 A picture containing text

Description automatically generated

**DASHFIT**

**AR Virtual Try-On for clothing items using Unity and Pose Detection**

This BS Project report is submitted to the Department of Computer Science as partial fulfillment of Bachelors of Science in Computer Science degree.

by

**Adeen Atif**

**Dania Ahmed**

**Mir Hamza Ali**

**Muhammad Saad Tariq**

Supervised by

**Dr. Syed Ali Raza**

Assistant Professor

Department of Computer Science

School of Mathematics and Computer Science (SMCS)

Institute of Business Administration (IBA), Karachi

Spring 2024

Institute of Business Administration (IBA), Karachi, Pakistan

Copyright: 2024, Adeen Atif, Muhammad Saad Tariq, Dania Ahmed, Mir Hamza Ali

All Rights Reserved

TABLE OF CONTENTS

[Acknowledgement vi](#_Toc168051978)

[1. Abstract vii](#_Toc168051979)

[2. Introduction 1](#_Toc168051980)

[3. Software Requirements 3](#_Toc168051981)

[3.1 Non-functional Requirements 3](#_Toc168051982)

[3.2 Functional Requirements Specification Use Cases 3](#_Toc168051985)

[4. Software Design 11](#_Toc168051986)

[4.1 Sequence Diagram 11](#_Toc168051987)

[4.2 Architectural Style 11](#_Toc168051988)

[4.3 System Internal Collaboration Diagram 12](#_Toc168051989)

[4.4 Overview Class Diagram 13](#_Toc168051990)

[4.5 Identifying Subsystems 14](#_Toc168051991)

[4.6 Persistent Data Storage 16](#_Toc168051993)

[4.7 Network Protocol 16](#_Toc168051996)

[4.8 Global Control Flow 16](#_Toc168051997)

[4.9 Hardware/Software Requirements 17](#_Toc168052000)

[4.10 Algorithms and Data Structures 19](#_Toc168052001)

[5. Conclusions and Future Work 21](#_Toc168052002)

[6. References 22](#_Toc168052003)

[Supplementary 23](#_Toc168052004)

# Acknowledgement

We are indebted to Dr. Syed Ali Raza and Mr. Arsalan Rashid for their invaluable guidance and support during the development of our final year project, DashFit (AR powered Virtual Apparel Try On). Their deep knowledge, constant encouragement, and insightful feedback were instrumental in shaping our project and helped us navigate its challenges. Their mentorship and dedication not only significantly enriched our learning experience but also played a pivotal role in the success of DashFit.

# Abstract

Buying clothes online and receiving ill-fitted items 7 or more days later is a problem In recent years, the sales of online clothing stores have skyrocketed. As the retail landscape shifts from traditional store-to-store shopping to online platforms, the need to create an enhanced and enjoyable user experience has become increasingly urgent. One of the challenges customers face when ordering clothes online is the item not looking as expected once they receive it and try it on.

The proposed system aims to address this issue by allowing brands to bring their store experience to the virtual world. Our mobile application is an AR-based virtual try-room that utilizes Pose tracking and Unity’s AR Foundation frameworks to allow users to ‘wear’ clothes before they purchase them online.

# Introduction

Buying clothes online and receiving ill-fitted items 7 or more days later is a problem compounded by the increasing popularity of online shopping. Our application-DASHFit- was developed with the objective of providing a solution to this dilemma we like to call ‘size surprise’. Within the scope of our project, we have focused on creating an Augmented Reality (AR) based virtual try room that allows online shoppers to see how a piece of clothing fits and moves with them.

Users can select clothing items of their liking, click the try room button, and stand in front of their camera to see the clothes overlayed on themselves. Our application offers a 180-degree view and includes arm movements for a comprehensive and realistic try-on experience. Our application allows a 180-degree view while also encompassing arm movements so users can have a holistic view of the clothes they try on.

A unique feature in our application is its depth perception mechanism without the need for sensors or other external hardware. Using Unity’s AR Core framework and Google’s MediaPipe for real-time coordinates of the person’s body, DASHFit gives users a realistic fit of their chosen items before they move to checkout.

At this stage, our application caters to different aspects of the try room experience such as ensuring visual cloth realism, fit of the clothes during movements while also including various models of shirts, such as long-sleeve and short-sleeve options, for both men and women in different sizes. However, it is outside the scope of our project to build the clothing models from scratch.

DASHFit adopts the following methodology to render clothes onto a person’s body in real time:

1. Coordinates are received from the MediaPipe plugin integrated with unity AR that helps identify the person’s position.
2. Pose detection places the 3D shirt onto the person.
3. Unity’s cloth physics feature enables the clothes to adjust according to body movements.
4. Visual cloth realism ensures the person is able to ‘wear’ the clothes realistically.

Ultimately, DASHFit seeks to enhance the online shopping experience, reduce the rate of returns and exchanges due to size issues, and contribute to customer satisfaction and confidence in their online clothing purchases.

**2.1 Objectives**

The main objective of this project is to give users the complete in-store shopping experience on online shopping platforms. DashFit will allow users to try their clothes on and find what size fits them best and avoid the hassle of return and exchanges. Our motivation stems from the limitations of current virtual try-on solutions, which either require external hardware such as depth sensors or are exclusively iOS-based, leveraging built-in depth sensors in newer iPhone models. The goal is to develop a solution for Android devices that does not require any external hardware. Thus, by harnessing the power of augmented reality and machine learning, we seek to bridge the gap between the convenience of online shopping and the assurance of traditional in-store try-ons.

# Software Requirements

## 3.1 Non-functional Requirements

3.1.1 Performance Requirements

The application’s implementation should be sufficiently optimized to ensure smooth, even visuals while viewing the augmented reality clothing. Program should run in a resolution that provides accurate display, enabling all created models and images to be realistic. The program should promptly provide visual response to any input provided by the user.

The performance requirements need to be met in order to accurately render the clothing with as high accuracy as possible.

3.1.2 Safety Requirements

DashFit is software compatible with most low-end mobile phones capable of running a unity mobile application. However, it should be ensured that the software does not consume a lot of memory or compromise other operations of the mobile system. Excessive load on the system can potentially cause overheating of the system, which may result in hardware damage.

## 3.2 Functional Requirements Specification Use Cases

**3.2.1 Browse Catalog**

***3.2.1.1 Description and Priority***

Browse Catalog is a basic function of the application. It is of High Priority since almost every other feature stems out from this one.

***3.2.1.2 Stimulus/Response Sequences***

**Step 1:** User opens the application DashFit

**Step 2:** Catalog is displayed

A diagram of a website

Description automatically generated

***3.2.1.3 Dependency/Assumption***

The primary condition to use this feature is to have the program running and to select the desired item.

**3.2.2 Select Item**

***3.2.2.1 Description and Priority***

The DashFit application allows you to select an item from the catalog using an onscreen click

***3.2.2.2 Stimulus and Priority***

**Step 1:** User opens the program DashFit

**Step 2:** The user browses the catalog using a swiping motion

**Step 3:** User clicks on a particular item

A diagram of a website

Description automatically generated

***3.2.2.3 Dependency/Assumption***

The primary condition to use this feature is to have the program running and to be present in the browsing catalog screen view. The user then selects the item they want to try on.

**3.2.3 View Item Description**

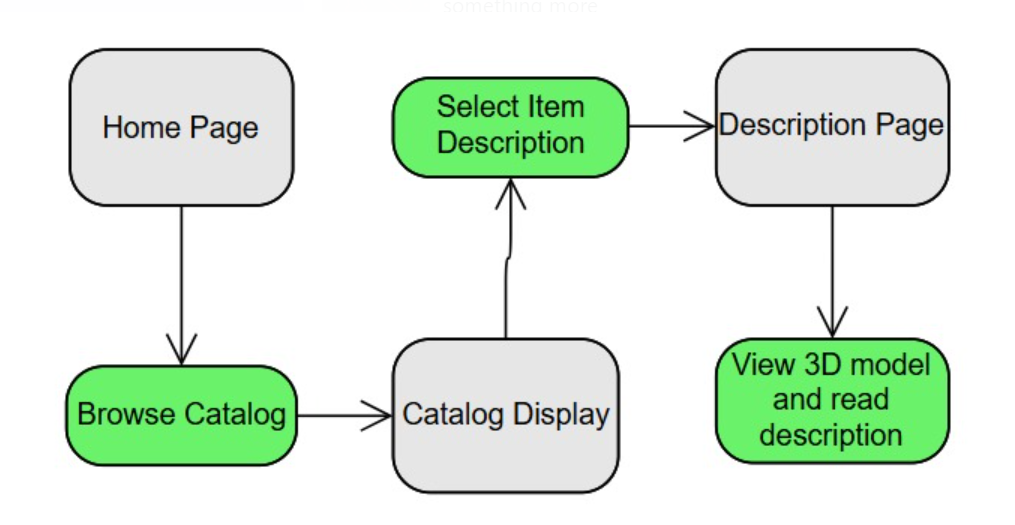
***3.2.3.1 Description and Priority***

Once an item has been identified by the user, the DashFit application will allow you to click on the item and direct you to an item description page where user can find all details about the chosen piece of clothing.

***3.2.3.2 Stimulus/Response Sequences***

**Step 1:** User opens the program DashFit

**Step 2:** User selects an item of their liking.



**3.2.4 View Virtual Clothing**

***3.2.4.1 Description and Priority***

When all preliminary conditions are met, the selected clothing will appear on the person and automatically fit according to where the individual is on the screen. You can point the camera towards a person, or you can point towards a mirror. The user will also be able to move and see how the clothes adjust according to these movements.

***3.2.4.2 Stimulus/Response Sequences***

**Step 1:** User opens the program DashFit

**Step 2:** Look for clothing item and select

**Step 3:** Look for body type and select

**Step 4:** Place camera in appropriate position and stand on marker

**Step 5:** View clothing item in augmented reality.

**A diagram of a product

Description automatically generated**

***3.2.4.3 Dependency/Assumption***

The primary condition to use this feature is to have the program running and to click on and select all necessary requirements for the clothing display.

**3.2.5 Select Size**

***3.2.5.1 Description and Priority***

The DashFit application allows you to try the item in different sizes using an onscreen click once in the try room

***3.2.5.2 Stimulus/Response Sequences***

**Step 1:** User opens the program DashFit

**Step 2:** Select an item from the catalog

**Step 3:** Goes to the try room

**Step 4:** Select the increase/decrease size button to adjust

A diagram of a product

Description automatically generated

**3.2.6 Select Multiple Items**

***3.2.6.1 Description and Priority***

The DashFit application allows you to select multiple clothes before entering the try room to allow for a better user experience.

***3.2.6.2 Stimulus/Response Sequences***

**Step 1:** User opens the program DashFit

**Step 2:** Select an item from the catalog

A diagram with green rectangles and white text

Description automatically generated**Step 3:** Select more items before clicking on try room button

**3.2.7 Switch Between Items**

***3.2.7.1 Description and Priority***

The DashFit application allows you to switch between the selected clothes in the AR try room to give a more realistic user experience.

***3.2.7.2 Stimulus/Response Sequences***

**Step 1:** User opens the program DashFit

**Step 2:** Select multiple items from the catalog

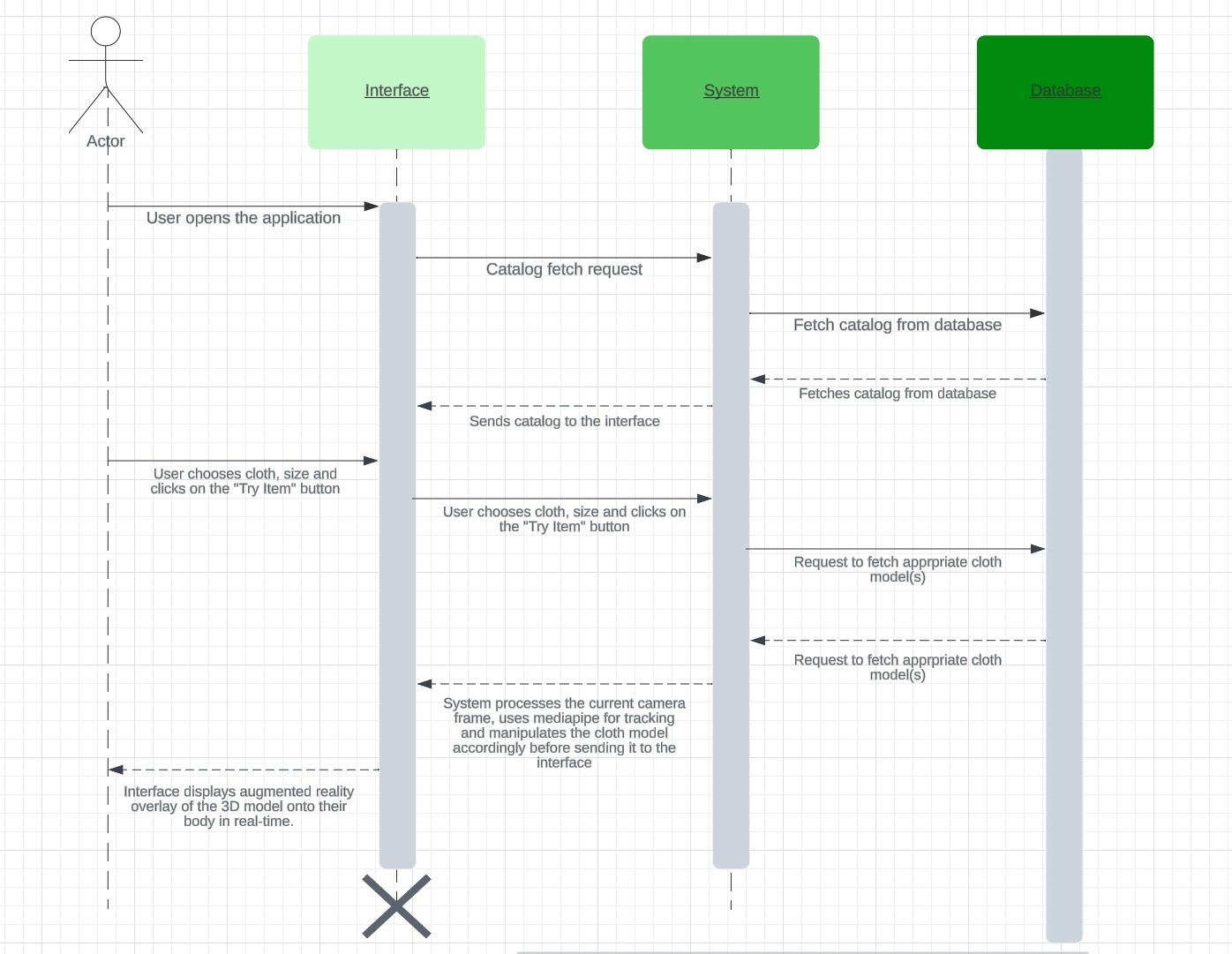
**Step 3:** Select try room button

A diagram with green rectangles and white text

Description automatically generated**Step 4:** Click on change item button to iterate between clothes

# 4. Software Design

## 4.1 Sequence Diagram



*Sequence diagram for the interaction of objects (interface, system, and database) within the system*

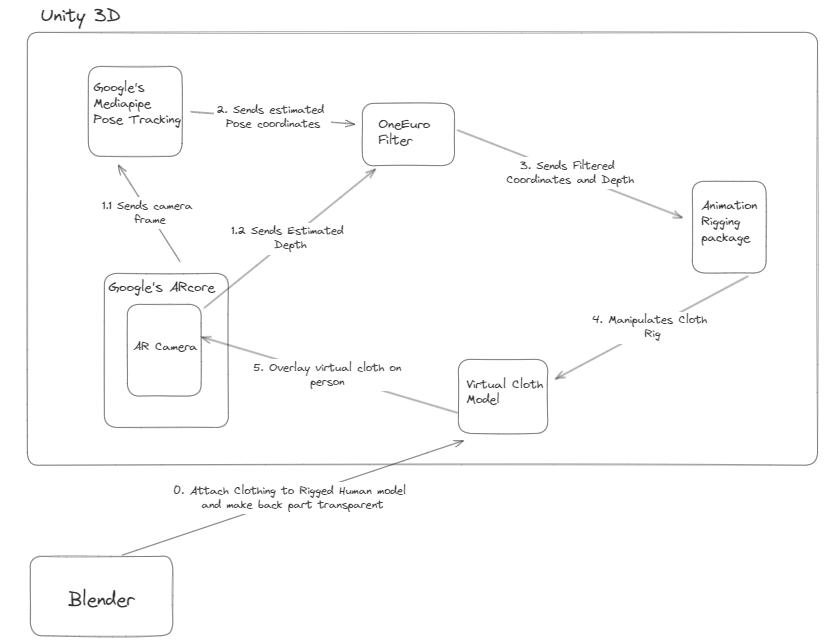
## 4.2 Architectural Style

Software Architectural Style is a hybrid between component-based architecture and event-driven architecture.

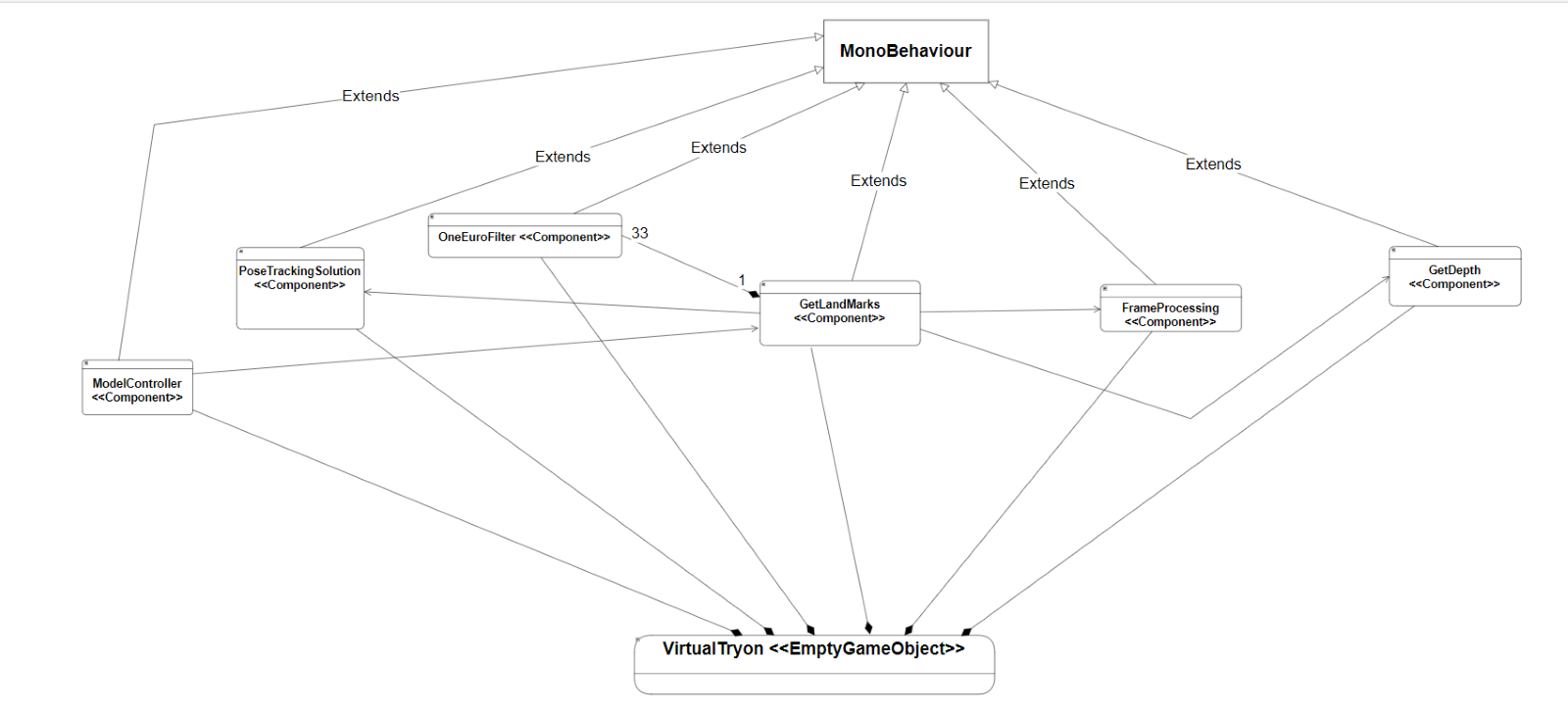
Unity extensively uses a component-based architecture. Our AR app consists of various components (like 3D models, scripts, AR tracking components) that are attached to game objects. These are loosely coupled, modular and reusable components.

Since our application works in real time, it also heavily relies on events (e.g., user interactions, tracking pose changes and movements). Functions are triggered in response to user interactions/movements or changes in the AR environment.

## 4.3 System Internal Collaboration Diagram

**

*Flow diagram to represent the working of the system sequentially.*

4.4 Overview Class Diagram*Class diagram consisting of the main components of the system*

*GetLandMarks receives filtered landmarks through the OneEuroFilter. These are them transformed to fit according to the domain they are to reside in and from where they originated. The ModelController then uses these landmarks for algorithmic movement of the rig.*

Note: “MonoBehaviour” is base class that many Unity scripts derive from. Details of it can be found here: <https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>

“VirtualTryOn” is a Unity Empty ‘GameObject’, it is only composed of the scripts in our application, and it does not have any properties of its own. The scripts need to be attached to a GameObject in order to run.

## 4.5 Identifying Subsystems

A diagram of a system

Description automatically generated

### **Component Description**

1. **UnityEngine**: This is a Unity package which contains multiple packages/subsystems such as Input, User Interface, Physics and Augmented Reality subsystems.
2. **XR**: Unity supports XR development on multiple platforms and devices through its XR plug-in framework. This XR package contains the packages required to build XR applications such as Augmented reality, Mixed Reality and Virtual Reality.
3. **ARFoundation**: The AR Foundation package contains interfaces for AR features, but doesn't implement any features itself. To use AR Foundation on a target platform, you also need a separate provider plug-in package for that platform like ARCore for android.
4. **AR Subsystems**: The AR-related subsystems are defined in this package. This package only provides the interface for various subsystems.
5. **Session Management:** Refers to an instance of AR and controls the lifecycle of all AR-related subsystems.
6. **Occlusion:** Responsible for achieving realistic-looking blending of augmented and real-world content by making sure that nearby physical objects occlude virtual content that is located behind them in the shared AR space.
7. **Light Estimation:** Responsible for dynamically adjusting virtual object lighting in AR scenes to match the real-world lighting conditions.
8. **Depth:** The depth subsystem is an interface into depth information detected in the scene.
9. **Camera:** The camera subsystem manages a hardware camera on the AR device.
10. **ARCore:** The ARCore XR Plug-in package enables ARCore support via Unity's multi-platform XR API. This package implements the AR subsystems present in the ARFoundation.
11. **Input System:** Allows users to control your game or app using a device, touch, or gestures.
12. **UI:** Responsible for displaying content of the AR world to the user.
13. **Physics System:** Responsible for simulating physics (cloth physics in our case) in project.
14. **Google’s Mediapipe:** This package contains the various solutions for perception-based tasks such as hand tacking, face detection, pose estimation and more.
15. **Pose Tracking System:** Responsible for human pose estimation. The pose landmarker model tracks 33 body landmark locations, representing the approximate location of the following body parts. It also provides human segmentation image.
16. **Filtering System:** Responsible for reducing jitter and lag in the coordinates detected by the pose tracking system. This contains OneEuro Filter which is a low pass filter and helps in reducing noise from the signals.
17. **Animation System:** Responsible for creating and controlling dynamic, interactive animations within Unity applications.
18. **Animation Rigging:** Provides advanced rigging tools and features for creating more sophisticated character/cloth animations. This is responsible for manipulating and adding constraints to the rig of the clothes’ models.

## 4.6 Persistent Data Storage

### **4.6.1 Data Storage**

Considering our current scope of application, we will not be using the database unless necessary, since a limited amount of rigged 3D cloth models are available. We will be storing our 3D models in the local file system of the application for now. In case of scaling and increasing number of models, we will transfer the models to Firebase cloud file storage because it has good integration support for Unity.

### **4.6.2 File Format Description**

Our 3D models are stored in the FBX file format. FBX files are a type of 3D model file created using the Autodesk FBX software. FBX files typically contain mesh, material, texture, and skeletal animation data. It encapsulates 3D models, animations, textures, and scene information, making it ideal for the diverse needs of our AR application. FBX files provide a comprehensive way to integrate detailed models and animations into the AR environment. They are particularly well-suited for virtual and augmented realities because they support a wide range of data types, including geometry, animation, skinning, and lighting. The FBX can be represented on-disk as either binary or ASCII data; its SDK supports both reading and writing. While neither of the formats is documented, the ASCII format is a tree structured document with clearly named identifiers. Aside from the FBX file, textures for the models will be stored as PNG files.

## 4.7 Network Protocol

System works offline for now. If we scale and expand, where our 3D models are stored on cloud, then internet will be required, in such case Firebase Storage operations, such as uploading and downloading files (including 3D models in formats like FBX), are performed over HTTP/HTTPS.

## 4.8 Global Control Flow

### **4.8.1 Execution Order**

Our application is primarily event-driven, responding to user inputs, AR tracking events and pose landmark and segmentation events. This approach is integral for interactive features like pose detection and AR rendering in real time. When a frame is received, an event is triggered by ARCamera. After the frame has been processed, it is given to the Mediapipe’s pose detection graph which raises an event if a human pose is detected. Similarly, ARCore’s depth and environment images raise event and then they are processed.

Certain initialization and setup processes follow a linear execution flow, ensuring that essential components (like ARCore initialization, model loading) are ready before user interaction begins. After human pose coordinates are detected along with depth information, instances of OneEuroFilter filters the coordinates, and Animation Rigging package manipulates the rig in linear execution order.

### **4.8.2 Time Dependency**

Certain processing and computations have been made time dependent for optimizing performance and resource usage in AR functionalities. Frame Processing, depth calculation and gradient (for arm movements) calculations are all subject to time dependency. Frame Processing is performed every 0.1 seconds (tentative), depth calculation is performed every 5 seconds and gradient calculation is performed every 0.1 seconds. This helps in reducing noisy signals and saves computational resource. Unity Coroutines have been used for time dependency. A coroutine allows you to spread tasks across several frames. In Unity, a coroutine is a method that can pause execution and return control to Unity but then continue where it left off on the following frame. Unity’s coroutine system is employed extensively for managing sequences that span multiple frames. Coroutines allow for creating delays and scheduling tasks without blocking the main thread.

**4.8.3 Concurrency**

The application operates on a single-threaded model, with all processes running on Unity's main thread so there are no concurrency issues. Given the single-threaded nature, careful attention is paid to optimizing performance and ensuring responsive interactions without overburdening the main thread.

## 4.9 Hardware/Software Requirements

**4.9.1 Hardware requirements:**

Mobile Device Specifications:

* + Processor: A modern, multi-core processor (e.g., Qualcomm Snapdragon 845 or equivalent).
  + RAM: minimum 4 GB
  + Camera: A working and good-quality rear-facing camera
  + Sensors: Gyroscope and accelerometer
  + Storage Space: A minimum of 500 MB free space could be necessary. (Note: this is assuming that the 3D models are not saved on the device but rather on a cloud file system or database. If the models are stored on the device then this storage can be increase significantly, depending upon how many we models we store).

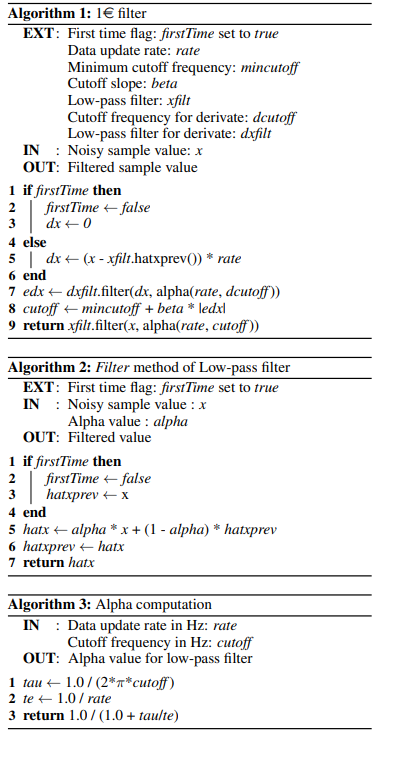
**4.9.2 Software requirements:**

Operating System: Android minimum API level: Android 7.0 'Nougat' (API Level 24) or higher.

Additional requirement: Device should also support ARcore Depth API. As of May 2023, approximately 89% of active devices support the Depth API and the list of supported devices can be found here: <https://developers.google.com/ar/devices>

## 4.10 Algorithms and Data Structures

**4.10.1 Algorithms:**



*OneEuroFilter Algorithm*

* + 1. **Data Structures:**

1. Arrays: These will be used for holding multiple components such as OneEuroFilter instances, 3d models and colors etc.
2. Vector3 (Unity Engine’s data structure): This data structure will contain float values of x, y and z coordinates which are being used for position and landmark coordinates.
3. Color (Unity Engine data structure): This data structure contains float values of R, G, B representing pixel values, used for frame/texture processing and texture manipulation.
4. Queues: This will be used to store pool of frames/textures.

# Conclusions and Future Work

The development of our AR virtual try-on application using Unity, integrating Mediapipe for pose tracking and ARCore Depth API for depth estimation, represents a significant milestone in accessible augmented reality technology. By harnessing these software tools, we've crafted a seamless virtual try-on experience that requires no external hardware beyond the user's smartphone, thus democratizing access to AR applications. Moreover, our focus on Android devices addresses a notable market gap, as iOS devices often include built-in depth sensors, unlike their Android counterparts. This application fills a crucial need for Android users seeking immersive AR experiences without additional equipment. Furthermore, the convenience of being able to utilize this technology from the comfort of one's home underscores its accessibility and utility. This project not only demonstrates the potential of AR technology in the retail and fashion sectors but also highlights the importance of innovation in making advanced technology easily accessible and inclusive for all users, even within the confines of their own homes.

Our application faces several limitations that guide our future improvements. Firstly, achieving 100% accurate depth and z coordinates without depth sensors remains a challenge, as current estimations facilitate correct cloth placement but struggle with accurately capturing z-axis arm movements, leading to occlusion errors during rotation. Secondly, mobile hardware constraints necessitate the use of restricted physics simulations and lite models for pose and depth estimation to ensure compatibility across various devices and prevent overheating issues on low-end devices. Thirdly, while our current cloth models lack adaptability to different body types due to time and expertise constraints with Blender, future iterations aim to address this limitation. Additionally, our focus on upper wear exclusively, such as shirts and kameez, restricts the variety of clothing types available. Finally, Mediapipe pose tracking limitations restrict the tracking to a single person's pose per camera frame, influencing the scope of our application's multi-person functionality.

In our future endeavors, we aim to broaden the scope of our current project, which presently focuses solely on upper wear, by seamlessly integrating lower wear clothing and other accessories. Additionally, our work holds potential for integration with augmented reality (AR) applications, facilitating immersive experiences such as AR games. Moreover, it offers the possibility of synchronizing virtual character movements in games or applications with those of the user, akin to the functionality seen in Xbox 360 Kinect games. Further enhancements could involve implementing precise body types and cloth sizes through tools like Blender, ensuring an even more tailored fit for garments. Furthermore, we envision extending the current application to incorporate features for cloth customization and wardrobe matching, enabling users to discover ideal clothing combinations effortlessly.

# References

|  |  |
| --- | --- |
| Dr.K. HelenPrabha, R. Ramyalakshmi, K. Renuka Devi, P. Sadhurya. n.d. "AR Based Try on Clothing." https://www.ijirt.org/master/publishedpaper/IJIRT151094\_PAPER.pdf.  Géry Casiez, Nicolas Roussel, Daniel Vogel. n.d. "1e Filter: A Simple Speed-based Low-pass Filter." https://inria.hal.science/hal-00670496/document.  Jessica Dias, Divya Chouhan, Preshit Churi, Pranay Parab. n.d. "Augmented Reality Based Virtual Dressing Room Using Unity3D." https://papers.ssrn.com/sol3/papers.cfm?abstract\_id=4111818.  Kim J-W, Choi J-Y, Ha E-J, Choi J-H. n.d. "Human Pose Estimation Using MediaPipe Pose and Optimization Method Based on a Humanoid Model." https://www.mdpi.com/2076-3417/13/4/2700.  Ohoud Alzamzami, Sumaiya Ali, Widad ALKIBSI, Ghaidaa KHAN. n.d. "Smart Fitting: An Augmented Reality mobile application for Virtual Try-On." https://www.researchgate.net/publication/372020005\_Smart\_Fitting\_An\_Augmented\_Reality\_mobile\_application\_for\_Virtual\_Try-On.  Prajakta Joglekar, Vinaya Gohokar. n.d. "Virtual Cloth Try-On Using Augmented Reality - Marker Based Approach." https://www.researchsquare.com/article/rs-1918747/v1.  Yiqiao Lin, Xueyan Jiao, Lei Zhao. n.d. "Detection of 3D Human Posture Based on Improved Mediapipe." https://www.scirp.org/journal/paperinformation?paperid=123355. |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Supplementary

**DashFit**

AR Virtual Try-On for clothing items using Unity and Pose Detection

Project Proposal

(7th Semester)

BY

Adeen Atif (22999)

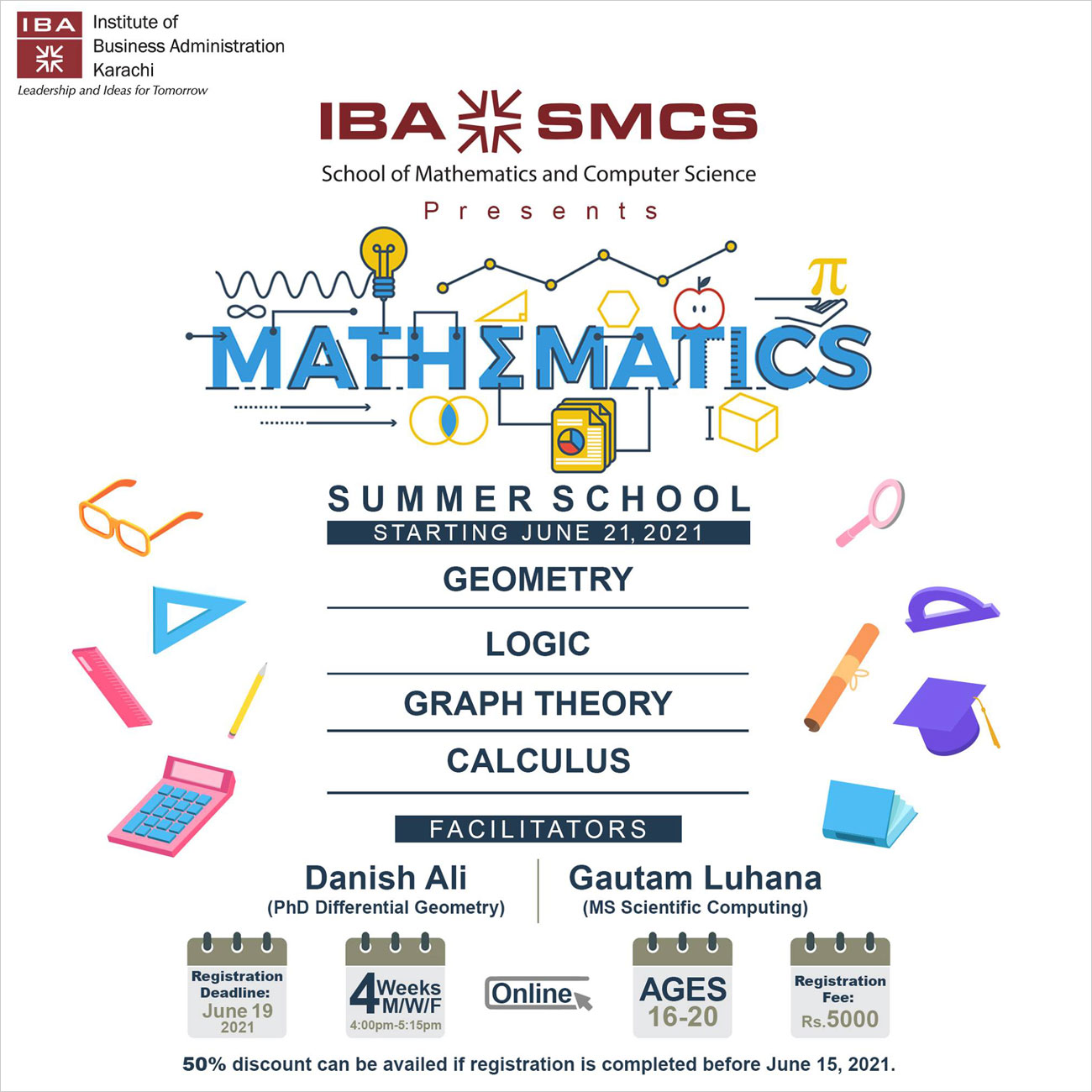
Muhammad Saad Tariq (22947)

Mir Hamza Ali (22767)

Dania Ahmed (22795)

ADVISOR: Dr. Syed Ali Raza

INDUSTRY MENTOR: Mr. Arsalan Rashid (Systems Ltd.)



**SUBMITTED TO**

**The FYP Committee**

Abstract

Our project proposes the development of an innovative mobile application that combines computer vision and augmented reality (AR) technologies to revolutionize the way users try on clothing. This AR-based system will allow users to virtually try on a wide range of clothing items from the convenience of their mobile devices. By leveraging 3D models and real-time camera input, users can visualize how various garments will look on them, ultimately enhancing their online shopping experience.

PROJECT OVERVIEW

|  |  |
| --- | --- |
| A | PROJECT TITLE: AR Virtual Try-On for clothing items using Unity and Pose Detection |
| B | ACRONYM: DashFit |
| C | TAGLINE: Say goodbye to size surprise |
| D | START DATE: 5th July 2023 |
| E | EXPECTED COMPLETION: 1st June 2024 |
| F | PROJECT BRIEF  Introducing our cutting-edge project, which aims to redefine the fashion retail experience. Our mobile application will serve as a virtual dressing room, offering users the ability to try on clothing items virtually. The clothing items included in the current scope of our project are tops and bottoms. Key technical features include:  - AR Visualization: Users can use their phone's camera to visualize the clothing on themselves, making informed purchase decisions.  - Computer Vision Integration: Our system will map clothing onto the user's body, ensuring a realistic fit and appearance.  - Realistic Interaction: Users can move, adjust, and explore the virtual clothing on their bodies, enhancing the accuracy of the try-on experience. |

PROBLEM STATEMENT

Recent years have seen the sales of online clothing stores skyrocket. According to Statista, the ecommerce fashion market was estimated to be US$752.5 billion in 2020**.** With this shift from traditional store-to-store shopping to online retail, the need to create an enhanced and enjoyable user experience has also become urgent. In this regard, one of the issues that customers of e-commerce fashion stores face when ordering clothes online is the item not looking what they expected it to once they receive it and try it on.

The proposed system will allow brands to bring their store experience to the virtual world by enabling users to visualize how a piece of clothing will look on them without having to physically try it on.

Our intended user base is business owners running e-commerce fashion sites and customers purchasing items from such websites. The former will have the role of admin while the latter will be considered a user of the system which can be integrated within the online shopping site.

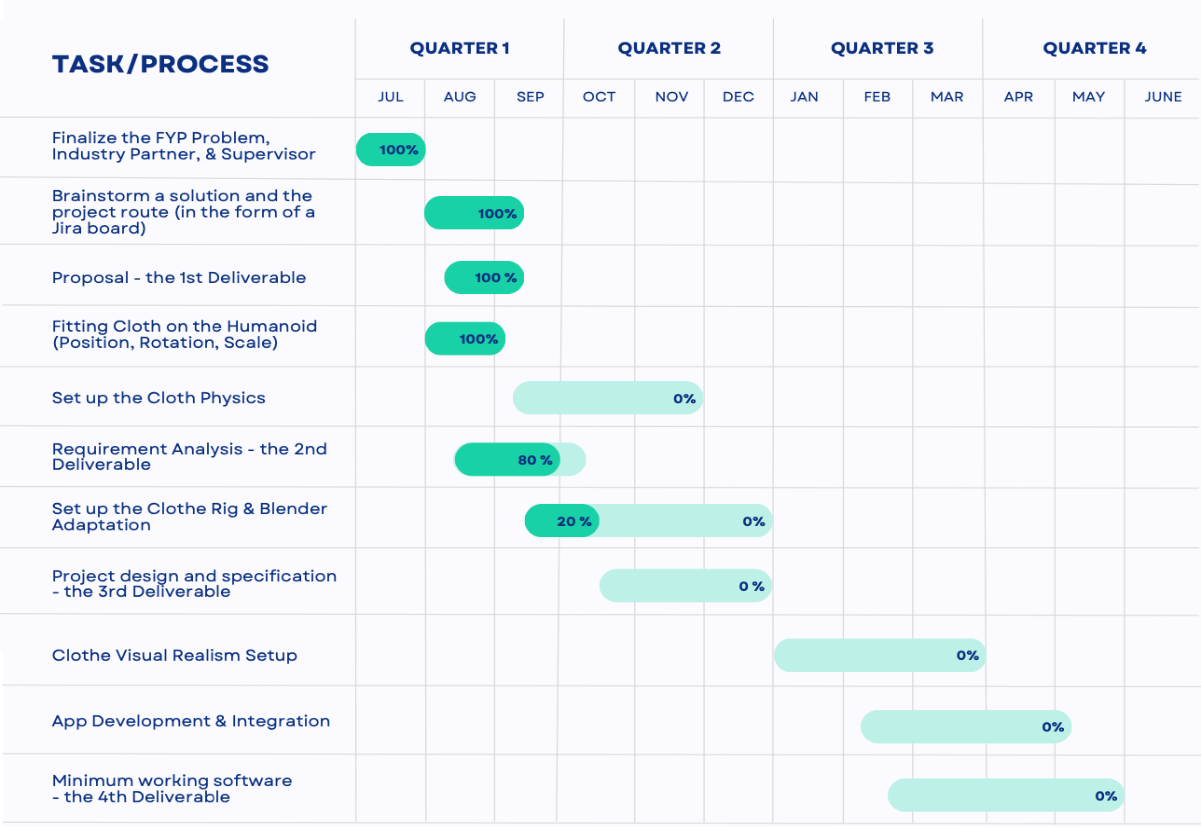
MARKET/LITERATURE REVIEW

|  |  |  |  |
| --- | --- | --- | --- |
| Existing Technology | Description | Inadequacies | Reference Image |
| Zero10’s AR Mirrors  And  Kinect for Windows for Retail Clothing | Mirrors installed in stores-such as the Tommy Hilfiger outlet in London- allow a customer standing in front of it to virtually try on the chosen piece of clothing. | * Only available in stores and not online * Hardware intensive and thus expensive * Cannot cater to multiple users simultaneously |  |
| YourFit by 3D look | This virtual fitting technology enables consumers to try on garments for size, fit, and style without physically wearing the product. As a camera-equipped device captures the customer, underlying AR technology maps a realistic virtual representation of the product over their real-world image to show how the product would look on the customer’s body | * Requires pictures as input rather than real-time video |  |
| Try on clothes by HB- Built using Snap AR | A snapchat lens created using Snap AR (Snap Inc.'s augmented reality platform that allows distribution on both iOS and Android) that allows the user to place a limited collection of clothes on themselves | * Lack of cloth visual realism- clothes appear rigid and animated * Not built for online shopping purposes and therefore cloth sizing or fitting not included |  |
| Smart Fitting: An Augmented Reality mobile application for Virtual Try On | An iOS mobile app utilizes the back camera of an iPhone to overlay a digital representation of clothing articles onto the user. The primary functionalities of the application encompass displaying a comprehensive array of available clothing categories, presenting details regarding colors and sizes for each garment, and offering users the capability to virtually try on clothing items. | * Occlusion not handled * Lack of cloth visual realism | (cited under references) |

PROJECT DETAILS

|  |  |
| --- | --- |
| A | PROPOSED SOLUTION:  Our project provides a solution to the common online shopping dilemma of uncertainty regarding clothing fit and style. By seamlessly integrating computer vision and AR technologies, our application offers a unique and user-friendly way for customers to confidently explore and virtually try on clothing items from the comfort of their homes.  Our distinct advantage lies foremost in the fact that we are designing this technology for online shoppers. With an easy-to-use interface, customers can conveniently check if a piece of clothing looks good on them or not. Added to that, we will be giving considerable attention to ensuring accuracy and realism of the virtual try-on experience, empowering users to make informed purchasing choices. |
| B | PROJECT OBJECTIVES:  **Integration of Virtual Attire onto Live User Image:** Develop the foundation for the application- overlay virtual clothing onto the real-time live image of the user, employing advanced augmented reality (AR) techniques  **Wardrobe Diversity:** Expand the system to encompasses a diverse array of clothing items and sizes, enabling users to choose and visualize various clothing items on themselves  **Mobile Application:** Design and construct a user-centric mobile application that incorporates interactive 3D models of clothing items, facilitating an immersive experience for users while utilizing the aforementioned technology.  **Realistic Fitting of Virtual Garments:** Ensure a realistic and visually accurate fitting of virtual clothing on the user's live image, enhancing the overall authenticity of the augmented reality simulation. |
| C | METHODOLOGY:  The following is our proposed methodology to implement the project described above along with the tools and technology we will be utilizing.   * Utilize **Unity AR foundation** and **ARcore** for building AR mobile application * Integrate **Google** **MediaPipe** plugin with unity AR * Use pose detection to place the 3D cloth model on the person * **Final Inverse Kinematics** or **Unity Animation Rigging** package for inverse kinematics and mapping human pose to cloth rig * Expand the functionality to a variety of clothing items and size range * Work on cloth physics- allowing it to adjust to the user’s movements * Ensure cloth visual realism- 3D models of the clothes are realistic and the user ‘wears’ it realistically in different lighting conditions * Design the UI for the mobile application * Develop the mobile application using **Flutter** and integrate the technology within it.   The project management methodology we intend to implement is a combination of agile and incremental methodology. Agile would be best suited considering it is particularly effective when dealing with projects that have a high degree of complexity and uncertainty. In our project, there are multiple aspects to consider, agile will allow us to adapt to changing requirements and address uncertainties as they arise. Similarly, using an incremental methodology would allow us to break the work into manageable increments as outlined in the bullets above. |
| D | THE PRODUCT:   * Mobile Application * User Manual |

PROJECT MILESTONES AND DELIVERABLES



WORK DIVISION

Mir Hamza Ali

* Initialize the AR app in Unity, utilizing MediaPipe for pose detection, plane detection, and ARCore depth API for realistic cloth fitting
* Fitting cloth on a humanoid, incorporating a variety of clothing models, and enabling position, rotation, and scale adjustments for a comprehensive fitting experience within a specified area.
* Oversee the “Cloth Physics” and “App Integration” modules of the project
* Build the ideal project architecture, leveraging the most effective combination of technologies, libraries, and packages to achieve the project's best outcome

Dania Ahmed

* Explore dynamic cloth movement techniques in Unity, synchronizing cloth rig with detected poses, and experimenting with animation rigging, predefined constraints, and ARKit's body tracking capabilities.
* Supervise the seamless integration of the Mobile Application to ensure a straightforward and trouble-free deployment process
* Create intricate wireframes and prototypes, leveraging industry-standard tools and techniques, to visualize and refine project design elements.
* UI/UX Designing of every interface that is going to be formulated in the project employing artistic flare to make the final product as user-friendly as possible

Adeen Atif

* Employ Blender and 3D modeling techniques to rig the humanoid model and its individual clothing items, enabling versatile size manipulation
* Integrate cutting-edge rendering technologies to enhance visual realism and optimize performance in Unity's 'Cloth Visual Realism' module.
* Tailor a project-specific Software Development Life Cycle (SDLC) framework to optimize workflows, enhance quality assurance, and drive project success.
* Spearhead the general management of the project, as the scrum master, be it setting strategic sprints, or being a point of contact between the team members

Saad Tariq

* Handle the “Cloth Visual Realism” module of the project in Unity, specifically targeting shaders with URP and HDRP render pipelines
* Oversee the “Cloth Rig” module of the project, ensuring the cloth rig moves according to the pose detected using Inverse Kinematics in Unity
* Oversee the DevOps aspects. Handling deployment and integration related workflows and assisting with code merges, branching strategies, and resolving version control-related issues.
* Employ Blender and 3D modeling to rig the imported models as well as ensuring compatibility with assigned weights according to the rig armature for multiple sizes

COLLABORATION

We have been able to secure a collaboration with Systems Ltd and have been working under the mentorship of Mr. Arsalan Rashid, VP-QA Services. As of August, we have conducted six successful meetings with him. Our first few meetings were based on discussing various project ideas and narrowing down to one based on our expertise, the scope of the project and its potential market value. The remaining meetings were aimed at working on a proposal, proof of concept and project plan. Throughout our meetings, Mr. Arsalan has helped us refine our idea and guided us on the best way forward.

COSTING

The current scope of our project does not entail any cost and can be accomplished with the resources we have and through packages and services available for free online. However, there is the potential of funds required if the resources currently available do not produce the intended results. This cost might be covered by Systems Ltd. The tentative cost is as follows:

|  |  |
| --- | --- |
| Package/Subscription/Service | Cost Range |
| 3D model | $5-$40 |
| Final Inverse Kinematics- for rig manipulation and inverse kinematics) | $90 |
| (For Cloth Natural Physics) | |
| Cloth Dynamics | $70 |
| Magica Cloth 2 | $25 |
| Obi Cloth | $47 |

REFERENCES

Schwarz, M. (2023, March 24). Try on clothes with AR effects at Hilfiger in London, Milan, Berlin. *MIXED Reality News*. <https://mixed-news.com/en/try-on-clothes-with-ar-effects-at-hilfiger/>

Dlook. (2023). Virtual try-on for clothing: the future of fashion? *3DLOOK*. <https://3dlook.ai/content-hub/virtual-clothing-try-on/>

*Try On Clothes Lens by HB - Snapchat Lenses and Filters*. (n.d.). [Video]. Snapchat. <https://lens.snapchat.com/8dd033a7a73b49d38f4c265e79a54842?share_id=yNTdnS6sQMo&locale=en-US>

ALZAMZAMI, O., ALI, S., ALKIBSI, W., KHAN, G., BABOUR, A., & BITAR, H. (2023). Smart Fitting: An Augmented Reality mobile application for Virtual Try On. *Romanian Journal of Information Technology and Automatic Control*, *33*(2).

APPENDIX

The following are four screenshots from the proof of concept followed by a snippet of the project plan that has been created on jira to ensure a smooth project flow.

