Clothing Virtual Try-On

Jorge Olivares

Robin Martinez

Alan Vela

**Final Report**

REVISION - 0.1

23 February 2022

**Table of Contents**

[**Figure 1. System Overview**](#_heading=h.2xcytpi)  **11**

[**Executive Summary**](#_heading=h.1ci93xb) **12**

[**Introduction**](#_heading=h.3whwml4) **13**

[Background](#_heading=h.2bn6wsx) 13

[Overview](#_heading=h.qsh70q) 13

[Referenced Documents and Standards](#_heading=h.3as4poj) 14

[**Operating Concept**](#_heading=h.49x2ik5) **15**

[Scope](#_heading=h.2p2csry) 15

[Operational Description and Constraints:](#_heading=h.147n2zr) 15

[System Description](#_heading=h.3o7alnk) 15

[Modes of Operations](#_heading=h.23ckvvd) 16

[Users](#_heading=h.ihv636) 16

[Support](#_heading=h.32hioqz) 16

[**Scenario(s)**](#_heading=h.1hmsyys) **17**

[Clothing Retail Industry](#_heading=h.41mghml) 17

[Personal Use](#_heading=h.2grqrue) 17

[Virtual Reality](#_heading=h.vx1227) 17

[**Analysis**](#_heading=h.3fwokq0) **18**

[Summary of Proposed Improvements](#_heading=h.1v1yuxt) 18

[Disadvantages and Limitations](#_heading=h.4f1mdlm) 18

[Alternatives](#_heading=h.2u6wntf) 18

[Impact](#_heading=h.19c6y18) 18

[**Introduction**](#_heading=h.nmf14n) **22**

[Purpose and Scope](#_heading=h.37m2jsg) 22

[Responsibility and Change Authority](#_heading=h.1mrcu09) 22

[**Applicable and Reference Documents**](#_heading=h.46r0co2) **23**

[Applicable Documents](#_heading=h.2lwamvv) 23

[Reference Documents](#_heading=h.111kx3o) 23

[Order of Precedence](#_heading=h.ctrl72p3hari) 24

[**Requirements**](#_heading=h.3l18frh) **25**

[**System Definition**](#_heading=h.206ipza) **25**

[Characteristics](#_heading=h.2zbgiuw) 26

[Functional / Performance Requirements](#_heading=h.1egqt2p) 26

[Video](#_heading=h.n9yc3bb2hns5) 26

[Analysis Time](#_heading=h.xvrobotmbdzr) 26

[Physical Characteristics](#_heading=h.3ygebqi) 26

[Android Phone](#_heading=h.zf66naqb1ilp) 26

[Software Requirements](#_heading=h.2dlolyb) 27

[Use of SMPL](#_heading=h.ej1qduu72f3q) 27

[Communications Requirements](#_heading=h.sqyw64) 28

[Database to Application](#_heading=h.gv8qb4qpxymy) 28

[Failure Propagation](#_heading=h.3cqmetx) 28

[**Appendix A: Acronyms and Abbreviations**](#_heading=h.43ky6rz) **29**

[**Appendix B: Definition of Terms**](#_heading=h.2iq8gzs) **29**

[**Overview**](#_heading=h.ayraj2odi7yz) **33**

[**References and Definitions**](#_heading=h.nkjxddk1nzhm) **34**

[References](#_heading=h.4my28wnr73s7) 34

[Definitions](#_heading=h.flyqrmpwa7j0) 34

[**Physical Interface**](#_heading=h.8hy1hr3fjmg1) **35**

[Model](#_heading=h.czpuo0lkz4tc) 35

[Camera](#_heading=h.jcbg6tggn4wo) 35

[**Database Interface**](#_heading=h.hl6y1cfhzl8c) **35**

[**Graphic User Interface**](#_heading=h.3hv69ve) **35**

[**Machine Learning Model to App Communication**](#_heading=h.1x0gk37) **35**

[**GUI Subsystem Introduction**](#_heading=h.j6sx3xxcr9zt) **41**

[Android Studio](#_heading=h.lbzn2s3qdt7b) 41

[**General GUI Description**](#_heading=h.j6sx3xxcr9zt) **41**

[Splash Activity](#_heading=h.tx53ikwue91o) 41

[Login Activity](#_heading=h.g9rs3h3heshj) 42

[User Profile Activity](#_heading=h.hc3n2nc3qmjq) 43

[Dashboard Activity](#_heading=h.45fzoumip1wt) 44

[Upload Video Activity](#_heading=h.ekgrx2b23p7k) 45

[Shopping Cart Activity](#_heading=h.ha9rh8xscnmt) 46

[Checkout/Order Details Activity](#_heading=h.bxjbeuhlfdbf) 47

[Sold Products Activity](#_heading=h.gmvl1fg0e1jg) 48

[**DataBase Interaction**](#_heading=h.7onkjjwili67) **48**

[Firebase](#_heading=h.vj50m7bgubzl) 48

[**Validation**](#_heading=h.w3j3kj4uhj1i) **49**

[Log in services](#_heading=h.uxt5jyg28zxl) 49

[Video Upload](#_heading=h.pvtx7r9h6cjy) 49

[**Subsystem Conclusion**](#_heading=h.u9m9x6hw6re9) **49**

[Further Development](#_heading=h.fb4289ne7p4g) 49

[**MongoDB**](#_heading=h.79i938gsm31b) **53**

[**Mongoengine**](#_heading=h.kmnvax3c2bew) **53**

[**Schema for Database**](#_heading=h.853pi6oe7gj) **53**

[Users](#_heading=h.owkk77557vsa) 54

[Clothing](#_heading=h.r5qg9pokk1go) 54

[**Validation**](#_heading=h.dcv7gfxhmp9b) **54**

[**Subsystem Conclusion**](#_heading=h.ixev3q2q9r0) **55**

[Further Development](#_heading=h.2zmd945p68yi) 55

[Integration with other subsystems](#_heading=h.4sxhrkcu5jzi) 55

[Horizontal Scaling](#_heading=h.fy8csj982s4) 55

[**Octopus**](#_heading=h.647h0brz3ogt) **59**

[SMPL](#_heading=h.q04884ps811e) 59

[Openpose](#_heading=h.gaeoadqbdopy) 59

[PGN Semantic Segmentation](#_heading=h.55ktg7v3duur) 59

[**Machine Learning Project Flow**](#_heading=h.78c2j0et64j2) **60**

[Data Processing Pipeline](#_heading=h.en84cygln4fv) 60

[Setting Up on the Olympus Server](#_heading=h.8m9vhhs7f277) 60

[**Validation**](#_heading=h.12s2x26g6xj2) **61**

[**Subsystem Conclusion**](#_heading=h.xyfmlhknbwnu) **61**

[Further Development](#_heading=h.67gbef68ym82) 61

[Integration with other subsystems](#_heading=h.d6jp9h1upzvs) 61

[Future Improvements](#_heading=h.mmgniiw7jsup) 61

**Execution Plan**

| **Android Application Subsystem** | **Cloud Database Subsystem** | **Machine Learning Subsystem** | **Date** |
| --- | --- | --- | --- |
| Learning Android Studio | Setting up AWS server host | Studying Python | 2/14/22 |
| Create Setup for GUI | Communicate with server Host | Verifying original Model runs - Dependencies | 2/21/22 |
| GUI calling for phone operational components | Study SQL | Verifying Original Machine Learning Model runs - Output | 2/28/22 |
| GUI receives information / Testing | Create SQL database on MySQL | Translating original model to Python 3 - Dependencies | 3/7/22 |
| Create log in services | Populate database | Translating original model to Python 3 -  Source Code | 3/14/22 |
| Establish host for log in services | Test database information retrieval. | Translation original model to Python 3 - Source Code | 3/21/22 |
| Test log in services | Create clothing designs. | Translation original model to Python 3 - Source Code | 3/28/22 |
| Connect GUI with neural network | Communicate with a neural network to provide design. | Testing for 3D model output | 4/4/22 |
| Test transfer of information | Testing for communication with neural network | Setting up Python-for-Android | 4/11/22 |
| Connect GUI with  database | Communicate with UI for User Login information | Testing communication between app and model | 4/18/22 |
| Test information storage and retrieval with databases. | Testing for communication with UI | Outputting 3D model to application | 4/25/22 |

**Validation Plan**

| **Test Name** | **Success Criteria** | **Status** | **Engineer Responsible** |
| --- | --- | --- | --- |
| Verify original model output | Dependencies must not interfere with one another.  Original Python 2 code must output a 3D model. | Untested | Robin M. |
| GUI information request | Success when receiving video information from an android device. | Untested | Alan V. |
| Access to storage Host | Able to save information on cloud | Untested | Jorge O. |
| Log in services | Successful login for the user. | Untested | Alan V. |
| Database information retrieval | Success if extraction of information is possible. | Untested | Jorge O. |
| Verifying new model output | Dependencies must not interfere with one another.  Python 3 code must output a 3D model. | Untested | Robin M. |
| Transfer of database information to neural network | Success when the neural network retrieves clothing designs from a database. | Untested | Jorge O.  Robin M. |
| Transfer of GUI information to database | Success when database gives access to user GUI login | Untested | Alan V.  Jorge O. |
| Communication between app and neural network | Successful when GUI is able to show 3D model to user | Untested | Robin M.  Alan V. |

**Change Record**

| **Rev.** | **Date** | **Originator** | **Approvals** | **Description** |
| --- | --- | --- | --- | --- |
| **0** | [02/1/22] | [Jorge Olivares]  [Alan Vela]  [Robin Martinez] |  | Draft Release |
| **1** | [02/6/22] | [Jorge Olivares]  [Alan Vela]  [Robin Martinez] |  | Revision 1 |
| **2** | [02/22/22] | [Jorge Olivares]  [Alan Vela]  [Robin Martinez] |  | Midterm Report |
| **3** | [04/29/22] | [Jorge Olivares]  [Alan Vela]  [Robin Martinez] |  | Final Report |

Clothing Virtual Try-On

Jorge Olivares

Robin Martinez

Alan Vela

**Concept of Operations**

REVISION - Draft

06 February 22

Concept of Operations

for

Clothing Virtual Try-On

Team <38>

Approved by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Jorge Olivares Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Kalafatis Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Skyelar Head Date

**List of Tables**

**List of Figures**

# **Figure 1. System Overview 12**

# **Executive Summary**

The Internet is a great tool that has provided a great number of innovations. Online shopping is a big part of it; however, it does create an issue. People don’t get to have a personal look at the object they are purchasing leading to dissatisfaction with the product when it finally arrives and then returning it. This cycle affects the apparel market heavily because Americans tend to return 10% of their purchases, but in apparel, 35% of products get returned. For companies, this is money being lost and environmentally it creates waste.

We are going to implement an application that is accessible to all public and helps reduce the quantity of returns done by customers shopping for clothes. The application is going to take a recording of the user. Once the video is uploaded, our application will create a 3D model of the user. This model is able to change the clothing that it has on, therefore we are able to create an opinion of how an outfit would look on us. Now that we are able to create a criterion of how clothes look on the customer without purchasing, people are going to make shopping decisions more accurately. We believe that this will help reduce the amount of returns.

# **Introduction**

In this document we are going to introduce the Virtual Clothing Try-on project. This project is going to be an application that has a wide range of utilities. Its versatility comes from the use of a simple RGB camera that is compatible with augmented reality. The camera is going to take a video recording of a user and turn it into a 3D model. Moreover, we are going to use the model to try-on different sets of clothing. Also, we are creating a database that will contain different clothing designs for all genders. In the very end, it will help with the reduction of returns to retail.

## ***Background***

Currently the process to verify that clothes fit is physically trying the clothes on. This process is not very efficient since it has a waiting time that can vary from a pair of days to even a month for clothing to arrive if bought overseas. This costs time for the customer and money to the company when the product is returned. Our main goal is that the application will allow for the customer to have a preview of the clothing on them and allow for a more informed decision when purchasing.

## ***Overview***

Figure 1 System Overview

In order for the application to work we decided to divide the system into two areas. The first area will be the neural network which is going to take care of the heavy lifting for the application. This neural network will process the video information. Once the information is processed by the system we will have a 3D model that will be stored on the android phone.

The second area of our system is the UI. This will be interacting with both the neural network system and the user. One area that will be interacting directly with the user is the front end, therefore we have to give a comfortable environment for the user. We want to ensure the user’s enjoyment when using the application. Another area of the UI is the back end. This area is mainly in charge of the database containing different clothing designs. Nevertheless, it will aid with the proper transfer of information between the neural network and the front end of the application.

## ***Referenced Documents and Standards***

* Alldieck,Thiemo;Magnor,Marcus;Xu,Weipeng;Theobalt,Christian;Pons-Moll,Gerard “Video Based Reconstruction of 3D People Models”
* https://www.mckinsey.com/industries/retail/our-insights/returning-to-order-improving-returns-management-for-apparel-companies

# **Operating Concept**

## ***Scope***

The Clothing Virtual Try-on application proposed in this document was designed to allow customers to freely choose clothing on a 3D model of themselves and get an opinion of how the clothes will look on them before purchasing. The customer will be able to choose between a selection of clothing having different designs, sizing, and colors of which they can see how it looks on their 3D model. This will help customers reduce the amount of returns when purchasing clothing.

## ***Operational Description and Constraints:***

Clothing Virtual Try-on application is intended to be used for people who shop online and companies that want to advertise their clothing. Several selections of clothing will be displayed for the user to choose from and will be projected on their 3D model. The user will be able to select clothing they are satisfied with and purchase it on the application.

The application must meet all the following criteria:

* The user must have an android AR compatible phone
* User must use backend camera for video
* User must use clothing that is fitting
* User privacy. The software side of the program must be able to handle 3D Model data without transmitting personal information
* The video needs to be at least a 24 fps, 30-second video that is colored

## ***System Description***

Our Clothing Virtual Try-on clothing application will consist of three main subsystems: the front end, backend, and neural network.

**Front End**: This is the main subsystem that will include the user interface (UI). This is the aesthetic area of the application of the user will see. This will allow the user to interact with the application on their phone. The UI enables display customization such as viewing objects that the user can interact with. This will include buttons and textboxes.

**Back end**: The back end of the application will be interacting with both the main UI to receive the information from the video recording and the neural network to process the information. However, the most important part is going to be establishing the database with the catalog of different clothing the model can change into.

**Neural Network**: Using a monocular RGB video of a person moving, a 3D model of them would be created. The neural network normalizes the videos data to 24 fps and a certain resolution. The model then does semantic segmentation in each frame to isolate the person into parts such as arms, legs, torso, etc. From there a skeleton is generated using preexisting work. The neural network model then generates textures onto the skeletal model. Finally, clothing textures taken from the clothing database are applied onto the 3D model.

## ***Modes of Operations***

The Clothing Virtual Try-on application will have a primary mode of operation when the user opens the application on the phone. In this mode, the user will use the backside camera of their android phone and upload a video to the application. The collected data from the user will be used to construct a 3D model of themselves via the Virtual Clothing Try-On neural network. With this model and the selection of clothing stored in our database, the user is presented with a model of themselves having a variety of different clothing to choose from.

The application will also have an error notification system where it would output a message to the user in the event of a problem that occurs within the database, internet speed, or simply the application itself. This will allow for the application to not all crash and let the user know when something is wrong to try again later.

## ***Users***

Our Clothing Virtual Try-on application will be marketed to people of all ages, but that mostly shop online. This will allow customers to be more comfortable shopping for clothing online as well as mitigate the need to return clothes after unsatisfactory purchases.

Our application can be also marketed to clothing companies that would want to implement their clothing using the application. This will create a great way of advertising their clothing and getting more purchases without the loss of money from returns.

## ***Support***

Support for the Clothing Virtual Try-on application will be provided on the application itself. This will include instructions on how to use the application, a video example of how to video record yourself for the 3D model using the back camera, and tech support to be able to contact.

# **Scenario(s)**

## ***Clothing Retail Industry***

The Clothing Virtual Try-On app will find its main use in the clothing retail industry scene. The app would allow companies such as Amazon to advertise clothes from their catalog to customers. This would eliminate the loss produced by customers returning clothing to their retailers.

Manufacturers of clothes and established brands can also use the app to advertise upcoming clothing lines.

## ***Personal Use***

The Clothing Virtual Try-On app could also be configured for personal use. Users would be able to store 3D models of all their articles of clothing in their wardrobe on a server. This would allow them to mix-and-match clothing and envision outfits they would like to wear for the day. Time wasted from trying on different outfits would be removed from the morning.

## ***Virtual Reality***

Using augmented reality from an application can help companies use virtual reality by using the 3D models created by the neural network. Companies such as Metaverse could use the data from the neural network to personalize models for individual users. This would allow more flexibility for the users to create an avatar that resembles themselves.

# **Analysis**

## ***Summary of Proposed Improvements***

The Clothing Virtual Try-On app will provide improvements such as

* Easier access to clothing catalogs of retail stores
* Fewer returns on unsatisfactory sales
* Selling upcoming lines, allowing for pre-ordering on clothes

## ***Disadvantages and Limitations***

The Clothing Virtual Try-On app will have limitations that include:

* Only works on android phones with ARCore compatibility
* 3rd party GPU sources must be used to train the model
* The video needs to be at least a 24 fps, 30-second video that is colored

## ***Alternatives***

Some alternatives to the Clothing Virtual Try-On app are:

* Conventional shopping and trying on clothes in-store
* Stores with lenient return policies that reduce loss from returns

## ***Impact***

If the app were to be successful this would have a major impact on the clothing retail scene. There have not really been successful applications for the construction of personalized 3d models and applying textures onto said models. Clothing chains would be able to retain a bigger audience for their product as they would be able to easily advertise their merchandise.

There would be privacy and security concerns in regards to the manner in which these personalized models are stored. Questions would be raised on the rights to these models.

Virtual Clothes Try-on

Jorge Olivares

Robin Martinez

Alan Vela

**Functional System Requirements**

REVISION – 0.2

19 February 2022

Functional System Requirements

for

Virtual Clothes Try-on

Prepared by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Author Date

Approved by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Jorge Olivares Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Kalafatis Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Skyelar Head Date

**List of Tables**

**Table 1. Assigned Responsibilities 21**

**Table 2. Applicable Documents 22**

**Table 3. Reference Documents 22**

**List of Figures**

**Figure 2. System Overview Flowchart 24**

# 

# **Introduction**

## ***Purpose and Scope***

The Virtual Clothes Try-on application is a more effective way of shopping online. It will provide a visual of how different clothes will fit. The document will help to identify the requirements for the Virtual Clothes Try-on project. An overview of the system and how it is divided is given in figure 1.

## ***Responsibility and Change Authority***

The team leader is going to be responsible for the approval of any changes done to the project and that the desired requirements are met in the provided deadlines. These changes need to be accepted by the sponsor before actual implementation. Subsystems responsibilities are shown in the following table.

| **Subsystem** | **Responsability** |
| --- | --- |
| 3D People Model application | Robin Martinez |
| UI Design and security | Alan Vela |
| Cloud Database | Jorge Olivares |

Table 1 - Assigned Responsibilities

# **Applicable and Reference Documents**

## ***Applicable Documents***

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

| **Document Name** | **Revision/Release Date** | **Publisher** |
| --- | --- | --- |
| Video Based Reconstruction of 3D People Models | 2018 | Computer Vision Foundation |
| Detailed Human Avatars from Monocular Video | 2018 | Computer Vision Foundation |
| MySQL Manual | 3.08 | Oracle Corporation |

Table 2 - Applicable Document

## ***Reference Documents***

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

| **Document Name** | **Revision/ Release Date** | **Publisher** |
| --- | --- | --- |
| People Snapshot  Datasheet | 2018 | Computer Vision Foundation |
| semantic human texture stitching | 2018 | Computer Vision Foundation |

## ***Order of Precedence***

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings or other documents that are invoked as “applicable” in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

# **Requirements**

This section is going to identify the minimum requirements needed for the application to work. These constraints were decided in an attempt to create a comfortable environment for the user and good reliability.

# **System Definition**

The virtual clothes try-on system will be split into three different subsystems: Neural Network, UI application, and Cloud Database. The neural network is going to be the main focus of the application since it is going to create the 3D model of the user with a video recording. Additionally, it will allow the user to change the textures of the model allowing different designs on the avatar. The UI application will interact directly with the user of the application. This subsystem has to ensure user satisfaction and be easy to use. Moreover, it will take care of the security of the user when their information is used. Lastly, we have the cloud database. The database is going to store different designs of clothing the user will be able to change the 3D model. Also, it will store the 3D model with the information of the user. Preventing the user from having to record themselves every time they want to use the application.

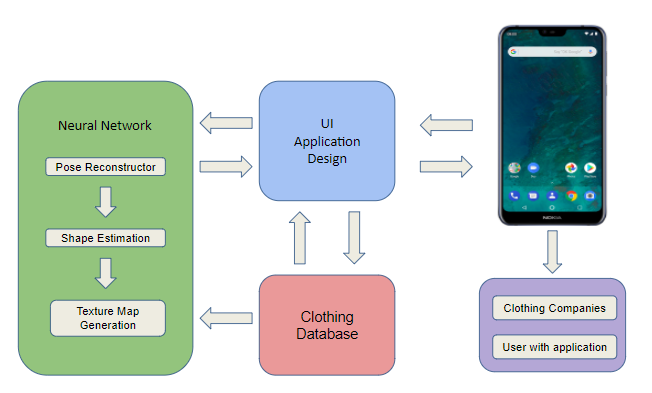


Figure 2 - System overview flowchart

## ***Characteristics***

### **Functional / Performance Requirements**

#### **Video**

The video rendered by the neural network has to be at least 24 frames per second, 30-second video, and color.

*Rationale: This is a core system performance requirement that the software requires to render the video.*

#### **Analysis Time**

The total time needed to evaluate the video to render a 3D Model should not exceed 45 seconds.

*Rationale: The time needed to render a 3D model should not take a long time for the user to see a result.*

* + - 1. **Database Size Requirements**

The database should be able to hold 50 to 100 different textures of shirts and trousers

*Rationale: Specified by sponsor*

### **Physical Characteristics**

#### **Android Phone**

The user has to have an android phone that can use OS 7.0 Nougat that is compatible with Augmented Reality applications. A back camera is required.

*Rationale: This is a requirement specified by the sponsor.*

### **Software Requirements**

#### **Use of SMPL**

The detection and characterization algorithms will use the SMPL model to track the pose of the User

*Rationale: A learned model of human body shape and pose-dependent shape variation that is compatible with the Python coding language*

* + - 1. **Programming Language - App Subsystem**

The Android application will be coded using Java

*Rationale: Due to constraints set when one is using Android Studio to design an app*

* + - 1. **Programming Language - Machine Learning Subsystem**

The GitHub repository on the “Detailed Human Avatars from Monocular Video” will be translated from Python 2.6 to python 3.6

*Rationale: Specified by the team sponsor due to the constraints of the report this project is building off of*

* + - 1. **Programming Language - Database Subsystem**

The database for User accounts and clothe textures will be written in SQL

*Rationale: The Python language is a cross-platform language that is easy to pick up for novice programmers.*

* + - 1. **Database**

The database will be constructed using MySQL

*Rationale: MySQL is an open source relational database management system.*

* + - 1. **Android Application**

The application will be designed using Android Studio

*Rationale: An integrated development environment for Google’s Android operating system that is very customizable for ease of use.*

### **Communications Requirements**

#### **Database to Application**

Python will be used to contact amazon storage services. The database is going to use SQL to have the proper organization of queries. Queries will be done in python to retrieve or store data from the SQL database.

#### **Failure Propagation**

Any exceptions are caught within the script of code and will throw errors to be

read by the user if one or all the subsystems is down. All data will remain safe in the database and a system restart should be prompted to the user. System failures caught in exceptions will be analyzed for future resolutions.

# 

# 

# 

# 

# 

# 

# 

# 

# 

# **Appendix A: Acronyms and Abbreviations**

MySQL My Structured Query Language

SQL Structured Query Language

GUI Graphical User Interface

RGB Red Green Blue

SMPL Skinned Multi-Person Linear Model

3D 3-dimensional

IP Internet Protocol

# **Appendix B: Definition of Terms**

Neural Network - A series of algorithms that attempt to understand relationships between data sets that they receive

Database - An organized collection of data stored on a computer

Monocular - With, for or in one eye

Augmented Reality - Interactive experience of a real world environment replicated by a computer

Internet Protocol - Method by which data is sent from one computer to another

Virtual Clothes Try-on

Jorge Olivares

Robin Martinez

Alan Vela

**Interface Control Document**

REVISION – 0.2

19 February 2022

Interface Control Document

for

Virtual Clothes Try-on

Prepared by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Author Date

Approved by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Jorge Olivares Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Kalafatis Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Skyelar Head Date

**List of Tables**

**Table 4. Reference Documents 33**

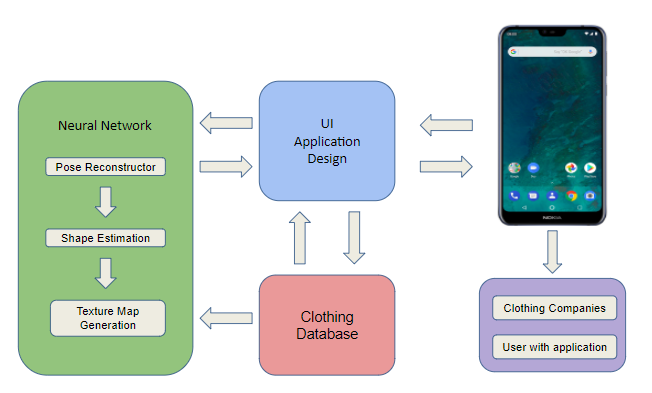
**List of Figures**

**Figure 3. System Overview Flowchart 32**

# 

# **Overview**

This document describes the interfaces between all the different subsystems of Virtual Clothing Try-On. The system is designed to be centered on three different subsystems: the android application, database, and neural network. A system diagram of how the different subsystems interact is shown in the **figure ?.**

Figure 3 - System overview flowchart

# **References and Definitions**

## ***References***

| **Document Name** | **Revision/ Release Date** | **Publisher** |
| --- | --- | --- |
| People Snapshot  Datasheet | 2018 | Computer Vision Foundation |
| semantic human texture stitching | 2018 | Computer Vision Foundation |

**Table 4 - Reference Documents**

## ***Definitions***

MySQL My Structured Query Language

GUI Graphical User Interface

RGB Red Green Blue

SMPL Skinned Multi-Person Linear Model

3D 3-dimensional

APK Android Package

IP Internet Protocol

# **Physical Interface**

## ***Model***

An Android Phone that can use OS 7.0 Nougat and that is also compatible with Augmented Reality applications is needed to provide a video able to create the 3D model.

## ***Camera***

The phone has to use a back camera that can capture video in RGB. The camera needs to process videos that are at least 24 frames per second and colored.

# **Database Interface**

The subsystems will interact with MySQl database through an IP established in the host services. Data requests will be managed with Python to ensure flexibility between the subsystems. Each of the systems will be able to communicate with the database independently.

# **Graphic User Interface**

The user will interact with the GUI on their android phone that was created on Android Studio using Java. Users will be able to login and have a profile with information they will be able to edit on. There will also be a shopping cart that the user will be able to add or remove items they want to purchase. This allows the user to freely and conveniently use the application.

# **Machine Learning Model to App Communication**

For communication between the SMPL model and the app interface we will be using Python-for-Android. This would allow us to package the machine learning code into android APKs that can be uploaded to the Android app store. Python-for-android would compile the Python interpreter with its dependencies, backend libraries and python code.

Virtual Clothes Try-on

Jorge Olivares

Robin Martinez

Alan Vela

**Subsystem Report**

REVISION – 0.3

29 April 2022

Subsystem Report

for

Virtual Clothes Try-on

Prepared by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Author Date

Approved by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Jorge Olivares Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Kalafatis Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Skyelar Head Date

Virtual Clothes Try-on

Jorge Olivares

Robin Martinez

Alan Vela

**GUI Subsystem Report**

REVISION – 0.3

29 April 2022

Subsystem Report

for

Virtual Clothes Try-on

Prepared by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Author Date

Approved by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Alan Vela Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Kalafatis Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Skyelar Head Date

**List of Figures**

**Figure 1 40**

**Figure 2 41**

**Figure 3 41**

**Figure 4 41**

**Figure 5 42**

**Figure 6 42**

**Figure 7 43**

**Figure 8 44**

**Figure 9 45**

**Figure 10 46**

**Figure 11 46**

**Figure 12 47**

**List of Tables**

# **GUI Subsystem Introduction**

The GUI is the main part of the subsystem where the user interacts with. This is what the user will be seeing and is considered the front end subsystem.

## **Android Studio**

The reason for choosing android studio is because it provides a complete IDE with code editor and app templates to use for users. It also allows the user to use tools for development, debugging, testing, and performance that make it faster and easier to develop apps. I find it the easiest to use as well.

# **General GUI Description**

The GUI Phone Application is separated by different activities (window screens which the app draws its UI) where the user is able to see and interact with.

## **Splash Activity**

The splash activity is a display to the user with a delay of 1.5 seconds when opening the app on the phone. This will allow the user to know that the application is opening and shows the user the phone logo/name. Figure 1 below shows the splash screen.

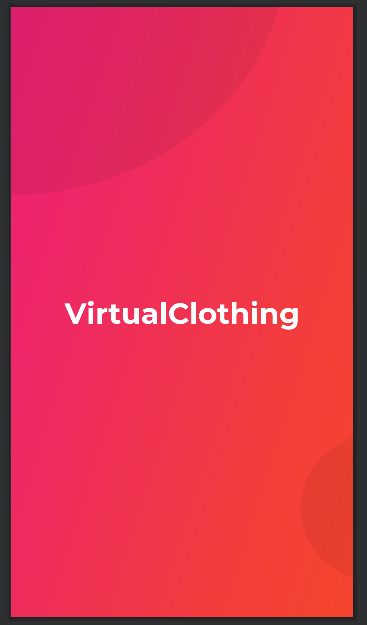


Figure 1 - Splash Screen

## **Login Activity**

The login activity is a display to the user where the user is allowed to login, register, and forget password. This will give access to the user with the correct credentials to be able to enter the main display of the application. If the user is not able to correctly input information it will prompt the user to input that information or it won’t register. This can be said if the user inputs the same email it will prompt the user to input another email. Figure 2 will show the login screen. Figure 3 will display the register screen. Figure 4 will show the forgot password.

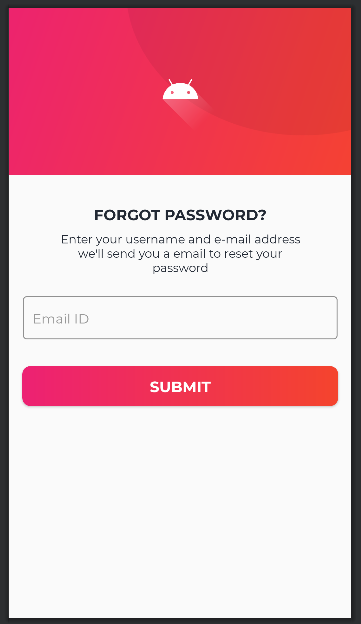
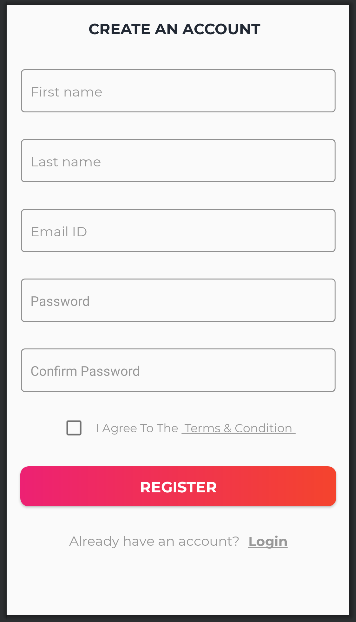
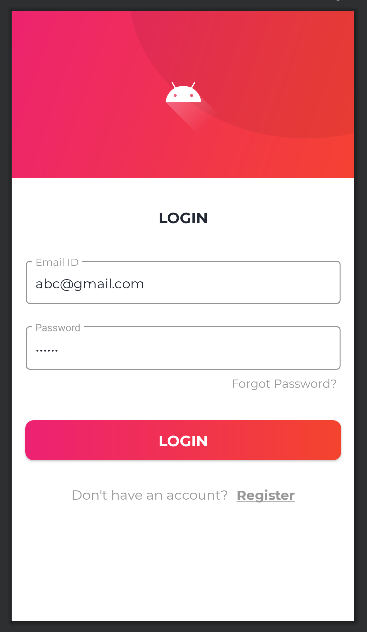


Figure 2 - Login Screen Figure 3 - Register Screen Figure 4 - Forgot Password Screen

## **User Profile Activity**

When logging in for the first time, the user will be prompted to enter information for their profile. This will allow the user to customize their profile and edit it when it seems fit. This will also include a section where the user can add their address on their profile so when purchasing an item it will be sent there. Figure 5 will display the profile information. Figure 6 will show the add address information.

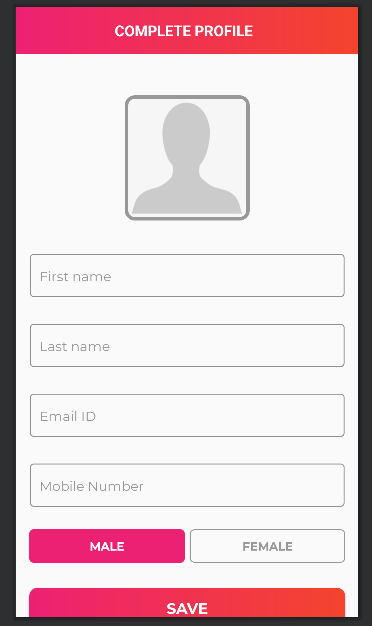
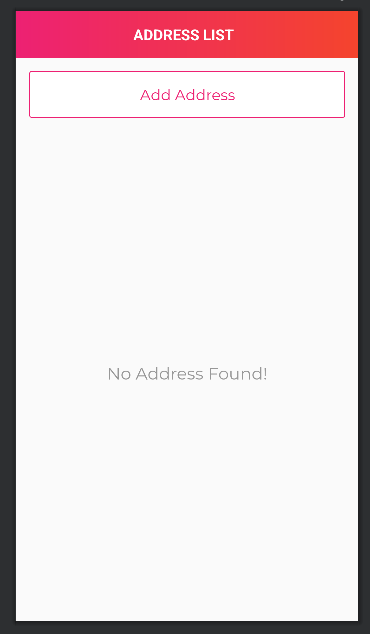
 

Figure 5 - Profile Information Screen Figure 6 - Add Address Screen

## **Dashboard Activity**

After the user logins and inputs profile information the user will be prompted on the dashboard screen. This is where the user will be displaying the main part of the GUI. It will be showing clothing to be selected and chosen from. It will also allow the user to select different screens to be displayed including the orders, products, and sold products. Figure 7 shows the dashboard screen.

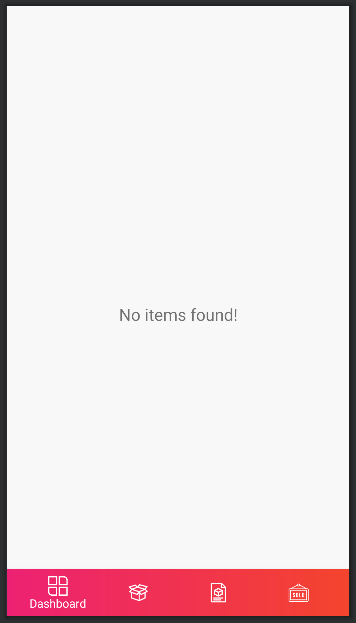


Figure 7 - Dashboard Screen

## **Upload Video Activity**

The user will be able to upload a video of themselves. They will have the option of using the phone camera to upload a video or the phone’s gallery. The video will be displayed to show the user that the video is rendered. This video information will be used to create a 3D avatar of the user. Figure 8 shows the upload video screen.

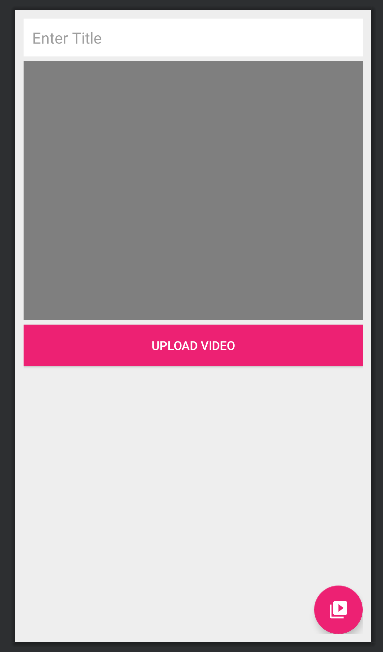


Figure 8 - Upload Video Screen

## **Shopping Cart Activity**

The shopping cart activity will show the user if any products they selected will be saved to be able to purchase. It will also show the user the subtotal, shipping charge, and total amount of the number of products they selected for checkout. Figure 9 shows the cart screen.

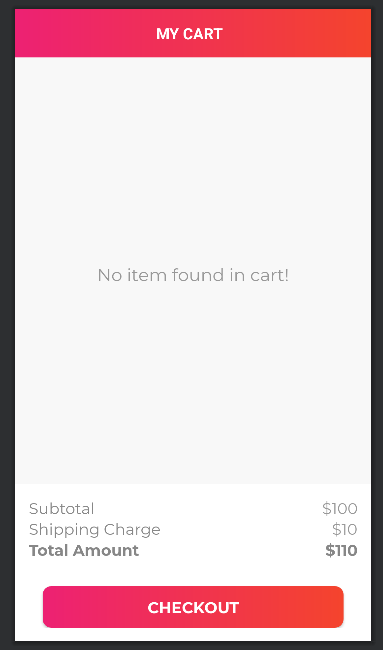
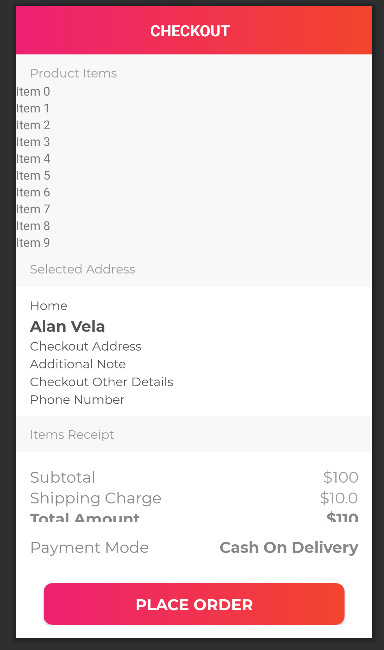
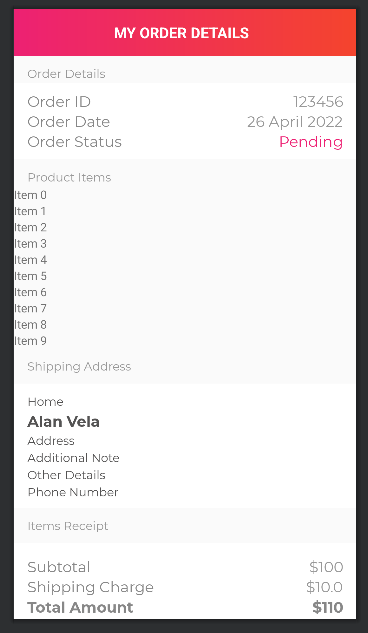


Figure 9 - Shopping Cart Screen

## **Checkout/Order Details Activity**

The checkout screen will show the number of items that the customer ordered and able to display the details of what they ordered to them. It will also show how you will order with cash or credit card. As for the order details screen it will display the order ID and status of the items you selected. Figure 10 shows the checkout screen. Figure 11 shows the order details screen.

  Figure 10 - Checkout Screen Figure 11 - Order Details Screen

## **Sold Products Activity**

The sold products screen will show the user the products that you have sold. This app is an ecommerce/shopping so the user will be able to sell their products if they wanted to. Figure 12 shows the sold products screen.

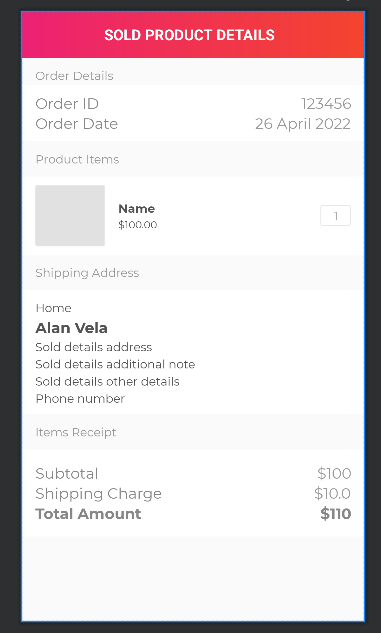


Figure 12 - Sold Products Screen

# **DataBase Interaction**

The GUI will be interacting with MongoDB, however for testing purposes I was using firebase.

## **Firebase**

Firebase is a realtime database that allows secure access to the database directly. Data is stored locally so even when offline the data will still be on fire giving the user a responsive experience. Firebase was the easiest and most known to use with android studio for testing purposes.

# **Validation**

## **Log in services**

Validation was performed for the login services by requesting data from the user, storing that data into firebase and accepting that data to login on the application. This was successful and any user is able to register, login, and forget password if need to.

## **Video Upload**

Validation was performed for the video upload by requesting a video from the user either from the gallery of the phone and use the camera of the phone to take a video of the user. Firebase once again was used to store the video in storage and able to display the user with the video of themselves to show that it was uploaded. This was successful.

# **Subsystem Conclusion**

In conclusion, the GUI subsystem was successful and completed in time as expected.

## **Further Development**

* + 1. **Integrating with Other SubSystems**

For most of the GUI, to be able to test I had to use a backend database which was firebase to able to see if anything worked. In 404, the GUI subsystem will have to change to mongoDB and use machine learning to create the 3D avatar.

* + 1. **Creating 3D Avatar**

The GUI subsystem will have to be able to read an obj file that the octopus outputs and display a 3D avatar to the user. This will be done in 404 working with the machine learning subsystem.

* + 1. **Creating Clothing Product**

The main part of the subsystem will be displaying clothing product to the user in the application to shop. A lot of the clothing product is already in process and will be soon completed in 404.

Virtual Clothes Try-on

Jorge Olivares

Robin Martinez

Alan Vela

**Database Subsystem Report**

REVISION – 0.3

29 April 2022

Subsystem Report

for

Virtual Clothes Try-on

Prepared by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Author Date

Approved by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Jorge Olivares Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Kalafatis Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Skyelar Head Date

**List of Figures**

**Figure 1 42**

**List of Tables**

**Table 1 44**

**Table 2 44**

# 

# 

# **MongoDB**

MongoDB is a NoSQL database that doesn’t need a relation when storing data on it. It can assist with creative and non-conventional data storage. Furthermore, even though it is not a strict non-relational database it is still very good at managing high volumes of data at very good speeds. This is the database selected for the project since it can manage information storage files of any kind and it has tools that can allow for files greater than 16MB to be stored. The scalability for the deployment of a project is very easy because the cloud offers different storage options at their respective rates.

## **Mongoengine**

This tool is an ORM(object-relational mapping) for working with MongoDB while using python script. The interaction between the database and the script is very smooth since this is a high-level tool it uses a declarative API to work.

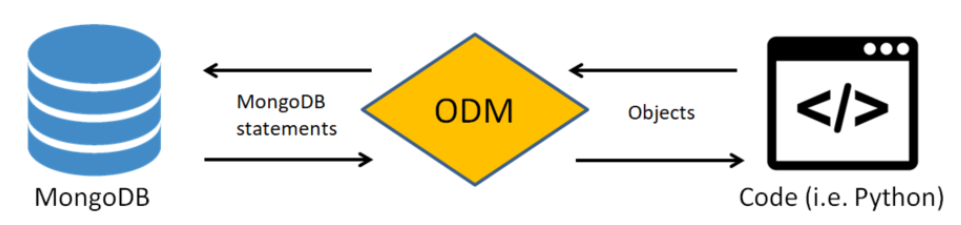


Figure 1: Mongoengine process of operation

# **Schema for Database**

In MongoDB the database is set up by a set of collections. Inside each collection, you can decide what kind of data is stored for later use. These collections have a schema layout that simplifies the organization process of information. The use of python classes with mongoengine creates a schema that doesn’t flag any issues with the database.

## **Users**

The user collection contains the information that is going to be provided by the user. The information is going to be the username and password. Only one 3D model avatar is going to link with the user itself.

| Tag | Description |
| --- | --- |
| User ID | Unique ID for user information storage |
| Username | Username provided by user |
| Password | Password matching the username |
| Model | Path storage for 3D model |

Table 1: User information stored in the database

## **Clothing**

The collections for clothing are going to contain the information needed for the texture to be retrieved from the database and the name provided whenever the file is uploaded to storage.

| Tag | Description |
| --- | --- |
| Clothing ID | Unique ID for clothing storage |
| Name | Name provided for a piece of cloth |
| Texture | Path to the storage of texture for cloth piece |

Table 2: Clothing information stored in the database

# **Validation**

Created a series of queries that will upload and retrieve the information to and from the database. These queries displayed the proper data and files requested from the database. Once the file was downloaded and opened on a different computer to verify that it worked and wasn’t corrupted.

# **Subsystem Conclusion**

The subsystem was completed on plan and met the requirements set in the validation process.

## **Further Development**

### **Integration with other subsystems**

The settings for fully automated queries match the inputs provided by the GUI. The GUI is also going to be able to request the information and scale it as needed for display. This will take care of the inputs needed for the GUI. Moreover, the machine learning subsystem will need to communicate with the database to store the model avatar created, for the user to have access to whenever this information is needed.

### **Horizontal Scaling**

The number of documents provided for the development of the application are subject to change whenever the application is ready for deployment. However, MongoDB provides tools to facilitate the upgrading storage process. The use of JSON in order to store documents and the use of nodes for data storing facilitates the upscale of storage.

Virtual Clothes Try-on

Jorge Olivares

Robin Martinez

Alan Vela

**Machine Learning Subsystem Report**

REVISION – 0.3

29 April 2022

Subsystem Report

for

Virtual Clothes Try-on

Prepared by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Author Date

Approved by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Jorge Olivares Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Kalafatis Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Skyelar Head Date

**List of Figures**

**Figure 1 42**

**List of Tables**

**Table 1 44**

**Table 2 44**

# **Octopus**

Octopus is convolutional neural network model that predicts human shape. With just a few frames taken from a video in which the subject slowly turns, it is able to create a model within 4 to 5mm accuracy. Using semantically segmented images and a skeletal model, Octopus produces a clothed model based off SMPL, a learned model of human body shape and pose-dependent shape variation.

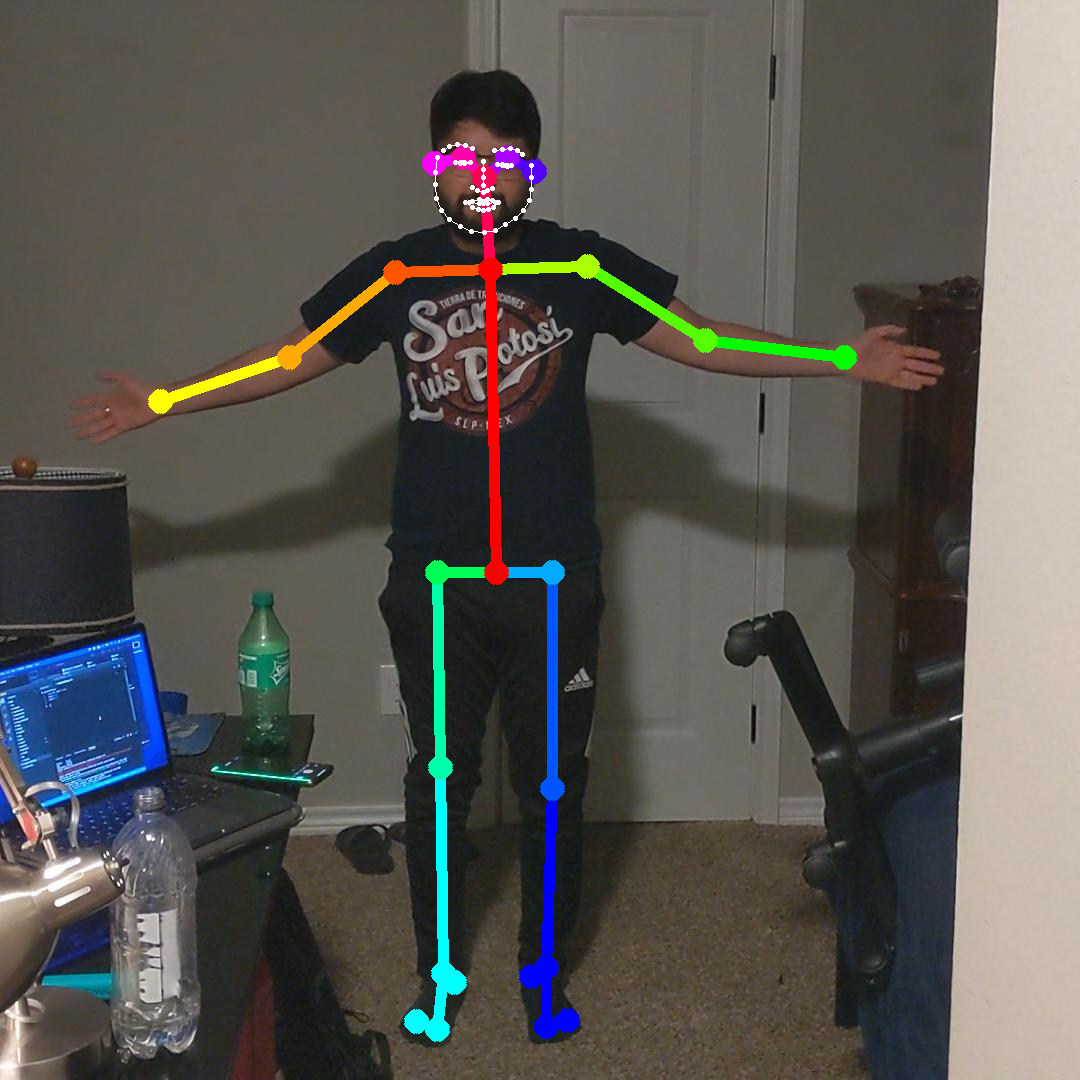
Octopus is written around depreciated libraries and APIs from 2019. Therefore, dependencies were either updated or replaced to eventually achieve a working model.

## **SMPL**

Skinned Multi-Person Linear Model is a skinned vertex-based model Octopus is built off of. It can produce various body shapes in natural human poses as it’s parameters can be modified. The model’s parameters are learned from rest post templates, blend weights, pose-dependent blend shapes, identity-dependent bled shapes and a regressor from vertices to joint locations.

## **Openpose**

Is a program that can detect human body, hand, facial and foot keypoints to construct a 3D skeletal model. It receives a .mp4 video and maps out 25-keypoints on the body for every frame in the video.



## **PGN Semantic Segmentation**

PGN is a deep learning method for semantic segmentation, instance-aware edge detection and instance-leve human parsing built using tensorflow.

# **Machine Learning Project Flow**

Here I describe the machine learning process for implementation.

## **Data Processing Pipeline**

Before an image is fed to OpenPose and PGN, the data must be formatted correctly to receive them. I wrote a program here that would take an .mp4 video, split the video into images, crop those images to 1080x1080.

## **Setting Up on the Olympus Server**

Configuring the container in which octopus would be maintained ending up being a larger undertaking than previously considered. Within the container, I built a python environment that held the octopus code as well as any dependencies it needed to run.

The largest issue I ran into was when I needed to configure Cmake. Cmake controls the software compilation process and is needed for Dirt, a renderer used.

First I ran into a problem where I couldn’t get cmake to download and tried several fixes including considering replacements for it. I was able to get past this issue by requesting root access to the servers via a sandboxed root that would allow me to download any dependency I require.

I also had to investigate where the linux singularity container nvidia driver files are hidden on olympus as dirt was not pointing at the right location. I had to edit csrc text files to to further assist Dirt in defining the correct gpu-architecture location.

Another large problem, was finding the OpenGL and EGL files needed to make cmake work. This again required editing csrc text files and find the correct location.

Finally, issues were ran into as cuda architecture had to be changed between 10.2, and 11.0 to allow compatibility with Dirt.

# **Validation**

Was able to output a skeletal model using Openpose. Was not able to properly update the dependencies required for Octopus to run on the Olympus server.

# **Subsystem Conclusion**

The subsystem was not completed on plan and did not meet the requirements set in the validation process.

## **Further Development**

### **Integration with other subsystems**

Once the machine learning model has been trained off of hundreds of data inputs, it would be located on the MongoDB database server. From there, the server would communicate with the app to receive video files that it would then send through Octopus and the other preparatory programs. Once done, Octopus would spit out an obj file that would be collected by the server and delivered back to the application.

### **Future Improvements**

The subsystem could be improved by replacing OpenPose and PGN with Facebook’s Detectron 2. Detectron 2 is an object detection platform that uses AI to create both a skeletal model and a semantically segmented image. It improves upon OpenPose and PGN by taking many more keypoints on the body, therefore, producing a smoother model. It also would significantly reduce the runtime needed to create the inputs for the Octopus model.