to the design, it can be done by field research and customizing the features according to the user groups. In this respect, the interface of the application and its in-application usability will be designed simply and appealingly.

1.2.2 Maintainability

Maintainability should be added as one of the aims for the continuation of the project. One of the most important purposes in the design of the Dressy application is on this. To achieve this maintainability, application design should be advanced over an object both on the frontend and on the backend. This object-resistant design will give us both the freedom to make changes and much easier coping with the difficulties encountered during development.

1.2.3 Privacy

The privacy policy is one of the most important features that technology adds to applications. It can be attributed as an important structure rather than a feature. This building is also among the design purposes, which will have a great place in Dressy. For this purpose, certain structures will be used on the back end of the application to store the visual and personal information of the people.

1.2.4 Reliability

Reliability is important for Dressy, which is an image-oriented application, from the following point of view: There should be a solid structure in order for a person to easily and smoothly try the outfit he wants to try in the studio. It is also among the design principles that this structure can progress with instant data without causing any problems.

1.3. Definitions, Acronyms, and Abbreviations

IaaS	Infrastructure As Service with cloud data systems. It can be referred to as a transfer and store system for application. [2]
REST API	Application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. [3]
MySQL	Type of database management system.
DoS	Attack meant to shut down a machine or network, making it inaccessible to its intended users. [4]

Figure 2. Definitions, Acronyms, Abbreviations

1.4. Overview

Dressy is a mobile application designed for people of all ages who want to try and shop online. The purpose of the application is to see the booth, trial, and various categorical products provided by the stores. The part where this is presented to the user is to instantly create a model from the user's body measurements and image and continue shopping by trying the clothes on the customer.It has also been determined and proven through statistics that people mostly tend to shop online as a result of quarantines that are valid worldwide with the coronavirus. This orientation creates a great opportunity by creating a space for applications such as Dressy, and most often by allowing shopping from within the home.

The user can use Dressy as long as he/she carries a mobile phone with him and his / her internet is on. It is important that this application has a simple interface and usability appealing to all ages.

Dressy, which will be supported by android studio on the front end, will provide communication by reflecting and receiving images of the user to the application with the latest ARcore technologies provided by Google APIs on the backend. In addition to taking a snapshot, modeling will be made with the physical features of the user via Unity or Blender,

and an opportunity will be provided to the user at the point where he cannot provide an image at any time by providing an online model.

2. Current Software Architecture

2.1. Human Pose Estimation - DensePose

At the first stage of our senior design project, we have focused on processing and getting density maps from real time images from a camera, and creating a 3D model of a human body. In order to increase the usability of our project, we have made a decision to use RGB cameras for our application in the analysis phase. However, it is not an easy task with RGB cameras to get density and create a model, because they do not have a sense of depth. After searching which tool we can use to fulfill this goal, we have decided to use a project from Facebook AI Research, DensePose. It helps us to map a 3D human body from a 2D RGB image using deep learning techniques.

DensePose enables us to get the coordinates of the human body. This speciality is important for us, because in order to fit the clothes on a human body, we need to work with these coordinates and we should be able to map the texture of the clothes to the human body. DensePose allows us to process multiple frames in a second. As a result, we are able to take the 3D model, therefore, the position of a human body in real time. This feature helps us to create a real virtual room effect, because the model will move as the human moves and we will be able to show the user how the clothes look as they move.

The following figure shows 3D human body mapping from an RGB image. These 3 models are visuali part index, U, and, V coordinates of the created model.

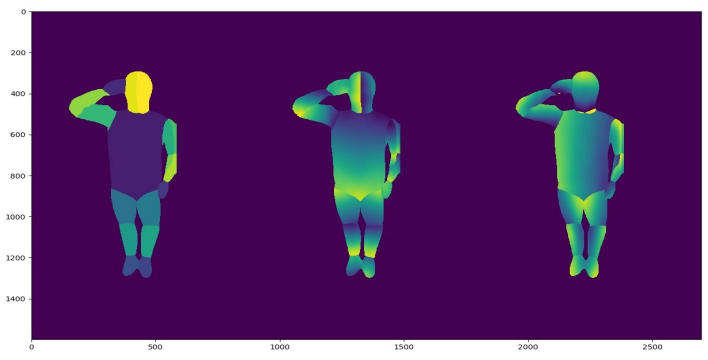


Figure 3. DensePose Output

3. Proposed Software Architecture

3.1. Overview

Dressy is designed as an application that people can easily access and use. The software architecture was built by combining the advantages of different platforms in this direction. For example, while customers can easily use and interact with the application using their phones, the main computing part of the application takes place on a remote server, which has much better computing power than the phone, without users' knowledge. In addition, the data that can be used later are kept in the database server completely separate from these platforms. In this way, it is aimed to combine the strengths of different platforms in an optimum software architecture. As it will be explained later, we used 3-tier architectural style, which consists of presentation, application, and data layers in order to combine our 3 platforms mentioned to presentation, application and data layers.

Another strength of this distinction is to share our human power in the best way and develop a fast and effective software process thanks to separation of concerns between these platforms. Additionally, we can address problems and provide solutions faster with this approach. In the subsequent sections, the proposed software architecture is explained in more detail.

3.2. Subsystem Decomposition

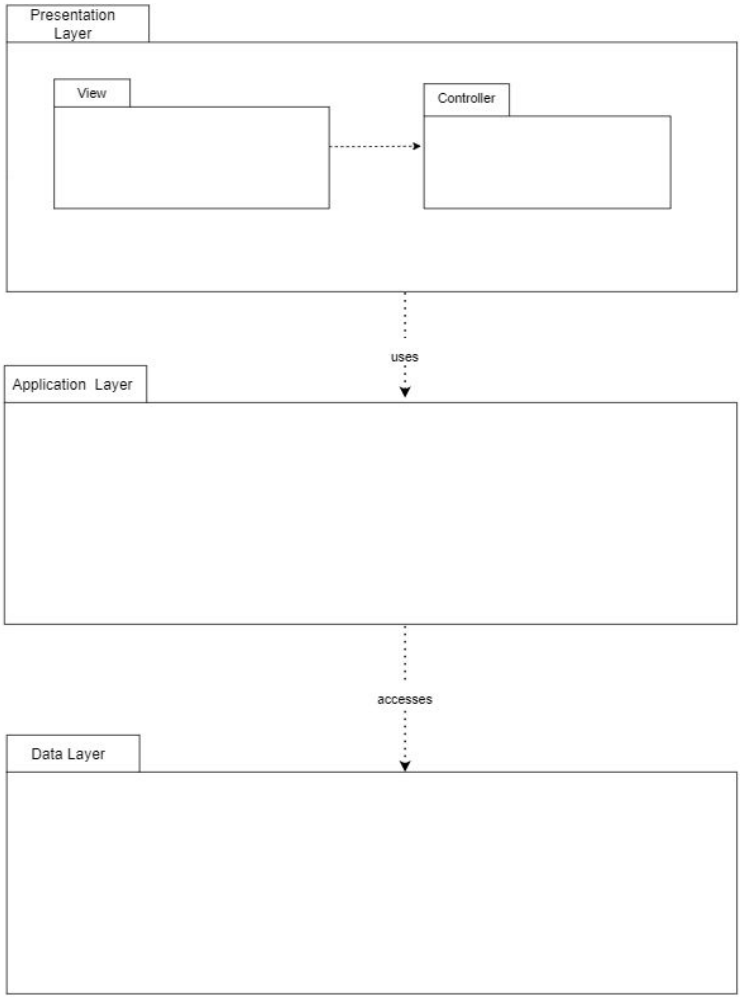


Figure 4. Subsystem Decomposition

Dressy application has a 3-tier architectural style, which consists of presentation, application, and data layers. Figure shows the subsystem decomposition of our application. This architectural design choice is made, because Dressy is a mobile application, and this architectural style is very suitable to decrease cohesion between subsystems for mobile applications. This approach helps to have more flexibility to develop and update each layer with minimum dependency to other layers. Decoupling is increased by further separating the presentation layer to controller and view subsystems, which can be considered as an example of model, view, controller architectural style, but model is presented by the data saved in the database.

Presentation layer is responsible for controlling any user interaction. This layer is basically Dressy's front end layer and runs on the client's mobile device. Presentation layer consists of two subsystems, which are View and Controller subsystems. View subsystem contains the graphical user interface of the mobile application and displays information obtained from and processed in the Application layer. Any user request is received through this subsystem and transferred to the Controller subsystem. Controller subsystem of the presentation layer is responsible for dealing with user requests with necessary information from the Application layer, providing essential communication to the Application layer, and managing information flow.

Application layer forms the core structure of the Dressy application. Any video processing, and database access is realized through this layer. Information coming from the client side, from the presentation layer, is processed in this layer and necessary information is saved to the database. Application layer is responsible for fulfilling any request from the client side, which may be a request for processed information or information saved in the database.

Data layer is responsible for efficiently saving any application related information by managing database interactions. Dressy mainly contains two main segments to save, user information and clothes information. Application layer can access and update this information through the Data layer.

3.3. Hardware/Software Mapping

Dressy will be an application for mobile systems and tablets. Our plan is for Dressy to work both on Android and iOS. Users of the application can interact with our services from their mobile phones. They can see our available clothes, manage their profiles and use virtual fitting rooms via their mobile systems. Additionally, new 3D clothes can be uploaded through mobile systems. Cameras of the mobile systems will be required to use in order to get body images of the users and images of the clothes.

We will not use mobile systems to process these images since it would require a lot of resources. We will use a remote system for this task. We will use Amazon Web Services Linux servers as infrastructure as a service (IaaS). This system will include our weights for the computation models and use GPU to apply the model to the images. After getting the images from cameras of the mobile systems, this data will be sent to our Amazon services through REST APIs. After getting the data the 3D images will be processed in the Amazon service and the results will be sent to the mobile system to be presented to the users. Amazon services also handles other requests as well. For example, authentication service is

also included in the Amazon server. Additionally, Amazon server plays the bridge role between the mobile systems and our databases.

There will be a database system to keep information related to the application. Databases will keep user data, 3D cloth data, 3D user body images and other necessary information. Database system will be connected to our Amazon server and the Amazon server will delegate the communication between the mobile system and database system.

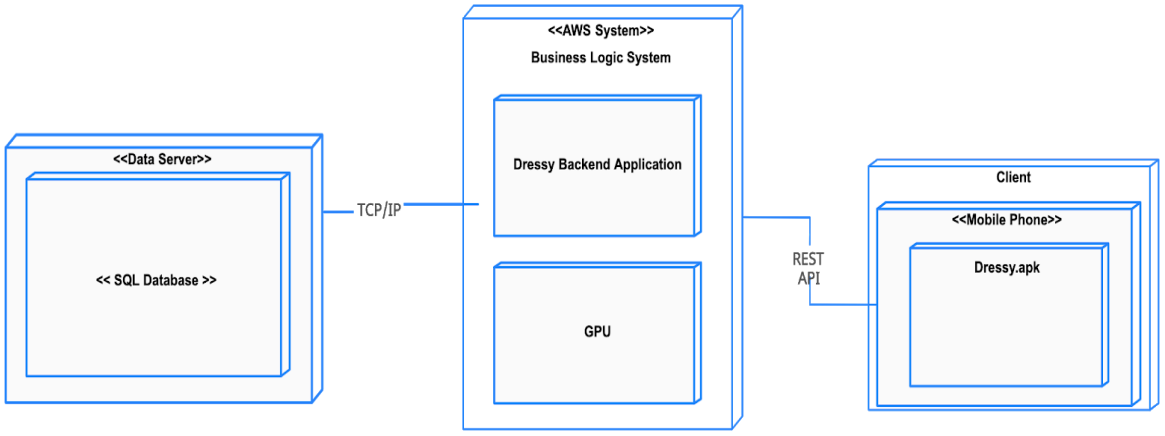


Figure 5. Hardware / Software Mapping

3.4. Persistent Data Management

In our application there is some crucial information that needs to be stored and persisted reliably. We can think of user data, 3D clothes data, 3D body data, model weights as information that needs to be persistent. All of these data needs to be available as much as possible. Additionally we need to provide secure access to this data so their integrity would not be damaged.

In order to reach high reliability while maintaining fast access we plan to use a Database Server with MySQL database. By using MySQL we ensure database integrity. We also plan to provide indexing to increase the speed of our Database system for specific data. Other than these crucial data, we plan to keep the user actions by using event tracking with Google Tag Manager. This data later will help us to analyze user actions so we can improve our application.

Other data that doesn't need to be persistent will be kept in local storage and caches. We are going to use client side caching for improving the usability of our application.

3.5. Access, Control, and Security

Dressy system includes different use cases for different users and also it is built on a multi tier system. In a system such as Dressy there should be mechanism and design to assure secure access and control over the application. In our application it is essential to restrict user roles to specific functionality in order to ensure integrity. As we stated earlier, we will keep the data for users, user actions, 3D clothes and 3D body images. Access control is particularly important in our setting since exposing these crucial data to parties that should have no access to them would put our application into a difficult position in case of reliability. We should be accountable to our users for protecting their data.

We have mainly 3 different user roles. These roles are admins, buyer users and seller users. Admins have the most power considering the roles. They can access most of the data to regulate our system. They can modify user information and read any data. The only concern is the passwords of users and since they are kept in hashed format, they are not exposed to the outside. Buyer users will have access to see our cloth database, comments and ratings for clothes, their own profiles and they can use our virtual fitting room. They can only have access to update their profile and add new comments. Both of these actions need to be approved by admins. Seller users are similar to buyer users but, in addition, they can add new 3D clothes to sell into their profiles.

3.6. Global Software Control

In this section, we explain the overall system control process on a global scale. We give information about how the requests are initiated and after these requests how the subsystems synchronize. Finally, we focus on some concurrency issues. For this section, we consider two important actors which are Client for mobile systems and Server for Amazon Web Server with the intent of providing a better abstraction. In brief, the software operates as requests to the server(s) are received from the client(s), these requests are then processed and the action taken in response is sent back to the client(s).

There are three main phases we focus:

1. Request Phase: This phase begins with the initiation of the requests on the client side. Data is prepared once the user is done preparing body image input from the camera of the mobile system. After the data is ready, the user will start the request phase. This user-action triggers a request which sends the body image data that the user has prepared to the server as an input to be processed by the generative models. A naturally arising problem here is that the server(s) may be overwhelmed by a flood of requests, this flood may be a result of an unexpected increase in the number of users or a potential Denial of Service (DoS) attack. As a precaution for such scenarios, we will implement rate-limiting in our API, which will keep the client-side waiting until the server(s) are available.

2. Processing Phase: After a request arrives at the respective server port, the related module in our script should run and apply the transformation to the 3D body model or to the 3D clothes from the received input. This is the phase that delays the response the most. It could take up to minutes. Hence, it requires serious work to be parallelized. Due to the cost limitations, we will not be able to accommodate a large number of servers. However, if the user base of our application increases significantly, we need to purchase new servers to provide a better service to our users. In such a case a Load Balancing Algorithm should be implemented to fairly distribute the work among different units, this also requires incorporating a Load Balancer hardware to the server-side.

3. Response Phase: In the response phase, we basically send the model outputs back to the clients from the ports where the respective requests have been received. To introduce concurrency to this phase and also the request phase we aim to find an optimal port configuration such that the delay experienced in the client-side is minimized.

3.7. Boundary Conditions

Dressy includes three boundary conditions. These boundary conditions are: initialization, termination and encounter a failure in the system. These boundary conditions are explained in the following subsections below.

Initialization

First of all users should have downloaded the application from either Google Play Store or as an apk and install it to their mobile system. Secondly, users should accept the permissions to access the features of our application. Having internet connection is essential for our

system since computation would happen at our Amazon services and the data is sent from the server as well. If users don't have internet access they can only access to the data from their local cache. Users should have an account to use the virtual fitting room feature of our application, make comments to the clothes and rate the clothes.

Termination

When the user turns off the application, the application is terminated on their phone. Any running process on the server related to this user will also be terminated. If the user doesn't want to use the local storage cache, the cache will also be deleted. If the user is not logged out, then their session would stay open, meaning that they don't have to sign in again. However, if they choose to log out, they have to sign in again in order to use some features of our system.

Failure

Failure can occur in three cases. First case can be seen while getting input from the camera. If the battery drops under a certain level, the camera stops working. Users will be informed that the camera has been closed due to a low battery and if they connect the battery to the power outage, the process will continue. Second case of the failure is seen when the internet connection is lost while the mobile system is in communication with Amazon server. In such a case, the system will try to recover by sending a request every 30 second periods for 5 minutes. If the internet connection is back and the situation is recoverable, the process will continue. Furthermore, users will be notified about this situation and an information message will prompt saying that the "Network connection is lost". Third case of the failure is seen if one of the modules stops working due to a bug in code or a problem in the server. In this case, users will be informed that the server is not available at the moment and they will be notified when the server is ready to use again.

4. Subsystem Services

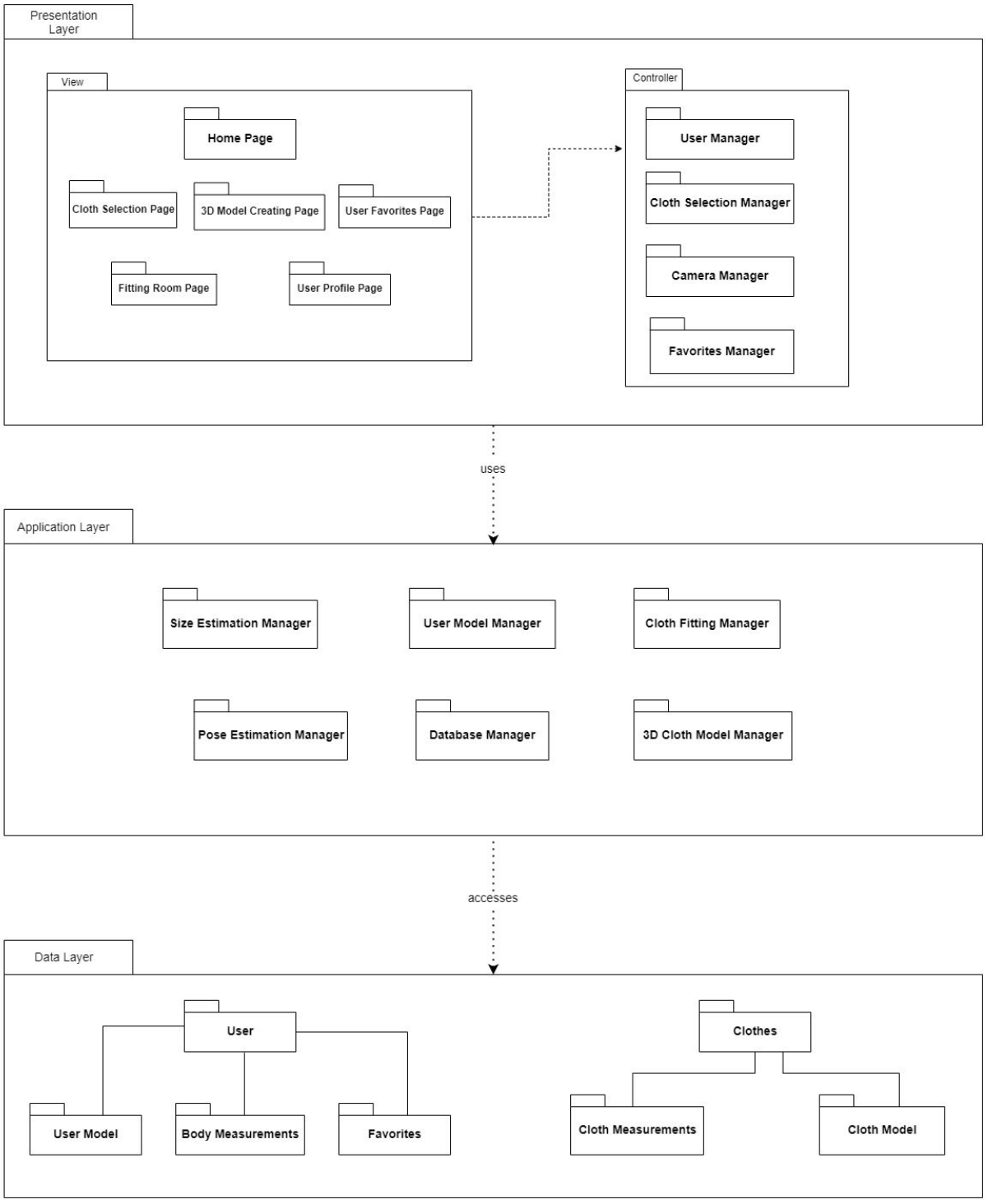


Figure 6. Subsystem Services

4.1. Presentation Layer

The Presentation Layer manages the interaction between the user and main logic part of the application. There are two subsystems in the Presentation Layer that are View and Controller.

4.1.1 View Subsystem

View Subsystem consists of views of pages from the user interface.There are 5 main views in this subsystem.

Cloth Selection Page: This view is for displaying clothes that can be viewed and be added to favorites by the user. Clothes will be displayed here with its specifications like size, color, gender, type etc.

3D Model Creating Page: This view is for displaying instructions and creation area for the 3D model of the user. This view is in communication with the user manager and camera manager to create a 3D body model of the user.

User Favourites Page: This view is for displaying favourite clothes of the user. There will be clothes with details as a list view. Each clothes can be removed by a button or tried on the 3D model by another button.

Fitting Room Page: This view is for displaying the 3D model with the selected cloth(es). Clothes will fit the model body realistically.

User Profile Page: This view is for displaying profile details of the user. The user will be able to view and change the profile settings by User Profile Page.

4.1.2 Controller Subsystem

This subsystem is the Controller layer of our application. Controller Subsystem contains controllers to communicate between UI and the Application Layer. It consists of 4 managers.

User Manager: Each user has a different body model, size, customizable preferences and favorites clothes. This class is used to get the user's own data from the Data Layer and

provide only required data to the View layer. Subsystems in the View Layer will be able to get its convenient data by this manager, not from directly the Data Layer.

Cloth Selection Manager: This class is for providing selectable clothes and their details to the UI from the clothes database. Also, it will send selected clothes to Database Manager to add selected clothes to the favourite clothes list of the user.

Camera Manager: The application needs to get photos from the user to create his/her model. Therefore, it needs a camera controller. This class is responsible for running the camera and transferring taken data to the Application Layer.

Favourites Manager: The application can suggest size and clothes according to favourites. This class is responsible for creating a link between the UI and the Application Layer. This class will communicate with Database Manager and Size Estimation Manager according to the favourite clothes list.

4.2. Application Layer

Application Layer manages the main logic part of the application. It consists of 6 manager modules.

User Model Manager: User Model Manager is needed for extracting the 3D model of the user. It uses machine learning to extract a model of the user. Then, it saves the user model into the database through the Database Manager module for later usage.

3D Cloth Model Manager: 3D Cloth Model Manager is similar to User Model Manager, but it is used for creating 3D models of clothes. In this module, again machine learning will be used to extract a model of cloth. Due to this model, users can extract and load clothes to the application.

Size Estimation Manager: Size estimation module will be used for advising proper cloth size for user's body size. To estimate proper size, it will use body measurements of the user and measurements of the cloth.

Cloth Fitting Manager: The Cloth fitting module will provide the machine learning input-output exchange that is required to fit the selected clothes on the user's model.The module will use a user model and cloth models.

Pose Estimation Manager: Pose estimation module is needed for moving fitted cloth with the user's movements simultaneously. If the user tries an outfit on the 3D model, this module will map the user's movements on the 3D model so that the cloth can move with the user's model.

Database Manager: Database module is needed for saving to the database or loading from the database. It is the bridge module between Data Layer and Application Layer.

4.3. Data Layer

Data Layer manages database interactions. It communicates with the Application layer to service the requested data. Data layer consists of two main storage related segments: user and clothes information.

User: User id is needed to save in the database in order to be able to access favorites or any model related information. User preferences can also be associated with this component.

User Model: After a model of the user is created, this model should be saved in the database in order to use later to fit the clothes. Otherwise, a user will have to create a user model each time s/he uses the application.

Body Measurements: User’s body measurements should be kept in the database in order to be able to make size recommendations for clothes.

Favorites: Users should be able to add/remove any clothes to their favorites. Each user has its own favorite list and it should be kept in the database in order to satisfy users’ request to view favorites.

Clothes: Clothes’ information should be kept in the database in order to provide selection freedom to users from provided clothes and fit them on their own models.

Cloth Measurements: These measurements are necessary to provide size recommendations for the user.

Cloth Model: Cloth model contains necessary information to fit the clothes on a person, which is the texture of a 3D model.

5. Consideration of Various Factors in Engineering Design

Corona virus has taken over the world in its hands. Due to restrictions maybe even carantina situations pushed people to adapt home life even more than before. People need to stay inside the house for many days and the needs are provided by the online markets, shops and websites such as Hepsiburada, Yemeksepeti. In the consideration of these matters, the idea of online shopping has merged with online product trials in our mind. It was a lighting in our head that we need this, we need some product which will serve people as outside world experience. So in the consideration of many factors, our application might be considered as a life saver. It does not affect social interactions since we provide an interface for interaction in the application. The environmental factors are more on the positive side of the effect on the application.

Figure 7 has been given to analyze the factors and effects.

	Effect level	**Effect**
Public health	10/10	Coronavirus affects people's both psychological and physical health. Due to this effect, our application will be a great provider and relief to people.
Public safety	5/10	People need to share their body image and maybe share their preferences. Even though the information will be protected by the network security, the leaks and the attack may cause some damage to people's privacy. So we may need to restrict some of the features due to these privacy issues.
Public welfare	0/10	This factor has no effect on the project.
Global factors	0/10	Globally, getting dressed is a general thing for every country. The

		application can be applied in any place. So global factors are not major effects.
Cultural factors	7/10	People adopted online shopping but virtual try-on might be not accepted by the elderly or conservative people.
Social factors	10/10	Current social situations are the effect which feeds the idea of the application.

Figure 7. Factors that can affect analysis and design

6. Teamwork Details

6.1 Contributing and functioning effectively on the team

	Musab	Aybüke	Fatih	Doğukan	Asuman
Contributions	Leading the team and managing issues faced during the development and design process.	Planning meetings and making system analysis on development about all designs.	Full Stack development on the coding side. For first design part, he and Doğukan made huge progress on coding of designed architecture	Full Stack development on the coding side. For first design part, he and Fatih made huge progress on coding of designed architecture	Design of general user interface and analysis on efficiency and usability of application Dressy. Field search on usage of such applications.
Functionalities	- System Architecture Designer - Team Leader - Product Manager	- System Analyzer - HR/Process Manager - Tester	-Fullstack Developer -System Developer	-Fullstack Developer -System Developer	-UI Designer/Developer -Full stack Developer

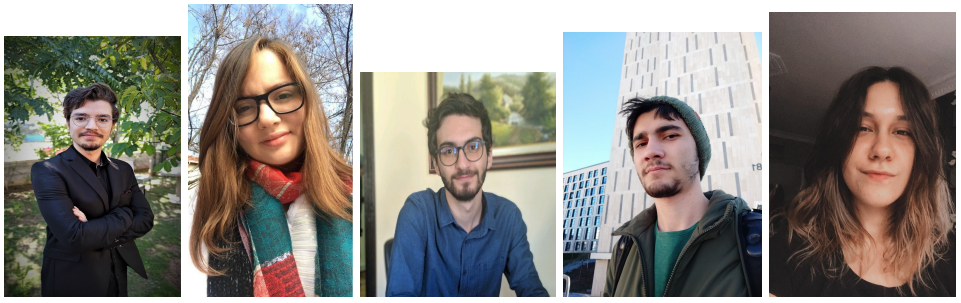
Figure 8. Task Sharing in The Team

6.2 Helping creating a collaborative and inclusive environment

Since the idea of developing the Dressy application was an exciting goal and idea within the group, the working environment was shaped accordingly. On the one hand, we have the

advantage of being in a group that we cooperate with. Weekly meetings are held to ensure that the roles allocated within the team are fulfilled by each individual. In these meetings, next week's plan and past week's criticism are made. In addition to these, the team leader Musab handles the urgent work with an emergency plan with a short meeting, creating a relaxing and more understandable working environment for the group. The protection of the general structure also develops and grows over time, depending on the ambition we have for the members of the group and what to do for the Dressy application. The meetings are held over the Discord application, and the meetings with our supervisor Uğur Güdükbay are conducted over Google Meet.

6.3 Taking lead role and sharing leadership on the team



	Musab	Aybüke	Fatih	Doğukan	Asuman
Roles	Team Leader	System Developer/Tester	Full-Stack Developer	Full-stack Developer	UI Designer/Developer

Figure 9. Roles and Pictures of Team Members

The Dressy team, led by Musab and Aybüke in general, is also a team where each individual has a say. In this respect, we reflect the importance of team members to the team spirit as well as the trust in both Musab and Aybüke.

7. References

[1] "COVID-19 has changed online shopping forever, survey shows," *Unctad.org*. [Online]. Available: https://unctad.org/news/covid-19-has-changed-online-shopping-forever-survey-shows. [Accessed: 21-Nov-2020].

[2] Cloud & Data Center Platform: Sparkle. (n.d.). Retrieved December 27, 2020, from https://www.tisparkle.com/our-platform/cloud-data-center-platform

[3] What is a REST API? (n.d.). Retrieved December 27, 2020, from https://www.redhat.com/en/topics/api/what-is-a-rest-api

[4] Read, 3. (n.d.). What is a denial of service attack (DoS) ? Retrieved December 27, 2020, from https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos