DOKUZ EYLÜL UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF COMPUTER ENGINEERING

3D CLOTHING TRYING AND COMBINE SUGGESTION APPLICATION

by
2021510091 Arzu YALDIZ

Advisor
Prof. Dr. Derya BİRANT

NOVEMBER, 2024
İZMİR

3D CLOTHING TRYING AND COMBINE SUGGESTION APPLICATION

A Thesis Submitted to the
Dokuz Eylül University, Department of Computer Engineering
In Partial Fulfillment of the Requirements for the Degree of B.Sc.

By
Arzu YALDIZ

Advisor
Prof. Dr. Derya BİRANT

NOVEMBER, 2024
İZMİR

Abstract

In today's fast-paced world, choosing daily outfits can often feel like a time-consuming and exhausting task, especially for individuals with physical limitations or busy lifestyles. This thesis presents the design and development of a mobile application that allows users to virtually try on clothes and receive outfit combination suggestions using augmented reality (AR) and machine learning technologies. The application enables users to upload photos of their clothing, which are then transformed into 3D models and mapped onto the user's body using real-time body tracking. Personalized outfit suggestions are generated based on the user's wardrobe, preferences, and contextual data such as weather conditions.

The project was inspired by a real-life event when one of the team members faced difficulty getting dressed due to a temporary disability. This experience shaped the motivation behind creating a tool that is accessible, practical, and inclusive for all users. By eliminating the physical need to try on clothes, the application aims to reduce decision fatigue, enhance fashion creativity, and help users rediscover forgotten pieces in their wardrobes. The system is implemented using Unity for AR capabilities, Python and TensorFlow for AI-based recommendation logic, and Android Studio for the front end. This work demonstrates that technology can effectively improve everyday routines and offer inclusive solutions in the fashion and clothing sector.

CONTENT

LİST OF FİGURE

CHAPTER ONE

1. INTRODUCTION

One of the biggest challenges we face today is the lack of time. Our time is valuable, and we need to spend it efficiently. Many people don't have the time to choose outfits, and some may struggle to change their clothes once dressed due to disability or age. With this project, we aim to create a mobile application that allows users to select their outfits virtually before actually trying them on. Users can save their favourite clothing combinations for faster outfit selection in the future. Additionally, our app will suggest possible clothing combinations to help users dress with ease and style.

1.1. Background Information

Our initial project idea was to create a touchscreen mirror that would allow users to try on clothes virtually without actually wearing them. However, after some discussions, we realized that a touchscreen mirror would be costly to produce and would require a larger team. Considering user accessibility and purchasing power, we decided that a mobile application would be a more practical solution, reaching a broader audience. This application will use augmented reality (AR) technology, enabling users to upload photos of their clothes, which the app will then convert into 3D models. Users can then "wear" these 3D models to create outfit combinations directly on their mobile devices.

At the time of this progress report, we have successfully developed body-tracking functionality, which will allow the application to track users and overlay clothing onto them virtually. With body tracking complete, we have begun researching methods for generating a 3D model of clothing from a single image. To complete this project, we need a solid understanding of Unity, as our application will be built on this platform. Machine learning is also essential, as it will enable the application to suggest personalized outfit combinations for each user.

1.2. Problem Definition

The main problem we aim to solve is the lack of time that many people face when trying to create outfit combinations, especially under time constraints. We know that some people buy the same outfit for not even thinking about what to wear, the most famous of these persons are Steve Jobs, Mark Zuckerberg, etc. We want to give this opportunity with fashion, not to think about what to wear the next day or for any event. We believe that our project will increase human life comfort a lot. In today's fast-paced world, people often struggle to find the time to plan their outfits, whether it's for work, social events, or daily activities. This can lead to frustration and added stress, particularly when individuals are in a rush. Our mobile application seeks to alleviate this issue by allowing users to quickly and efficiently create outfit combinations without the need to physically try on clothes.

Additionally, we want to help users rediscover clothing items they may have forgotten about. It’s easy for clothes to get buried at the back of the wardrobe, and many people often forget what they have, leading to the over-purchase of similar items. Our app will allow users to digitally explore their closets, reminding them of the items they own and helping them make the most of what they already have.

Another important challenge we aim to address is the difficulty individuals with disabilities face when changing clothes. For those with limited mobility or physical constraints, the process of changing outfits can be exhausting and time-consuming. Our solution will minimize the need for frequent physical changes by offering a virtual try-on feature, allowing users to visualize and experiment with different clothing combinations without the effort of changing clothes multiple times. This can not only save time and energy but also enhance the daily lives of people with disabilities by making outfit selection more accessible and less physically demanding. Ultimately, our goal is to make clothing selection easier, faster, and more inclusive for all users, regardless of their time constraints or physical limitations.

1.3. Motivation

After one of our team members had an accident, she found herself unable to get out of bed for an extended time. During her recovery, she encountered significant challenges when it came to something as routine as trying on outfits. The physical effort required to change clothes or try on different combinations became overwhelming and exhausting, making it difficult for her to plan her wardrobe or feel confident in her outfit choices. This personal experience, marked by frustration and a sense of limitation, led to the realization of a broader problem faced by many people with similar physical challenges.

Motivated by her struggles, she was inspired to create a solution that could help not only herself but also others who faced comparable difficulties. It was during this time of recovery that she began drafting the first version of this project, with the vision of developing an application that would allow users to select and try on clothes virtually. The goal was to eliminate the need for physical exertion in the outfit selection process, making it easier for individuals with mobility issues, disabilities, or those undergoing recovery to explore their wardrobe without the physical demand. What began as a personal need evolved into a project with the potential to make a meaningful difference for people facing similar challenges, ultimately shaping the direction of our work on this project.

1.4. Goal/Contribution

The goal of this project is to create an accessible and efficient solution for individuals who face difficulties in selecting and trying on outfits due to physical limitations, time constraints, or other challenges. Our motivation stems from the belief that everyone, regardless of their physical abilities or circumstances, deserves the opportunity to feel confident and comfortable in their clothing choices. We aim to empower users by providing a mobile application that allows them to virtually try on clothes, discover new outfit combinations, and make the most of their wardrobe—all without the need for physical exertion.

By leveraging augmented reality and machine learning, we strive to offer a personalized, time-saving experience that caters to the unique needs of each user. Whether it's helping individuals with disabilities, those recovering from an injury, or busy individuals with limited time, our goal is to reduce the barriers to outfit selection and create an inclusive platform that enhances users' daily lives. Ultimately, this project is driven by a deep commitment to improving accessibility, boosting confidence, and making the process of getting dressed easier and more enjoyable for all users.

1.5. Project Scope

The objective of this project is to develop a mobile application that helps users virtually try on clothes and create outfit combinations without the need for physical effort. The application will cater to individuals with mobility challenges, disabilities, or limited time, allowing them to explore their wardrobe, save clothing combinations for future use, and receive personalized outfit suggestions. The app will utilize augmented reality (AR) technology to enable users to visualize clothing on a 3D model and create seamless outfit combinations.

1.5.1 Features and Functionality

The application allows users to create and customize their profiles, which includes uploading photos of their wardrobe items, such as clothes, shoes, and accessories. This makes it easy for users to access their clothing collection within the app. Using augmented reality (AR), the app enables users to virtually try on clothes by overlaying 3D models of garments onto their personalized body model, offering an accurate representation of how the clothes would fit and look.

Users can mix and match different clothing pieces to create full outfit combinations, which they can save for future use or share with others. The app allows users to upload images of their clothing items, and it will then generate 3D models of these garments. To ensure accurate 3D model creation, the app provides a step-by-step guide on how to capture clothing images.

Additionally, the app uses artificial intelligence (AI) to analyze user preferences, past outfit choices, and available wardrobe items to offer personalized outfit suggestions. This feature streamlines the outfit selection process, enabling users to quickly choose an appropriate outfit without manually browsing through their entire wardrobe. Users can also mark their favorite outfits and save them in a personal archive, making it easier to access pre-selected outfits for specific occasions. This functionality helps busy individuals save time and make more efficient wardrobe choices.

1.5.2 Target Audience

The target audience for this application includes individuals who face various challenges related to outfit selection. This includes people with mobility limitations or physical conditions that make it difficult to change clothes or try on outfits. It also caters to individuals who are temporarily unable to physically try on clothes due to health reasons, such as recovering from an injury or illness. For such users, the app offers a solution by allowing them to virtually try on clothes, eliminating the need for physical exertion.

Additionally, the app is designed for busy professionals who have limited time to spend on selecting outfits, providing an efficient way to plan and choose their wardrobe. Furthermore, the app appeals to fashion enthusiasts who enjoy experimenting with different outfit combinations and visualizing their wardrobe in a dynamic way, allowing them to express creativity and make quick decisions. The application provides a valuable tool for a wide range of users by helping them save time, be more efficient, and make better wardrobe choices.

1.5.3 Technical Requirements

The application will be developed for Android mobile devices, utilizing Unity to enable augmented reality (AR) functionality. By leveraging AR technology, the app will provide users with realistic virtual try-on experiences, allowing them to see how clothes would look and fit on themselves. AI-driven algorithms will be integrated into the app to offer personalized outfit suggestions. These suggestions will be based on individual user preferences, past outfit choices, and the available clothing items in their wardrobe, helping users make quick and informed decisions about what to wear.

To create these virtual try-on experiences, the app will incorporate 3D rendering tools to convert photos of clothing into accurate 3D models. These models will then be mapped onto the users' virtual avatars, ensuring that they get a true-to-life representation of how the clothes would look on them. Additionally, the app will integrate with cloud storage to securely store user data, including wardrobe items, saved outfit combinations, and preferences. This cloud integration will ensure that users can access their wardrobe and outfit data from any device, providing a seamless and convenient experience.

1.5.4 Exclusions

The application will not include features such as physical clothing measurement tools or in-person consultations. Additionally, the initial version of the app will not integrate with physical store inventories, meaning it won't provide real-time data about clothing availability or stock from external retailers. Furthermore, the project scope does not include the creation of any physical devices, such as touchscreen mirrors or other hardware solutions, which may have been considered in the early stages. The focus of the project is strictly on developing a mobile application that uses augmented reality to help users try on clothes virtually, without the need for additional physical infrastructure.

1.6 Standards, Ethics, Constraints, and Conditions

The development of this project will adhere to a set of standards, ethical guidelines, and constraints to ensure the application's functionality, user privacy, and overall quality. First and foremost, the application will follow industry best practices for software development, particularly in terms of AR and AI integration, ensuring that the app is reliable, user-friendly, and secure. The app will prioritize data privacy and user consent, strictly complying with data protection regulations such as the General Data Protection Regulation (GDPR) for users in the EU and other applicable privacy laws in different regions. All user data, including wardrobe items, preferences, and outfit choices, will be stored securely in the cloud with appropriate encryption measures to prevent unauthorized access.

Ethically, the application will respect user autonomy, providing them with full control over their data, including the ability to delete their profiles and information at any time. The app will avoid any intrusive data collection and will only use personal data in ways that are transparent and beneficial to the user, such as for personalized outfit suggestions. Additionally, the AI-driven outfit suggestions will be designed to be inclusive and free from biases related to gender, body type, or other personal characteristics.

From a technical perspective, the project will be constrained by the capabilities of mobile devices and the requirements of AR and 3D rendering. The application will be optimized to work efficiently on both iOS and Android platforms, with a focus on minimizing battery usage and ensuring smooth performance even on devices with limited resources. It will also need to be adaptable to different screen sizes and user interfaces to provide an accessible and enjoyable experience across various mobile devices.

Finally, the application's success will be measured against user engagement, user satisfaction, and the effectiveness of its personalized features. The project will be developed with clear milestones and deadlines, with thorough testing and quality assurance processes to ensure a high standard of usability and functionality at each stage of development.

CHAPTER TWO

2. LITERATURE REVIEW

In recent years, research has predominantly focused on the 3D modeling and design of objects and humans in digital environments. Our project aims to bridge the gap between high-cost, hardware-dependent systems and user-friendly, cost-effective solutions. By leveraging the widespread accessibility and versatility of smart devices, we not only address the practical challenges of virtual try-on systems but also propose an innovative application that enhances user experience and ensures aesthetic appeal through artificial intelligence integration.

2.1 Related Works

Chen et al. (2019) “The Design of Real-Time Digital Clothing Projection System” in article authors make a clothing system similar to our goal This method aims to develop 3D scenes and virtual garments within Unity while also tracking the user's body and skeleton with the help of the Kinect. Over distances that range from 1.5 to 4.5 meters, the Kinect tracks 25 skeletal joints, capturing depth and color images at 30Hz rates. Each joint is computed by processing information from two distinct skeletal points to give a precise fitting to the virtual clothing as the user poses. Since the Kinect's 3D coordinate is measured in meters, the X and Y axes can take up alternative positive or negative values depending on the place of the sensor. The sensor is fixed to the floor so that the whole user's body is imaged, thus avoiding problems related to negative coordinates. Virtual clothing has been divided into five parts: hips, shoulders, torso, arms, and legs. These sections are all adjusted for sizing according to the user's body measurements, further ensuring that the virtual clothing fits properly during projection.

VFR replaces traditional fitting rooms and is being offered in more and more shops, malls, shopping centres, and online stores that support online garment sales. The VFR has been one of the trending research interests since the development of the tracking system in Microsoft Kinect. In this paper, an interactive 3D virtual fitting room using Microsoft's Kinect tracking, the rigging technique from the 3D modelling software Blender, and the implementation of the VFR was proposed. The VFR manages virtual fitting progress by forming three-dimensional simulations and visualizations of garments on virtual counterparts of real prospective buyers (users). Users can view clothing animations in various poses that follow their body movements. The system evaluates the user's match and guides them in choosing the suitable clothing size using Euclidean distance (Kurniawati, Kusumaningsih, Aliffio. 2020).

Cohen, Li (2021) used similar techniques. The process begins with the construction of a human body model and discusses three methods of 3D modeling. The first one applies to 3D scanners producing high-resolution models. This method is very common, easy, and exact, but quite costly and not easy to use. Thirdly, images are used to create a 3D model by representing it with a series of pictures captured from a variety of angles. Unfortunately, it is not easy to make accurate models from just images because of body structure complexity. The third, and also the one used in this study, method begins with a generalized target model and morphologically adapts it into an individualized model, remapping its prominent points to coincide with the images of the individual subject to be reconstructed. The method relies on the statistical average model created using the MPII human shape models built on the CAESAR dataset. The particular body shape of an individual is reconstructed using images in two views: frontal and side (left side). These images can be obtained with the help of a static camera or a smartphone, and the planes of the camera images are perpendicular to the ground. For example, the smartphone is fixed on a stand or leaning against a windowsill to allow for the front and side images to be captured. To account for the distance variation during image capture, the outline of the user is normalized in height after background segmentation into a uniform height across images. The user's height in pixels also correlates with their real-life height in the final 3D model to ensure correct scaling.

Liu, Li, Lu (2024) present a robust and highly realistic approach to generating 3D clothing models with visually consistent style and wrinkle distributions from a single RGB image. More impressively, the entire process only takes a few seconds. The high-quality results they realize come from a combination of learning and optimization, making the approach very robust. First, neural networks predict a normal map, a clothing mask, and a learning-based clothing model from the input image. The predicted normal map effectively captures high-frequency clothing deformations from image observations. Subsequently, a normal-guided clothing fitting optimization will adopt the normal maps to guide the clothing model to generate realistic wrinkle details. Finally, a garment collar adjustment strategy stylizes the clothing results using the predicted clothing masks. It leads to a natural extension of a multi-view version of the method for fitting clothes, while further improving realism without the tedious efforts. Extensive experiments validate that the proposed approach achieves state-of-the-art geometric accuracy and visual realism for clothes. More importantly, it is highly adaptable and robust when applied to in-the-wild images. Besides, it can be naturally extended to multi-view input for enhancing realism. In summary, the described approach offers a low-cost and easy-to-use alternative to the realistic modelling of garments.

In "Analysis Methods of the Design of Clothing Customization System in AR" article author's goal was to create a clothing system with AR also its advantages in AR systems for this mission. They originally thought of this system only for dresses that could be easily designed to be custom-made. VR denotes virtual realism, that is, using computer technology to digitally imitate a real environment. People are thus still able to experience a very real-life ambience involving the senses of sight, hearing and touch among others. VR involves simulating real scenes in a digital environment. Today, AR offers a richer and more realistic experience. It integrates some digital elements, such as virtual images, animations, and sounds, with real scenes on the fly. AR is capable of blending real and digital environments by combining dynamic and statically real scenes with virtual scenes. It has better scalability compared to virtual reality and provides significantly greater integration (Liying, Liyao, Weng.2022).

In the “VICO-DR: A Collaborative Virtual Dressing Room for Image Consulting” article the researchers from Italy made a similar system for gaming we want to make for daily usage. They approach the situation as a game design and use Unreal Engine 4 because it can support multiple platforms such as PC, mobile, AR and VR devices, also being written in C++, Unreal Engine 4 can be extended via native or third-party libraries. This work attempts to see the development of realistic and detailed 3D avatars for VICO-DR as well as other VDR-type applications with the MB-Lab Blender plug-in. MB-Lab allows the user to make 3D rigged avatars with inputs based on gender, ethnicity, and body measurements and fine-tunes them with the help of the Blender "Shape Key" method. The avatars are exported for Unreal Engine 4 (UE4) as FBX files, and a Python script solved the coordinate system incompatibility of Blender and UE4. The addition is a low-detail "collision mesh" that would ensure fast rendering, giving the illusion that the avatar is interacting with the clothes in reality. In addition to this, there is a replacement of the MB-Lab skeleton with the standard UE4 mannequin skeleton, and the bones are tweaked for various compatibilities. The virtual environment is kept to a minimum so that the attention is on the avatar and clothes (Manfredi et al. 2023).

In this project, the authors aim to model physical objects in 3D with 360-degree scans using Microsoft Kinect and point cloud technology and visualize them in virtual environments. The study focused on processing point clouds, filtering depth data, and combining them with triangulation methods. The resulting models were visualized virtually with tools such as MeshLab and OpenGL, and various scanning and filtering methods were examined in this process.

Virtual try-on applications let buyers see themselves wearing different garments without physically trying them on. The dominant approach to virtual try-on right now makes use of virtual fitting rooms where multiple cameras identify the user's skeleton and posture to render garments on their image. While this is realized through several different techniques, some users may feel uncomfortable standing in front of cameras inside a fitting room. Therefore, Sekhavat (2016) presented a different approach, which would enable customers to see a 3D model of themselves wearing garments on their personal mobile device using AR. A model selection technique is introduced to accurately match the user's anthropometric features among the automatically generated 3D human models. This approach includes body customization and face generation modules for creating a realistic representation. Several quantitative experiments and user studies were conducted to evaluate the accuracy, efficiency, usefulness, and privacy of the proposed technique.

While there exist models of 3D clothing learned from real data, no method currently predicts clothing deformation as a function of garment size. Tiwari et al. (2020) introduce SizerNet, which predicts 3D clothing conditioned on human body shape and garment size parameters, and ParserNet, which infers garment meshes and shapes under clothing with personal details in a single pass from an input mesh. SizerNet predicts the dressing effect of garments across different sizes, while ParserNet can directly edit the clothing on an input mesh without requiring any scan segmentation, which is a hard problem in its own right. They propose the SIZER dataset for training these models, which includes scans of 100 subjects dressed in casual clothes of various sizes, resulting in roughly 2,000 scans. The dataset contains the scans, registrations to the SMPL model, and segmented scans annotated with clothing parts, garment categories, and size labels. Their experiments show improved parsing accuracy and size prediction compared to baseline methods trained on SIZER.

In this study, 4 depth cameras (Kinect v1) were placed around the object in order to obtain the 3-dimensional structure of an object. Computer bandwidth and interference problems encountered during the operation of multiple Kinects were solved with additional USB 2.0 controllers and electronic shutters. Calibration was performed on infrared images to merge the acquired images. Alignment problems were solved by the Iterative Closest Point method and an alignment with 3.41% error rate was obtained. Poisson Surface Reconstruction was implemented in Meshlab software for 3D modeling objects with different geometrical characteristics (Tunçer, Gümüştekin. 2020).

During the last few years, several virtual prototyping technologies have been developed to innovate the apparel industry. From body acquisition to garment modeling and simulation, there is a dedicated tool for each step of the garment design process with the aim of making the process more efficient and streamlined. However, most of the existing solutions are based on expensive hardware and software systems. Vitalia, Rizzi (2018) focus on the first step of made-to-measure garment design: customer measurement acquisition. They introduce a plug-in named Tailor Tracking, which enables the collection of measurements by interacting with the customer's avatar using hand gestures, mimicking traditional tailoring methods. Tailor Tracking is implemented by low-cost devices including the Microsoft Kinect sensor, Leap Motion device, and Oculus Rift, together with open-source libraries such as VTK and Qt. In the approach presented here, several Kinect v2 sensors capture the customer's body and motion simultaneously, while emulating postures that provide more accurate measurements. The user is also given a virtual measuring tape to simulate the tailor's traditional tool. A case study was made for a men's shirt, and the first tests involved one tailor and 14 people with different levels of experience in garment design and virtual reality technology. Tailor Tracking is a practical alternative to the prior art of automatically extracting a customer's anthropometric measures from avatars.

Yoon et al. (2021) present a method for clothes retargeting, generating the possible poses and deformations of a given 3D clothing template model with the aim of fitting a person in a single RGB image. The authors have marked this problem inherently ill-posed since ground truth data images of people wearing the same 3D clothing template in the same pose-cannot be acquired. Limbert et al. mitigates this further by leveraging large-scale synthetic data through physical simulation, effectively mapping 2D dense body poses onto 3D clothing deformations. They propose a semi-supervised learning approach on the synthetic dataset for ensuring physical plausibility in the estimation of 3D deformations to align with prescribed body-to-cloth contact points, hence the clothing silhouette for ease of application on unlabeled real-world images. Their approach is implemented in a newly designed neural network, CRNet, which integrates the semi-supervised retargeting process in an end-to-end manner. It verifies that their method produces highly realistic 3D pose and deformation field predictions, yielding an accurate retargeting of the clothing model to real-world examples.

Zhedong et al. (2022) proposed a self-supervised method for 3D clothing reconstruction that recovers the geometric shape and texture of human clothing from a single image. They identified three primary challenges faced by existing methods: (1) 3D ground-truth meshes of clothing are usually inaccessible due to annotation difficulties and time costs; (2) Traditional template-based approaches struggle to model non-rigid objects such as handbags and dresses, which are prevalent in fashion imagery; (3) The inherent ambiguity compromises the model training, such as the dilemma between a large shape with a remote camera or a small shape with a close camera. In the work, the authors introduce a causality-aware self-supervised learning approach to adaptively reconstruct 3D non-rigid objects from 2D images without 3D annotations. Specifically, they addressed the ambiguity of four implicit variables: camera position, shape, texture, and illumination, by employing an explainable structural causal map to guide the model design. The proposed model structure follows the spirit of the causal map, which explicitly considers the prior template in the camera estimation and shape prediction. Their approach integrates causal intervention tools, including two expectation-maximization (EM) loops, to (1) disentangle the four encoders and (2) enhance the utilization of prior templates. Extensive experiments conducted on two 2D fashion benchmarks, ATR and Market-HQ, demonstrated that their method achieves high-fidelity 3D reconstructions.

2.2 Comparison with the Existing Solutions

In the theses reviewed above, the general focus has been on processing and designing objects and humans in 3D within computer environments. Among these, the project most similar to ours is the thesis titled "3D Magic Mirror: Clothing Reconstruction from a Single Image via a Causal Perspective" which involves designing a mirror that creates 3D models of individuals and shows how clothing would appear on them. However, our design aims to develop an application that can be easily installed and used on widely accessible devices such as smartphones and tablets, which are commonly used today.

This approach offers significant advantages over the mentioned studies. The cost is substantially lower because our solution does not require any additional hardware; it is purely application-based. Moreover, using widely adopted smart devices provides ease of use and portability. In contrast, devices such as mirrors or similar hardware utilized in other studies occupy considerable space and have disadvantages like fragility. Modern smart devices, on the other hand, are designed to be more durable against impacts such as drops and breaks, and are produced as much smaller, portable devices compared to large materials like mirrors.

Our project seeks to miniaturize high-cost and relatively cumbersome systems into the size of a smartphone while offering a design that is easy to use and accessible to everyone, regardless of age. With its affordability and universal usability, the application aims to provide ease of life for individuals with disabilities, address the common issue of clothing selection for many people, and, with the help of artificial intelligence, ensure users always look stylish.

CHAPTER THREE

3. REQUIREMENTS/REQUIREMENT ENGINEERING

In this chapter, we will outline the functional and non-functional requirements that form the foundation of our project. These requirements will define how the system is expected to operate, and the quality attributes it must fulfill to meet user and business needs.

3.1 Functional Requirements

Functional requirements specify the essential features and operations that the system must provide to meet the end user's expectations. These functionalities must be implemented as part of the system to fulfill contractual obligations. The program should perform the required operations and produce the expected outputs. All requirements stated by the user must be incorporated into our final application.

3.5.1 Collecting Data

In our program, data collection is the foundation of our project. All of the features we promised are based on the data we collected from users. The project collects data from various sources: user input, internal databases, and external APIs. The project will collect users' height, weight, age, and hair color in a user-friendly way. This information is required to be reentered every three months but users can change any of them on their profile. The data for the clothing, shoes, or accessories will be collected from the photos the users upload. Additionally, local weather data will be collected using the users' location, or the location the user gives. This information is used alongside the users' favorite outfit combinations, favorite color combinations, and their outfit combinations to be recorded and used for outfit suggestions. The collected data will be stored in our database for further analysis and processing.

3.5.2 Analyzing Data

The collected data from the users will be analyzed to give outfit combinations to users. The age, height, and weight of the user are used to compare the outfit combinations to other users of the same features and find the similarities they prefer in their outfits. The hair color of the user is used to find color combinations for the outfits by analyzing the user's past combination with the same hair color, or from other users with the same hair color. The weather data of the given location or the users' location is used to create outfit combinations by analyzing the past weather and outfit data or from the clothes categories.

3.5.3 Evaluation Metrics

Evaluation metrics are quantitative measures used to assess the performance and effectiveness of a machine-learning model. These metrics provide insights into how well the model is performing and help in comparing different models or algorithms. In a machine learning model, it is crucial to assess its predictive ability, generalization capability, and overall quality (Srivastava, 2025). Since our project is user-centric, we have chosen Normalized Discounted Cumulative Gain (NDCG) as the primary evaluation metric. NDCG enables us to assign scores to outfit combinations, prioritizing more relevant or higher-scoring outfits to be shown to users first, while less relevant or lower-scoring combinations are displayed later. User feedback on these combinations will influence and improve future outfit recommendations.

3.5.4 Integration

In our application, the system will integrate with various platforms and tools to enhance user experience and functionality. The application will connect with the device's camera and gallery to allow users to upload images of their clothing items directly or to try uploaded clothing on themselves. The uploaded images will be processed using an integrated image recognition API to identify clothing categories (e.g., shirts, pants, shoes) and extract details such as color. Integration with weather services will also provide tailored outfit suggestions based on the local weather conditions.

3.6 Non-Functional Requirements

Nonfunctional requirements are defined as quality attributes, system constraints, and operational characteristics. Our program should ensure user satisfaction and good technical performance. The difference between functional and non-functional requirements is functional requirements can be calculated by functions; this can help us to understand what can happen while the user is using our program or what happened on the system while it was working (past control) if some error popup on our program. Besides that, on usage user must feel satisfied with our program for this our program must have a nice; performance, usability, scalability, security, availability, compatibility, maintainability, and accessibility we can see more details about all our requirements below

3.2.1 Application platform

Our application platform choice is Android 10 and above for Android also iOS 13 and above because of safety, performance, availability, etc. We believe that most of our users will use these two operating systems. Our hardware requirements will be a little bit high for using our program efficiently and we will need 3 GB of RAM and a processor capable of running AR applications and a camera of at least 8 MP is recommended for accurate 3D clothing modeling. For the development framework the application will be built on unity AR functionality, and AR features will be implemented using ARKit for iOS and ARCore for Android. Also, users need a stable internet connection for downloading and uploading the clothes 3D models we will test our application on a range of most popular devices including mid-range and flagship smartphones to ensure broad completely.

3.2.2 User-friendly interface

Our application interface should be intuitive and easy to navigate for users of all age group, including individuals with limited technic knowledge also a comprehensive user manual and tooltips can be provided within the application

3.2.3 Processing Time

Our application should render augmented reality (AR) visuals with a latency of less than 100 milliseconds. The reason is a standard 60 Hz display refreshes every 1/60 seconds because of the formula below

$$F = \frac{1}{\mathrm{T}}$$

F: Frequency (Hz)

T: Period (s)

As we can see from the formula 1/60 = 16.67 ms so 60 Hz monitors are refreshing the page every 16.67 ms. In AR and VR development accept that 100 ms is often cited as the upper limit for acceptable latency to ensure a high-quality user experience

3.2.4 Availability

The application we made should support a growing number of users without significant degradation in performance and it should be capable of at least 1000 simultaneous users during peak users. Also, the application must be compatible with Android 10 and above, as well as iOS 13 and above (these systems are still in use). The application should function seamlessly across devices with different screen sizes and resolutions

3.2.5 Reliability

The application should maintain an uptime of at least 99.9% always ensuring reliability for users also cloud-based backups can be implemented to prevent data loss in case of server failure

3.2.6 Security

Our program contains so much personal information like pictures, clothes, clothing choices of members, favorite colors, etc. So, our system must be secure. This information should be encrypted during storage and transmission so we should use end-to-end encryption and other relevant data protection regulations. And our system should include solid authentication mechanisms for preventing unauthorized information access

CHAPTER FOUR

4. DESIGN

4.1 Database Design/ER Diagram

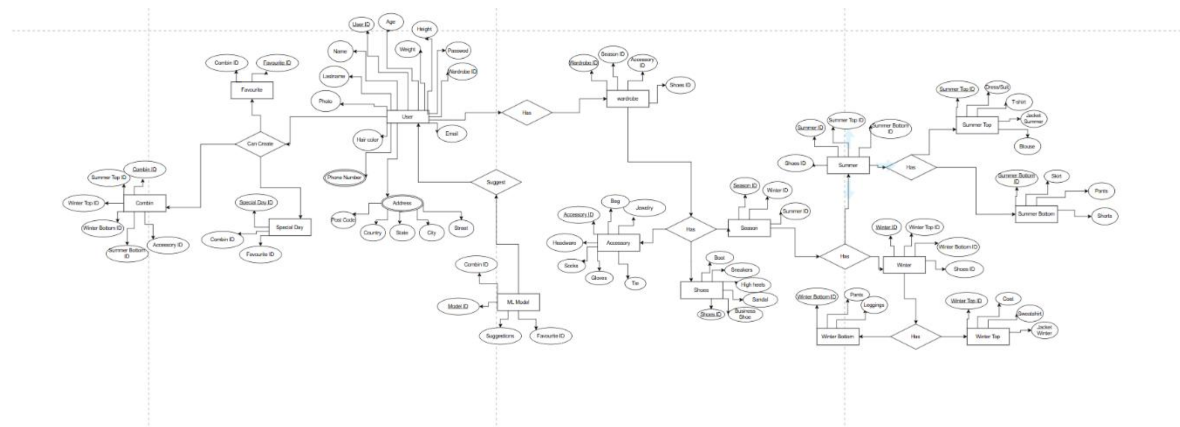

Figure 4.1: Complete ER diagram

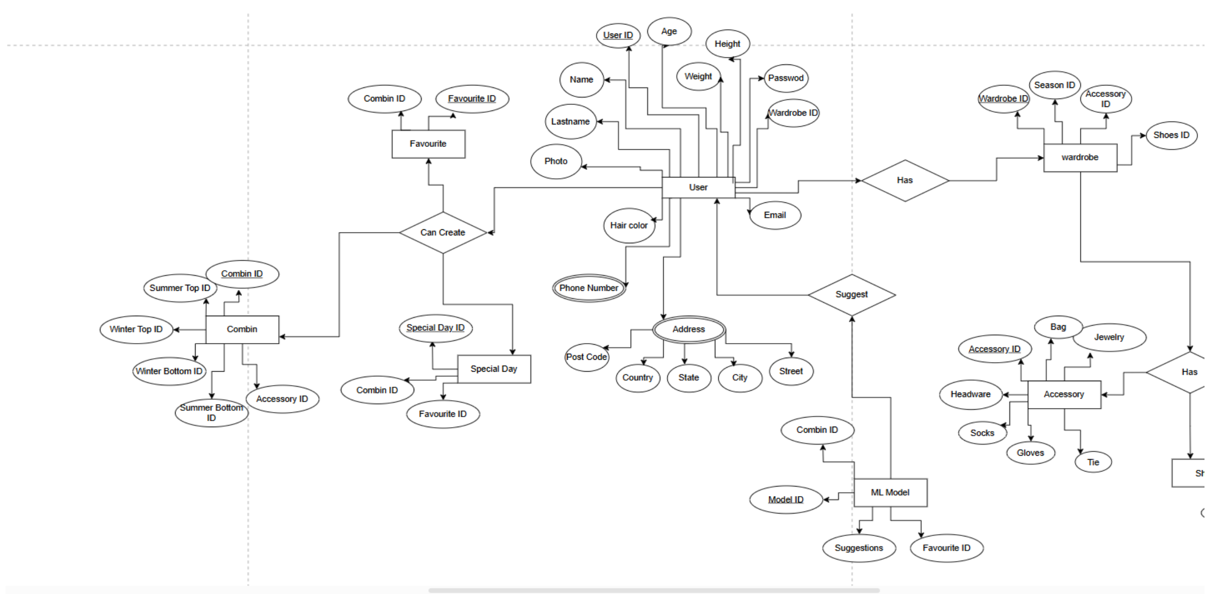

Figure 4.2: ER diagram part one

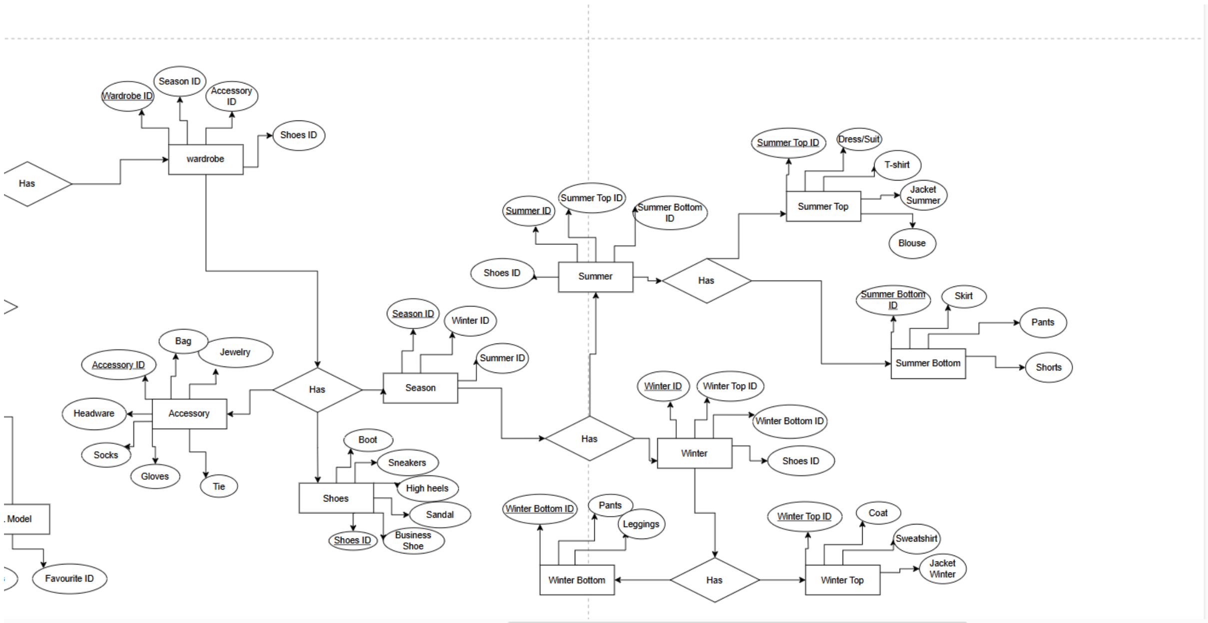

Figure 4.3: ER diagram part two

4.2 UI Design

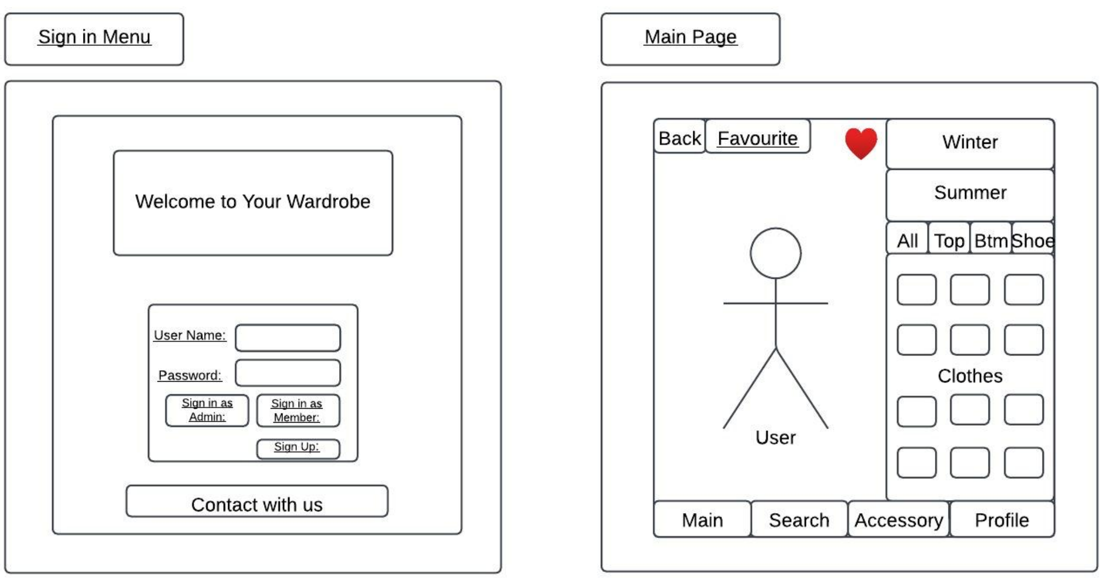

Figure 4.4: Sign-in menu and main page interface

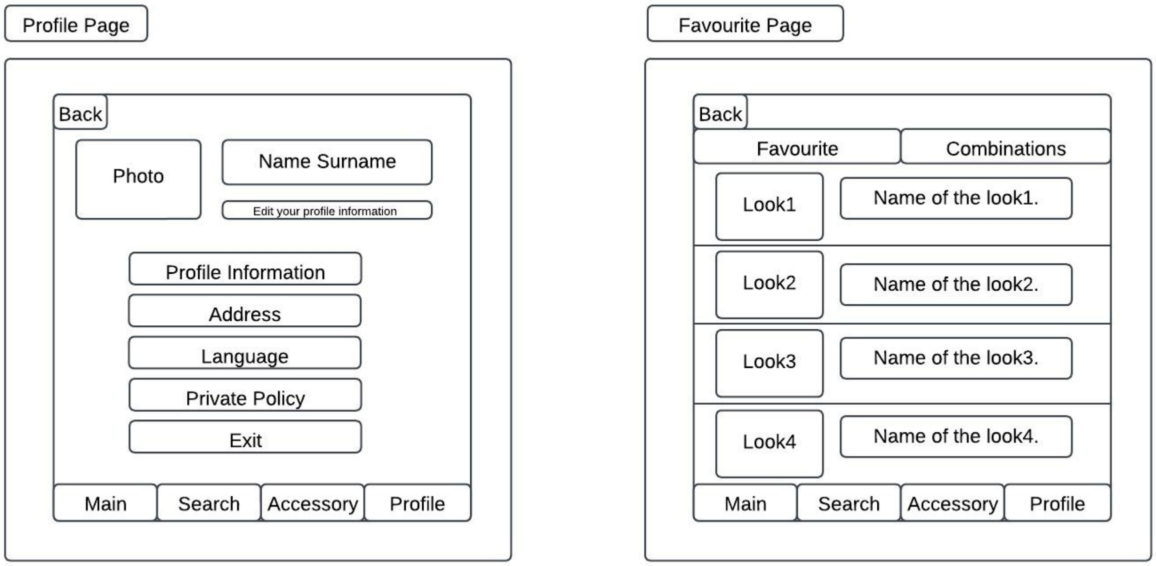

Figure 4.5: Profile page and favourite page interface

4.3 Sequence Diagram

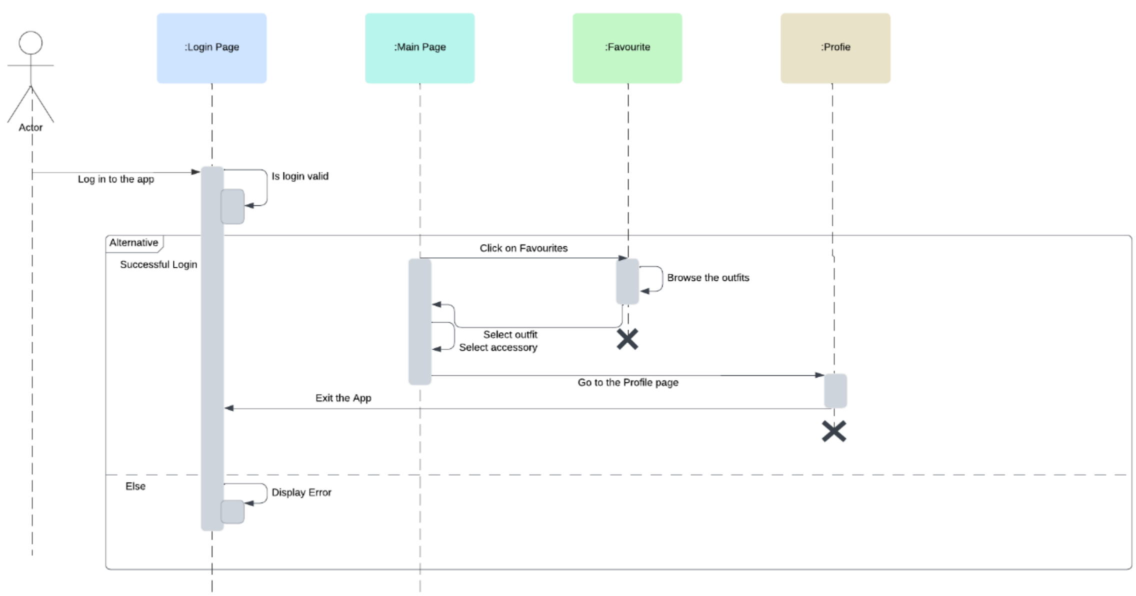

Figure 4.6: Sequence diagram of choosing an outfit from favourites

4.4 Activity Diagram

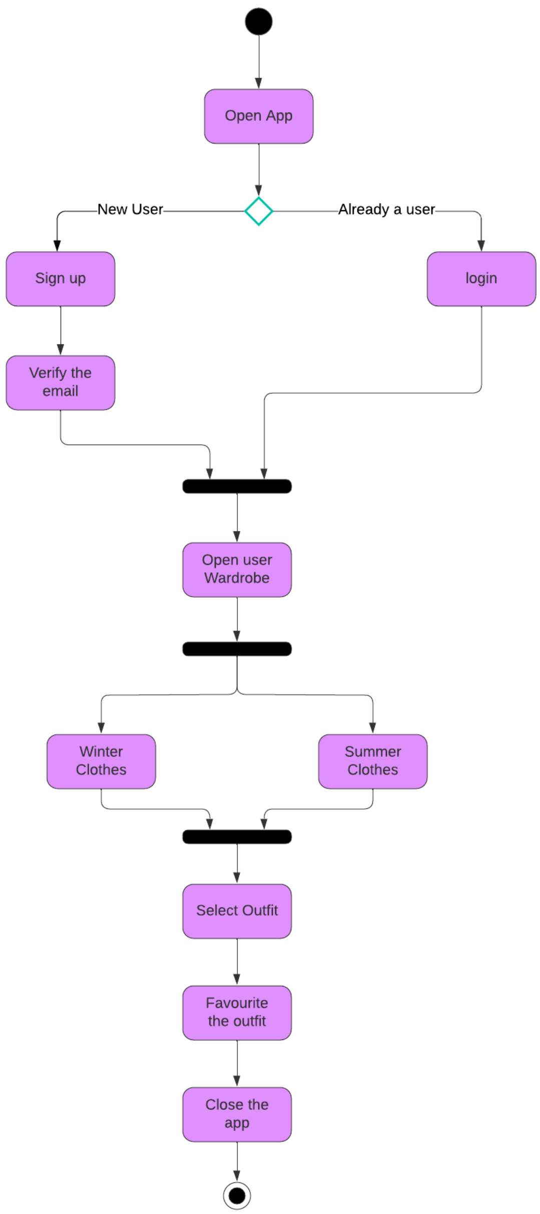

Figure 4.7: Activity diagram of a user logging in and choosing an outfit.

CHAPTER FİVE

5. IMPLEMENTATION

5.1 Dataset Description

In our project we decided to use MySQL Database because we both familier with it from our past projects. Our Dataset consists of 22(twenty two) tables and these are connected with each other with foreign keys. For attributes which needed a selection Enums we are used to allow users to choose selected categories. In our dataset each user have a wardrobe and the wardrobe divides as winter clothing and summer clothing. Seasonal clothings also divide between tops and bottoms. With tops and bottoms tables of each season, shoes and accessories divided to sub tables named tablename_images to store the images of the users clothing. The images were kept as url in our dataset. In the user table, user's information like heigth, weight, age, hair color, password, email and users's address for feature uses such us creating combines based on the users hair color or age. User's address right now is not used but in the future when the project is improved and be able to used for online shopping.

There are tables which were included to Database:

- **User** (Attributes: User_id (primary key), Password, Name, Lastname, Photo, Phone Number, Email, Hair Color, Age, Height, Weight, Wardrobe ID (foreign key))

- **Address** (Address_id (primary key), User_id (foreign key), Street, State, City, Country, Post Code)

- **Favourite** (Favourite_id (primary key), Combin_id (foreign key))

- **Combin** (Combin_id (primary key), Summer_Top_id (foreign key), Summer_Bottom_id (foreign key), Winter_Top_id (foreign key), Winter_Bottom_id (foreign key), Accessory_id)

- **Special_Day** (Special_Day_id (primary key), Combin_id (foreign key), Favourite_id (foreign key))

- **Accessory** (Accessory_id (primary key), Bag, Headware, Socks, Gloves, Tie, Jewelry);

- **Wardrobe** (Wardrobe_id (primary key), Season_id (foreign key), Accessory_id (foreign key), Shoes_id (foreign key))

- **Season** (Season_id (primary key), Winter_id (foreign key), Summer_id (foreign key))

- **Shoes** (Shoes_id (primary key), Boot, Sneakers, High_heels, Sandal, Business_Shoe)

- **Winter** (Winter_id (primary key), Winter_Top_id (foreign key), Winter_Bottom_id (foreign key), Shoes_id (foreign key))

- **Winter_Top** (Winter_Top_id (primary key), Coat, Sweatshirt, Jacket_Winter)

- **Winter_Bottom** (Winter_Bottom_id(primary key), Pants, Leggins, Skirt)

- **Summer** (Summer_id (primary key), Summer_Top_id (foreign key), Summer_Bottom_id (foreign key), Shoes_id (foreign key))

- **Summer_Top** (Summer_Top_id (primary key), Dress/Suit, T-shirt, Jacket_Summer, Blouse)

- **Summer_Bottom** (Summer_Bottom_id (primary key), Skirt, Pants, Shorts)

5.2 Methodology

To generate 3D images from 2D photos, we are utilizing an existing project developed by Zhedong Zheng, Jiayin Zhu, Wei Ji, Yi Yang, and Tat-Seng Chua, titled "3D Magic Mirror: Clothing Reconstruction from a Single Image via a Causal Perspective." This advanced project is capable of reconstructing realistic 3D clothing models from single 2D images. The prerequisites for running the project include Python 3.7, CUDA 11.1, and a Linux-based operating system, all of which are essential for ensuring compatibility and performance during the reconstruction process.

The original project supports training and testing on three different datasets. However, in our case, we plan to use only one of these datasets. At the current stage of development, the specific dataset has not yet been finalized. We expect to make that decision after completing the setup of our application and evaluating which dataset produces the most realistic and visually accurate clothing reconstructions on human models.

For developing the mobile application, we have chosen to use Python as the primary programming language. This decision is based on Python's simplicity, readability, and widespread use in machine learning and computer vision projects. Additionally, the 3D clothing reconstruction model we are integrating is also built in Python. By maintaining the same programming language across both the mobile app and the backend model, we aim to streamline the integration process. This consistency allows for easier communication between the app and the model—for example, capturing photos through the app and sending them directly to the model for 3D reconstruction. Ultimately, using Python throughout the entire pipeline improves development efficiency and simplifies data handling between components.

5.3 Tools/Libraries

For developing this project, we have chosen several tools and libraries to streamline the process of creating the mobile application and the 3D clothing reconstruction system.

- **Programming Language**: We are using Python as the primary programming language due to its simplicity, readability, and widespread use in machine learning and computer vision projects. Python's versatility and support for deep learning libraries make it an ideal choice for both the backend model and the application.

- **IDE and Development Platforms**: PyCharm is being used as the Integrated Development Environment (IDE). It provides robust support for Python and is well-suited for managing and debugging Python projects. Android Studio is being used for mobile application development, particularly for creating XML layouts and utilizing its visualization features for better UI design and testing. Android Studio allows us to create the front end without needing to save and open our app every time we make a change, thanks to its real-time design and preview features.

- **Computer Vision and Image Processing**: OpenCV is being used to access the camera and capture images from the user's device. This library is essential for image processing tasks and enables us to integrate the camera functionality into the mobile application. MediaPipe will be used to track the user's skeleton and capture key body landmarks in real time. This allows for more accurate pose estimation, which is crucial for the 3D clothing reconstruction process.

- **Machine Learning**: The machine learning model for 3D clothing reconstruction has not yet been fully designed, but we plan to use TensorFlow for model creation. TensorFlow is a powerful open-source library for machine learning and deep learning tasks. The model will be developed using Jupyter Notebook, which offers an interactive environment for data analysis and model experimentation before finalizing the deployment.

These tools and libraries will work together to enable the development of mobile applications, integrate computer vision techniques for user pose tracking, and support machine learning-based 3D clothing reconstruction.

5.4 Implementation Details

In our application architecture, the backend and frontend work together to deliver a seamless user experience. The backend is responsible for handling the core logic, data storage, security, and communication with the database. It is built using Spring Boot, where we define entities, services, repositories, and controllers to manage and process data efficiently. The backend exposes RESTful APIs that allow the frontend to interact with the server. The front end, on the other hand, serves as the user interface of the application. It is developed using Android Studio, allowing users to register, log in, update their profiles, and perform other operations through a mobile application. The frontend communicates with the backend through HTTP requests and displays the responses (such as user data or status messages) to the user.

Together, the backend and frontend create a fully functional system where the backend ensures reliable data processing and security, while the frontend focuses on delivering an intuitive and responsive user experience.

5.4.1 Frontend Implementation

A well-designed user interface (UI) is the cornerstone of any successful application, especially for solutions like the **3D Clothing Try-On and Combination Suggestion Application**, which caters to a diverse audience with varying levels of technical proficiency, preferences, and accessibility needs. In this project, the frontend serves as both the primary point of interaction and the central hub for ensuring accessibility and inclusivity. The goal was to create an intuitive, responsive, and user-friendly interface that empowers users to engage with the application effortlessly—regardless of their age, technological background, or physical limitations.

This approach reflects the philosophy that technology should adapt to users, not the other way around. With this mindset, the frontend of the Elayna project was developed to provide a seamless experience that supports all users, from casual shoppers to fashion enthusiasts. This section provides a comprehensive overview of the tools used, design principles followed, challenges encountered, and solutions implemented in building a modern, robust, and accessible user interface.

5.4.1.1 Tools and Technologies Used

Android Studio served as the foundation for the frontend development process. With a comprehensive feature set for creating, testing, and debugging mobile applications, Android Studio is the industry standard IDE for Android app development. Iterative UI development is made simple and effective by its layout editor, which allows XML-based visual design with real-time previews. Android Studio is based on JetBrains' IntelliJ IDEA, a robust and adaptable IDE. Because of this common framework, developers can easily transition between design and implementation, controlling application logic and user interface layouts in a standardized process.

5.4.1.2 User Interface Design and Page Descriptions

The target audience's expectations and challenges were taken into consideration when designing the user interface (UI). In order to minimize friction and empower users at every stage, each major screen was designed to support a particular aspect of the user experience.

The Sign Up Page's simple yet eye-catching design greets new users. In addition to a welcoming greeting ("Say Hi to Elayna – Your personal fashion designer"), it has unambiguous input fields for the username, email, and password. The emphasis on simplicity and clarity makes the onboarding process as easy as possible for users who are not familiar with mobile applications or technology. Real-time error messages help users avoid frustration during registration, and input validation mechanisms guarantee that users are directed toward creating robust, legitimate accounts. A Successful Registration Notification with the clear popup message "Success! User created successfully" confirms that the account was created after the registration form has been correctly submitted. Users are reassured and their experience with the application is streamlined by this instant feedback.

By changing their profile picture, choosing their hair color, and entering important personal data like height, weight, age, and phone number, users can customize their experience on the Profile Editing Page. The interface's easily identifiable icons and fields with clear labels were intended to make these actions simple. The application gives users the ability to easily update and manage their profiles, which gives them a sense of control. This is crucial for people who might have trouble keeping track of their wardrobe or whose needs change over time.

Both normal users and administrators can access the Login Page. With separate buttons for "Login," "Admin sign in," "Member sign in," and "Sign up," as well as fields for email and password, it offers a clear, user-friendly interface. For returning users, the "Welcome Back to Your Wardrobe" message creates a sense of familiarity and warmth, which lowers anxiety and encourages regular app usage.

Every UI page uses the same design language. The navigation flow is logical, requiring fewer taps to complete any task, the color schemes are purposefully soothing and inclusive, and the button shapes and font sizes are selected for readability and usability. Toast notifications and pop-up dialogs are examples of feedback mechanisms that keep users updated at every turn.

5.4.1.3 Challenges and solutions

There were numerous important challenges in creating a frontend that could accommodate a wide range of user needs. The first was making sure the app worked and looked good on a variety of Android devices, each with a different screen size, resolution, and hardware setup. The development team used responsive design techniques to solve this, making use of XML's ConstraintLayout and RelativeLayout and meticulously defining all dimensions, margins, and paddings in density-independent pixels (dp) and scale-independent pixels (sp). Both the Android Emulator and actual devices were used for thorough testing, which helped identify and fix problems prior to deployment.

Another major worry was the security of user data. Strong security procedures had to be put in place because the application would be handling personal data, such as emails and passwords. Passwords are never kept locally on the device; instead, they are sent over secure channels to the backend. To reduce the possibility of unwanted access, all sensitive data is hashed and encrypted before being stored on the server. Future enhancements might incorporate features like multi-factor authentication or biometric login for extra security, and secure authentication flows are enforced.

One of the main goals was to guarantee ease of use and accessibility for users of all backgrounds and skill levels. The design team avoided clutter and excessively complicated workflows in favor of a clear and simple interface. Even users with little technical expertise could use the app with confidence thanks to descriptive labels, visual cues, and carefully placed interactive elements. Throughout the user journey, user feedback mechanisms such as context-sensitive error messages and success notifications offered direction and assurance.

With Android Studio's "Live Preview" and "Instant Run" features, the team was able to see UI changes in real time and shorten the turnaround time for iterative improvements, increasing development and testing efficiency. For instance, the layout initially looked out of alignment on certain tablet devices during development. The team promptly fixed these discrepancies and recorded the fix for future use by modifying constraint settings and previewing in real time.

User experience and feedback presented additional difficulties. For example, handling loading states and error feedback carefully was necessary when managing asynchronous operations, like registration or login. In order to build user trust and patience even in the face of issues, the team implemented progress indicators and ensured that error messages were instructive but not technical.

5.4.1.4 Code Analysis

The project's emphasis on flexibility, clarity, and maintainability is reflected in the frontend code's structure. Android's Intent and Activity model handles screen navigation, resulting in smooth and logical transitions. Because each major function—such as registration, login, and profile updates, is handled in its own Activity class, modular development and simple debugging are supported.

Different XML layout files are used to define each user interface. Due to their extensive use of style resources and component reuse, these layouts enable quick screen prototyping and global changes to visual themes. The res directory contains static resources like colors, icons, and pictures, which are accessed through the Resource (R) system. This minimizes duplication and maintains consistency in the user interface.

Both the frontend and backend handle input validation, and users receive prompt feedback if they enter data incorrectly. To keep the user informed without being overbearing, Toast and AlertDialog notifications are widely used to convey success, errors, or necessary actions.

The code is designed with security in mind from the beginning. The device never stores sensitive information; all communications take place over secure channels. The architecture permits the future addition of client-side password hashing, SSL-only communication, and even biometric authentication, even though the current implementation relies on conventional password-based authentication.

Organizing the code is crucial. It is simple for new developers to contribute or for future team members to expand the functionality because methods and classes are well-documented with meaningful names and comments. While the modular design facilitates the incorporation of cutting-edge features like augmented reality, machine learning-based outfit recommendations, and wider accessibility support, the division of UI logic and business logic ensures simpler maintenance.

In the end, the frontend codebase is made to not only satisfy present specifications but also to offer a safe and scalable foundation for future development, guaranteeing that the application will continue to adapt to user demands and technical breakthroughs.

5.4.1.5 Frontend Conclusion

The Elayna project's frontend development process was driven by a dedication to security, usability, and accessibility. The team created a modern, dependable, and easy-to-use user interface by carefully analyzing the varied needs of the user base and utilizing strong tools and best practices.

Clarity and inclusivity were taken into consideration when implementing important features like simplified registration, unambiguous login processes, and adaptable profile management. Practical solutions to problems were found, frequently utilizing both technical know-how and input from early user testing. Future needs, such as more sophisticated user customization, new augmented reality features, or increased accessibility options, can be accommodated by the application thanks to its modular and well-documented codebase.

5.4.2 Backend Implementation

The backend implementation focuses on developing the server-side logic that powers the core functionalities of the application. We chose Spring Boot as our backend framework—a robust Java-based tool that simplifies the development and deployment of Java applications by reducing boilerplate configuration. This allows developers to concentrate more on implementing business logic. To promote clean architecture and maintainability, we applied the Factory design pattern in structuring our classes. The project is organized into seven main packages: Component, Configuration , Controller, Dto, Entity, Repository, and Service—each serving a specific purpose within the overall application architecture.

5.4.2.1 Component Folder

The Component folder contains the JWT authentication filter and the Schedule class. The JWT authentication filter verifies whether incoming requests include a valid JSON Web Token (JWT). It validates the JWT by checking its signature, audience, and issuer according to the HTTP filter configuration. Depending on the configuration, the filter can either immediately reject requests with invalid JWTs or pass the JWT payload to subsequent filters for further processing.

The Schedule class is responsible for periodically checking users' token expiration dates and deactivating users whose tokens have expired. This functionality is implemented using the "@Scheduled" annotation, which triggers the method execution at regular intervals defined by a cron expression. Cron expressions specify timing using fields for seconds, minutes, hours, day of the month, month, day of the week, and optionally, year. In our implementation, the expiration check runs every hour.

5.4.2.2 Configuration Folder

The Configuration folder contains three classes: WebConfig, ApplicationConfiguration, and SecurityConfiguration. The WebConfig class is responsible for configuring Cross-Origin Resource Sharing (CORS), enabling secure communication between the Spring Boot backend and our clients. It defines a WebMvcConfigurer bean that allows all endpoints (/**) to accept HTTP requests. This configuration permits the HTTP methods GET, POST, PUT, DELETE, and OPTIONS, and allows all request headers. This setup ensures that the Android application running on devices can interact seamlessly with the backend without being blocked by CORS restrictions.

The ApplicationConfiguration class sets up essential components for user authentication in the Spring Security context, including user details retrieval, password encoding, and authentication management. In contrast, the SecurityConfiguration class customizes the security settings by defining authorization rules, session management policies, integrating custom authentication mechanisms and JWT filters, and configuring logout behavior. Together, these configuration classes ensure the application is secure by managing both authentication and authorization according to the specified rules and policies.

5.4.2.3 Controller Folder

The Controller folder contains classes annotated with “@RestController”. The Spring “@RestController” annotation is a convenience annotation that combines “@Controller” and “@ResponseBody”. It is applied to a class to mark it as a request handler for RESTful web services using Spring MVC. This annotation maps incoming HTTP requests to handler methods and automatically converts the response body to JSON or XML. Using these controller classes, our Android application communicates with the backend. For example, the “UserController” class handles user-related endpoints such as verification, registration, login, and updating user information. It uses services and repositories to process requests and returns appropriate HTTP responses with relevant status codes and messages.

5.4.2.4 Data Transfer Objects Folder

In a Spring Boot application, Data Transfer Objects (DTOs) follow a design pattern used to encapsulate and transfer data between different layers of the application. DTOs help reduce the number of methods calls by bundling necessary data into a single object. The classes in this folder include only the fields relevant for data transfer. We use DTO classes in the controller layer to receive input from the client or to send the necessary data back in the response. For example, the “RegisterResponseDto” class contains the fields: status, body, and user ID. This class is returned by controller methods to convey the HTTP status, relevant client information, and the user ID to maintain the user context. Conversely, the “UpdateUserDto” class is used to capture updated user information from the client, containing fields such as user ID, profile image, height, weight, age, hair color, and phone number.

5.4.2.5 Entity Folder

The Entity folder contains classes that correspond to the database tables. Each class is annotated with "@Entity", which maps it to a specific table in the database. This setup allows us to interact with the database using the Java Persistence API (JPA), eliminating the need to write SQL queries manually. Within each entity class, we define attributes, along with their corresponding getters, setters, and constructors.

For attributes that represent foreign keys, we use appropriate annotations to describe the relationships between tables, specifying both the cascade type and fetch type. Cascading determines how operations (such as create, update, delete, or refresh) on one entity affect related entities. For example, to ensure that a user's associated token is deleted when the user is deleted, we used CascadeType.ALL. This cascade type ensures that all state transitions are propagated from the parent entity to the related child entities.

In Spring Boot with Spring Data JPA, fetch type defines when related entities are loaded from the database. There are two main strategies:

- FetchType.EAGER loads related entities immediately along with the main entity.
- FetchType.LAZY delays loading related entities until they are explicitly accessed.

In our project, we consistently used FetchType.LAZY across all entity classes to improve performance by avoiding unnecessary data loading.

5.4.2.6 Repository Folder

The classes within the Repository folder are annotated with "@Repository". A repository serves as a mechanism to encapsulate storage, retrieval, and search functionality for domain objects. In essence, repository classes act as the bridge between the application and the database, allowing us to perform data operations seamlessly.

These classes extend the JpaRepository interface, which is a specialized repository provided by Spring Data JPA. By extending JpaRepository, we automatically inherit a rich set of pre-defined methods such as save(), findById(), findAll(), deleteById(), and more. This greatly simplifies database interaction by eliminating the need for boilerplate code.

However, the built-in methods of JpaRepository are not always sufficient for all use cases. When more complex queries are needed, we use the @Query annotation to define custom queries. This annotation allows us to specify:

- The value, which contains the actual query to execute.
- The nativeQuery flag, which indicates whether the query is written in native SQL or in JPQL (Java Persistence Query Language).

By combining JpaRepository with custom queries, we achieve both simplicity and flexibility in database operations.

5.4.2.7 Service Folder

Service is a fundamental concept in Spring Boot applications, representing the layer responsible for handling business logic and encapsulating the core functionality of the application. Service classes are typically stateless and are designed to perform specific tasks that support the application's use cases.

In our project, service classes are organized within a package named service, which contains a subpackage called impl. The service package holds the interfaces of the service classes, while the impl package contains their corresponding implementations.

By separating service logic into interfaces and implementations, we achieve flexibility and maintainability. If a change in implementation is required in the future, we can simply create a new class that implements the interface, without modifying or deleting the original implementation. This makes our codebase easier to adapt and extend.

Within the service classes, once modifications are made to entities (such as updating user information or deleting a clothing item), we persist those changes to the database using the corresponding repository classes. By calling the save() method provided by JpaRepository, we efficiently persist changes without the need for writing custom queries or methods for each operation.

CHAPTER SIX

6. TEST/EXPERİMENTS

6.1 General Explanations

In this study, the project has been evaluated to determine whether its main objectives were successfully achieved. This section goes beyond simply presenting the work done; it also provides an analysis of the results and their significance. Since our project is a mobile application and does not involve any hardware components, we did not include graphs, hardware photos, or similar materials. Instead, we presented the application's initial testing phase, showcasing the appearance of the first pages during early development, and compared them with the current version of the app to demonstrate the progress and improvements made throughout the development process.

6.2 Test Results

Simulations were conducted to evaluate how the system would perform under ideal and controlled conditions. These tests provided valuable insights into the theoretical behavior and performance of the system. According to the results, the system generally behaved as expected in the simulations and achieved the desired objectives. However, some discrepancies were observed between the simulation outcomes and real-world measurements. These differences may be attributed to factors such as tolerance errors, software limitations, or environmental conditions that were not accounted for in the simulations. Nevertheless, the simulation results proved useful for planning and predicting system behavior. It can be concluded that while the simulated model offered valuable predictions, it is not sufficient on its own and must be complemented by real-world experiments for a comprehensive evaluation.

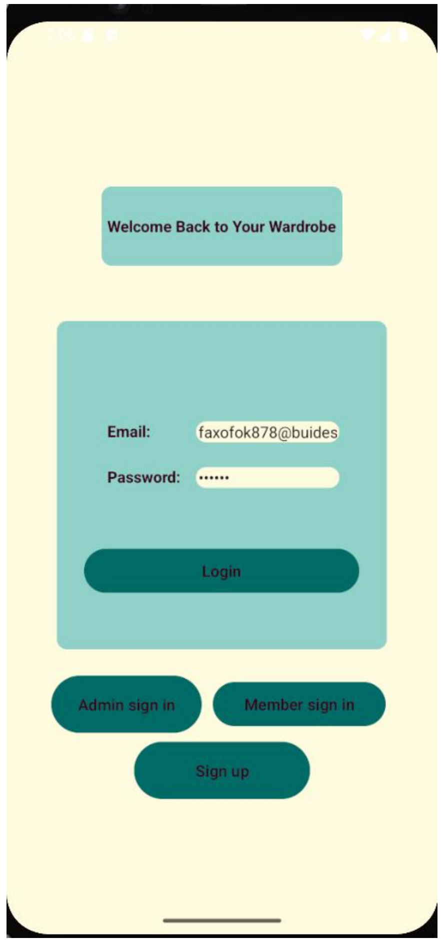

Figure 6.1: Login Page

Core functions, including login, registration, and profile page access, operated smoothly without any bugs encountered during testing.

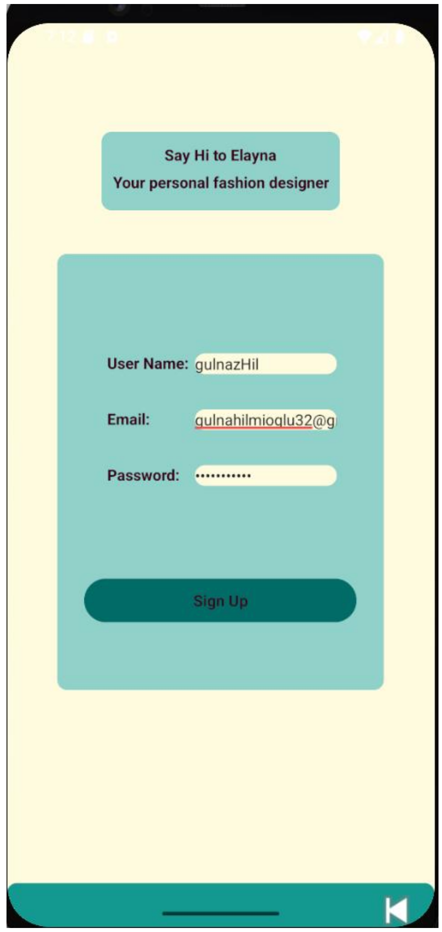

Figure 6.2: Register Page

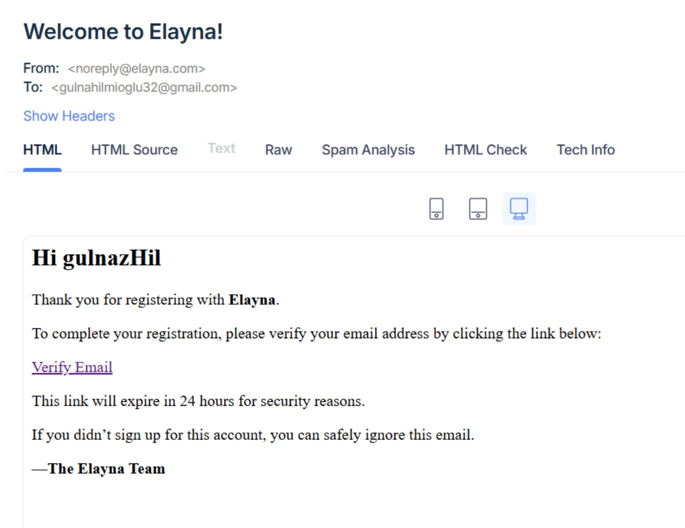

Figure 6.3: Verification Mail

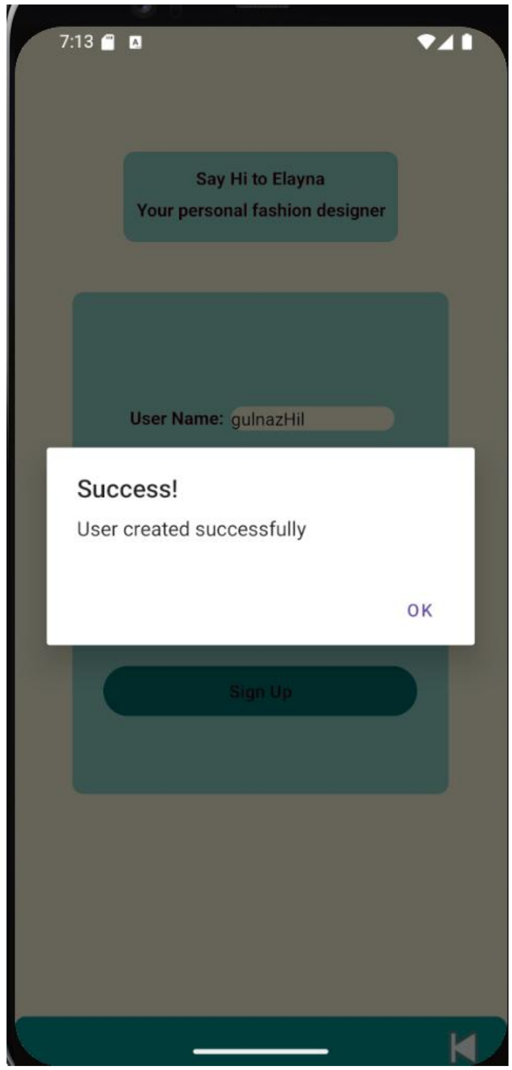

Figure 6.4: Toast message to confirm user creatin.

As can be seen from the images above, the mail-sending process worked without any errors. Users were able to register successfully and received verification emails. By clicking the link provided in the email, they were able to confirm their registration without any issues.

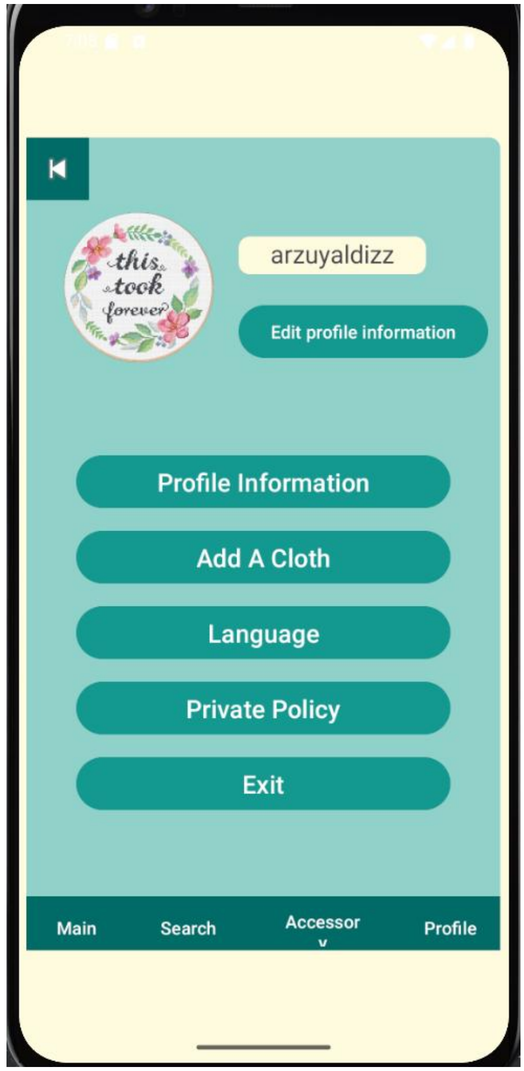

Figure 6.5: Profile Page

We did not encounter any issues when accessing users' profile photos from Firebase Storage and displaying them on the profile page.

Figure 6.6: Showing the clothing model on human

Accessing the photos and model files from Firebase did not cause any issues and the process went smoothly. However, we are still unable to display the 3D model on a human figure perfectly, and we aim to improve this functionality. In our initial tests, we used an older model that lacked visual quality in terms of both shape and color. To address this, we utilized Meshy.ai to generate more accurate and visually appealing 3D models.

Figure 6.7: Showing the clothing model on a human

CHAPTER SEVEN

7. CHAPTER SEVEN

7.1 CONCLUSION

In conclusion, this project has successfully fulfilled its primary objectives. We have developed a functional mobile application that allows users to log in securely, upload their clothing images, and view 3D visualizations of these clothes mapped onto a virtual human model. By achieving this, we proved that transforming 2D clothing into realistic 3D representations and displaying them on a user's body using augmented reality is not only possible but also highly useful for practical purposes.

Throughout the development process, we faced several challenges related to data processing, frontend-backend integration, and AR model rendering. Despite these obstacles, we managed to implement the key components of the system. The login system, user profile management, and 3D clothing visualization were completed and tested successfully. Although there are still some areas that require refinement, such as improving model alignment accuracy and enhancing the speed of 3D rendering, these can be addressed in future work.

Due to limited time and team size, we focused on completing the core features first. In the next phases, we aim to improve the user interface, add more customization features, and enhance the recommendation system based on user preferences and weather conditions. These enhancements will help us move from a working prototype to a fully developed, market-ready solution.

Overall, the project has been an important learning experience for us. We improved our technical skills in software development, AR technology, and database management, while also learning how to collaborate as a team and manage a real-world software project from start to finish. We believe that, with further development, this application has the potential to contribute meaningfully to the fashion tech industry and provide inclusive solutions for users with different needs.

REFERENCES

Chen, H. N., Chiu, C. F., Shih, T. K., Lin, C. Y., Utaminingrum, F., & Hui, L. (2019, August). The design of real-time digital clothing projection system. In 2019 Twelfth International Conference on Ubi-Media Computing (Ubi-Media) (pp. 102-107). IEEE, https://ieeexplore.ieee.org/abstract/document/9049561

Kurniawati, A., Kusumaningsih, A., & Aliffio, Y. (2020). Clothing size recommender on real-time fitting simulation using skeleton tracking and rigging. Jurnal Teknologi dan Sistem Komputer, 8(2), 127-132, https://jtsiskom.undip.ac.id/article/view/13559

Li, C., & Cohen, F. (2021). In-home application (App) for 3D virtual garment fitting dressing room. Multimedia Tools and Applications, 80(4), 5203-5224, https://link.springer.com/article/10.1007/s11042-020-09989-x

Liu, X., Li, J., & Lu, G. (2023). Modeling realistic clothing from a single image under normal guide. IEEE Transactions on Visualization and Computer Graphics, https://ieeexplore.ieee.org/abstract/document/10045805

Liying, F., Liyao, M., & Weng, N. G. (2022, March). Analysis Methods of the Design of Clothing Customization System in AR. In Proceedings of the 8th International Conference on Computational Science and Technology: ICCST 2021, Labuan, Malaysia, 28–29 August (pp. 411-423). Singapore: Springer Singapor, https://link.springer.com/chapter/10.1007/978-981-16-8515-6_32

Manfredi, G., Gilio, G., Baldi, V., Youssef, H., & Erra, U. (2023). Vico-dr: A collaborative virtual dressing room for image consulting. Journal of Imaging, 9(4), 76, https://www.mdpi.com/2313-433X/9/4/76

Özbay, E. (2013). Nesnelerin üç boyutlu modellenmesi için kinect tabanlı bir uygulama (Master's thesis, Fen Bilimleri Enstitüsü), https://acikbilim.yok.gov.tr/handle/20.500.12812/412048

Sekhavat, Y. A. (2016). Privacy preserving cloth try-on using mobile augmented reality. IEEE Transactions on Multimedia, 19(5), 1041-1049, https://ieeexplore.ieee.org/abstract/document/7782720

Srivastava T. (January 6, 2025). 12 Important Model Evaluation Metrics for Machine Learning Everyone Should Know (Updated 2025). AnalythicsVidhya. https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/

Tiwari, G., Bhatnagar, B. L., Tung, T., & Pons-Moll, G. (2020). Sizer: A dataset and model for parsing 3d clothing and learning size sensitive 3d clothing. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16 (pp. 1-18). Springer International Publishing, https://link.springer.com/chapter/10.1007/978-3-030-58580-8_1

Tunçer, E., & Gümüştekin, Ş. (2020, October). 3D object reconstruction using sequentially activated multiple depth cameras. In 2020 28th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE, https://ieeexplore.ieee.org/abstract/document/9302191

Vitali, A., & Rizzi, C. (2018). Acquisition of customer's tailor measurements for 3D clothing design using virtual reality devices. Virtual and Physical Prototyping, 13(3), 131-145, https://www.tandfonline.com/doi/full/10.1080/17452759.2018.1474082

Yoon, J. S., Kim, K., Kautz, J., & Park, H. S. (2021). Neural 3d clothes retargeting from a single image. arXiv preprint arXiv:2102.00062, https://arxiv.org/abs/2102.00062

Zheng, Z., Zhu, J., Ji, W., Yang, Y., & Chua, T.-S. (2022). 3D Magic Mirror: Clothing reconstruction from a single image via a causal perspective (arXiv:2204.13096). arXiv. https://arxiv.org/abs/2204.13096