

---

# **Software Requirements Specification**

**for**

## **Online Trail System Shopping Website**

**Prepared by Muhammad Somaan: 2528404, Madiya Bano: 2542561**

**Middle East Technical University Northern Cyprus Campus**

**Computer Engineering**

**27/04/2023**

# Contents

1 Introduction	3
1.1 Purpose	3
1.2 Scope	3
1.3 Related Work	4
1.4 Product overview	5
1.4.1 Product perspective	5
1.4.2 Product functions	11
1.4.3 User characteristics	12
1.4.4 Limitations	12
1.4.5 Assumptions and dependencies	13
1.5 Definitions	13
2 Specific requirements	14
2.1 External interfaces	14
2.2 Functions	15
2.2.1 UML Use Case Diagram	15
2.2.2 Detailed Use Case Descriptions:	16
2.3 Usability Requirements	30
2.3.1 Performance	30
2.3.2 Productivity	30
2.3.3 Accessibility	30
2.3.4 User Friendliness	30
2.4 Performance requirements	31
2.4.1 Concurrent Users	31
2.4.2 Response Time	31
2.4.3 Data Access	31
2.4.4 Scalability	31
2.5 Logical database requirements	32
2.6 Software system attributes	33
2.6.1 Reliability	33
2.6.2 Availability	33
2.6.3 Security	33
2.6.4 Maintainability	33
2.6.5 Portability	34
2.7 Supporting Information	34
3 References	34
4 Appendices	34
4.1.1 Acronyms and abbreviations	34

# **1 Introduction**

## **1.1 Purpose**

This document illustrates the software requirements for an online shopping website project and provides a comprehensive description of both functional and non-functional requirements for the project. The main objective of the project is to provide potential customers with an interactive 3D clothes trial system which allows them to try their short-listed items before buying. This feature will help them make more informed choice, reduce returns and thus increase efficiency in online shopping.

## **1.2 Scope**

The aim of this project is to develop a user-friendly, interactive 3D clothes trial system which allows the potential customers to virtually try their short-listed items on the 3D dummy created using their body measurements. This feature will provide the customers with a more accurate representation of how clothes will fit them, helping them make more informed choices, reducing the likelihood of returns and increasing customer satisfaction.

The objectives of this project will be to create user a registration system, an interface to register the measurements of the user and develop a database to store the user measurements. These measurements will then be used to design and develop a 3D virtual try-on dummy using the dummy creation system that integrates with the database. Furthermore, a selection system has to be implemented which would allow the users to select the items from product catalogues and try on to dummy. This system would be tested with real users to validate its accuracy and to continuously improve the system based on user feedback and data analysis. This project may have an additional chatbot feature which would help resolve customer grievances in a quicker way based on the feedback previously stored.

This project will help users make more informed choices when shopping online, leading to higher customer retention and increased sales. Furthermore, it reduces the need for physical store visits by making online shopping more interactive and convenient. It will also benefit the sellers by reducing the hassle involved with processing returns and refunds. And ultimately, pave a way for sustainable fashion by making sure the customers buy what they really like.

### **1.3 Related Work**

Online shopping became increasingly popular in recent years, with more and more customers choosing to purchase clothing items online. However, one of the main challenges of online shopping is the inability to physically try on clothing before making a purchase which often leads to sizing issues, that results in a high number of returns and customer dissatisfaction. To address this issue, the use of Virtual Try-On systems (VTOs) that allow users to see how clothes will fit them based on their body measurements has been proposed by researchers.

Several studies have investigated the use of VTOs for online clothing shopping. A study by Başeğmez and Tuncali Yaman, "The Role of Virtual Try-On Technology in Online Purchasing Decision" (2022) found that virtual try-on systems can increase customer satisfaction and reduce the likelihood of returns. The study used 3D virtual models to simulate how clothing items would look on customers based on their body measurements. The results showed that customers who used the virtual try-on system were more satisfied with their purchases and were less likely to return items due to sizing issues.

Another study by Hwangbo et al., "Effects of 3D Virtual 'Try-On' on Online Sales and Customers' Purchasing Experiences" (2020) compared the effectiveness of different types of virtual try-on systems, including 2D, 3D, and augmented reality (AR) systems. The study found that 3D VTOs were the most effective in reducing returns and increasing customer satisfaction. The authors suggested that 3D VTOs provide a more realistic representation of how clothes will fit customers, leading to more informed purchasing decisions.

The literature suggests that VTOs can be an effective solution to the problem of sizing issues in online clothing shopping. By providing customers with a more accurate representation of how clothes will fit based on their body measurements, VTOs can reduce the likelihood of returns and increase customer satisfaction. Zara, one of the pioneers of the fast fashion industry and a highly successful brand has not yet implemented the idea of VTOs on their website. Although our project idea of the implementation of a 3D dummy is in line with existing research, its effectiveness depends on its adaptability, accuracy and usability. Our project aims to increase awareness so that the idea of VTOs may be widely adopted and implemented. Besides, we are committed to continuously improve our systems based on user feedback and data analysis obtained during the development process and after.

## 1.4 Product overview

### 1.4.1 Product perspective

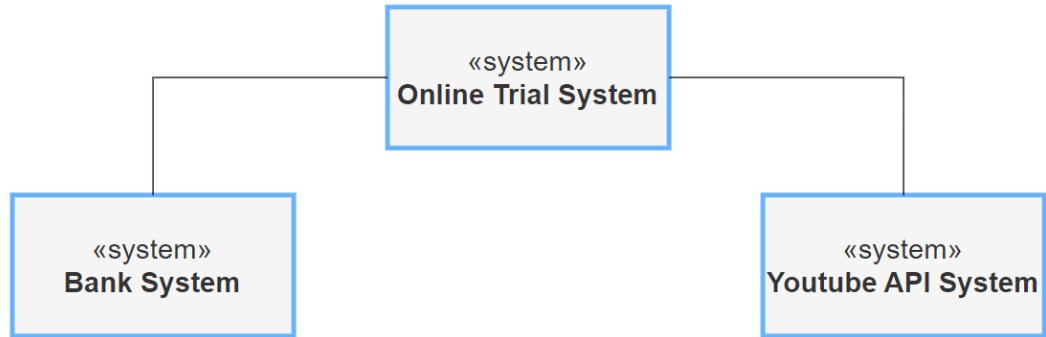


Figure 1: Context Diagram

We have our own system, which interacts with the Bank system for handling payment, and then the Youtube API system to handle the chatbot video playback.

#### 1.4.1.1 System Interfaces

- 1.4.1.1.1 **Youtube API:** We will be interacting with this system to playback a video by providing it a video ID. The video itself will be hosted by the external system itself, and we will only store the ID for it, and playback the video using the id when a visitor uses our chatbot.
- 1.4.1.1.2 **Bank:** We will be interacting with Bank system by providing payment details such as card number and amount to charge, and will expect a response of success or failure, with the transaction ID, from Bank. We will use this at the time when the customer is ready to purchase items.

### **1.4.1.2 User interfaces**

#### **1.4.1.2.1 Log In Page:**

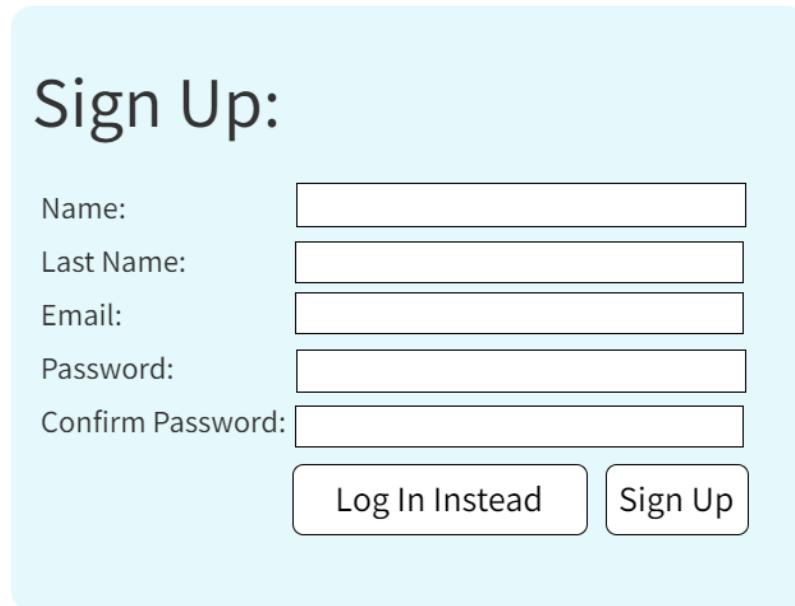
The image shows a light blue rounded rectangular form titled "Sign In:". It contains two input fields: one for "Email" and one for "Password". Below the fields are two buttons: "Sign Up Instead" and "Log In".

Email:	<input type="text"/>
Password:	<input type="password"/>
<input type="button" value="Sign Up Instead"/> <input type="button" value="Log In"/>	

Figure 2: Log In Page

This interface will be used by existing customers to enter their login information and then log in, if a user does not yet have an account, they can click on "sign up instead" button to go to the sign up page.

#### **1.4.1.2.2 Sign Up Page:**



The image shows a light blue rounded rectangular form titled "Sign Up:". It contains five input fields: "Name:", "Last Name:", "Email:", "Password:", and "Confirm Password:", each with a corresponding empty text input box. Below the input fields are two buttons: "Log In Instead" and "Sign Up".

Figure 3: Register Page

This interface will be used by users that don't have an account, to create an account, by providing their name, last name, email and password. They can then click on Sign up button to create their account. If they already have an account, they can click on "Log In Instead" button to goto the log in page.

#### **1.4.1.2.3 Create Dummy:**

### Create Dummy:

Chest Size:

Shoulder Length:

Hip Size:

Waist:

Height:

Age:

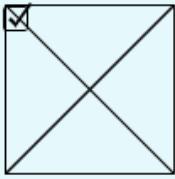
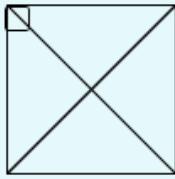
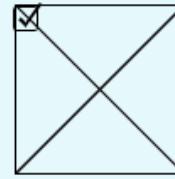
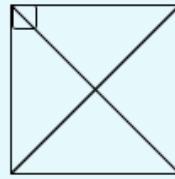
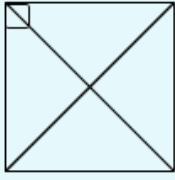
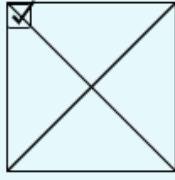
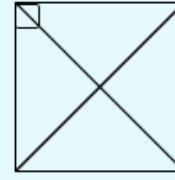
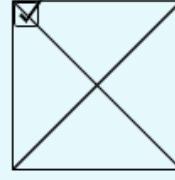
Submit

Figure 4: Dummy Creation Page

This interface is used by existing customers, to create a dummy by providing their measurements such as chest, shoulder, hip, waist size, their height and age. Click on Submit button will start the creation process of the dummy.

#### 1.4.1.2.4 Browse Items:

Browse Items:

			
Boots: 10\$	Shirt : 10\$	Shirt : 7.5\$	Jeans : 15\$
			
Boots: 20\$	Trousers : 10\$	T-Shirt : 12.5\$	Jeans : 25\$

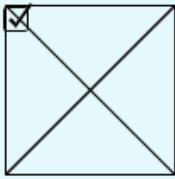
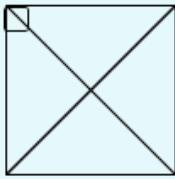
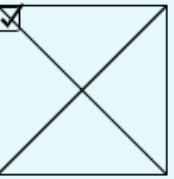
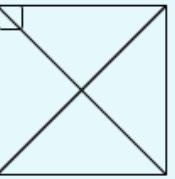
[Remove Items from Basket](#) [Add Items to Basket](#)

Figure 5: Browse Items Page

This interface can be used by any visitor to browse all the available items, and then they can click on the item to select the item, if they are registered user's they can click on "Add Items to Basket" and "Remove Items from Basket" button to add or remove the selected items from their basket.

#### 1.4.1.2.5 Basket Page:

Basket:

			
Boots: 10\$	Shirt : 10\$	Shirt : 7.5\$	Jeans : 15\$
Qty: <input type="button" value="1 ▾"/>	Qty: <input type="button" value="1 ▾"/>	Qty: <input type="button" value="1 ▾"/>	Qty: <input type="button" value="1 ▾"/>
<input type="button" value="Try Item"/>	<input type="button" value="Try Item"/>	<input type="button" value="Try Item"/>	<input type="button" value="Try Item"/>

**Max Total: 42.5\$**      **Selected Total: 42.5\$**

Figure 8: Customer's Basket Page

This interface shows a registered user's basket, which will show all the items in their basket along with their price and quantity and the total price. The user can click on items to select them, and then either click on "proceed with payment" button, where they will sent to payment page. If they click on "View Details" button it will show them the details of their items. The user can also click on "Remove Items from Basket" button to remove selected items from their basket.

#### **1.4.1.2.6 Payment Page:**

The screenshot shows a light blue web page with a white content area. At the top left, it says "Place Order:". To the right, it displays "ID: #2323" and "203.3\$". Below this, there are four input fields: "First Name:" and "Last Name:" each with a text input box, "Card Number:" with a text input box, and "CVV:" with a text input box. Under "Card Number:", there is a dropdown menu labeled "Expiry: MM/YY". At the bottom center is a large, rounded rectangular button with the word "Pay" in the middle.

Figure 7: Payment details Page

This interface will take registered user's card details for processing the payment for their order. It will display the total amount to charge and the their order ID. User can click on "Pay" button to complete their purchase.

#### **1.4.2 Product functions**

This software is for a shopping website with a trial system. The website will contain the list of items available for sale. The website has a basket system for adding and removing items unique to each registered user. The website only allows item purchases to registered users. Furthermore the website has a chatbot, which can present videos regarding the website's operational cycle and on how to create a dummy. Creating a dummy is the main feature of the website, it allows registered users to create a 3d model of their body using their measurements, and then allows them to try any item on their dummy from their basket. The website can then prompt the user if the size matches or not. This trial system allows users to be sure that they won't receive an item which does not fit them, if such case occurs they can submit an issue on the website, and an admin will look into the issue to resolve it. Only admins are allowed to add or modify items that are available for sale on website.

### **1.4.3 User characteristics**

The intended group of users is of those ages above 18+. A user need to have a credit/debit card to order, a working email address and have enough education level to be able to read and understand basic English language. It is difficult for those users who cannot see to use our website, but by using a text narrator on whatever device they are using, it is possible, but the product itself won't have a text narrator feature. A user must have the knowledge of how to navigate through a website on the device of their liking, and have knowledge on how to make payments online using their bank card. Although the trial system in the product is optional, meaning an item can be ordered without ever creating a dummy, but for dummy creation, the user must be knowledgeable enough to know their body parts and how to measure their size.

### **1.4.4 Limitations**

- 1.4.4.1 Bank Card:** The availability of a bank card is a must to fully utilize the product, otherwise the visitors can only browse or try items, and not be able to order them.
- 1.4.4.2 Email-Address:** The availability of an email-address is a must to fully utilize the product, otherwise visitors can only browse items.
- 1.4.4.3 User count limit:** Since 3d modeling of a dummy takes extensive computation, only a set number of users may use the trial and creation of dummy feature at one time.
- 1.4.4.4 External Systems:** The downtime of external systems such as Bank, can prevent user's from completing their order, similarly in case of downtime in Youtube API, the user's won't be able to watch the videos explaining the process of website.
- 1.4.4.5 Product Hosting:** The product must be hosted on a good enough computer/server that can handle the 3d modelling computations for several users at a time.
- 1.4.4.6 Storage:** Database must be used to store the 3d models and the items data, storing data on files is not an option, due to higher access time.
- 1.4.4.7 3d Model Viewer:** A 3d Model Viewer must be present in the user's browser to view the dummy model, otherwise the user won't be able to view the model.

## **1.4.5 Assumptions and dependencies**

- 1.4.5.1 Hardware:** Assuming that the user will use a device released in the previous 3 years, which supports an internet connection and has a browser.
- 1.4.5.2 Browser:** Assuming that the user will have the latest version of the top 3 current browsers.
- 1.4.5.3 Internet Connection:** Assuming that the user will have a stable internet connection to use the product.
- 1.4.5.4 Database:** A database setup is required for the software to operate.
- 1.4.5.5 Bank System:** The software is dependent on the Bank system so assuming that the bank system will be able to handle all the requests sent by the product, and send back a result in a reasonable time.
- 1.4.5.6 Youtube API:** The software is dependent on Youtube API, so assuming that youtube API is able to handle the requests made, and be able to properly show the videos.
- 1.4.5.7 DNS Setup:** Assuming that the website's routing is properly configured, such that it is not blocked in any specific country/region.
- 1.4.5.8 DDOS Protection:** The software is dependant on a ddos protection technique, to prevent any DDOS attack that may render the software unusable.

## **1.5 Definitions**

N/A

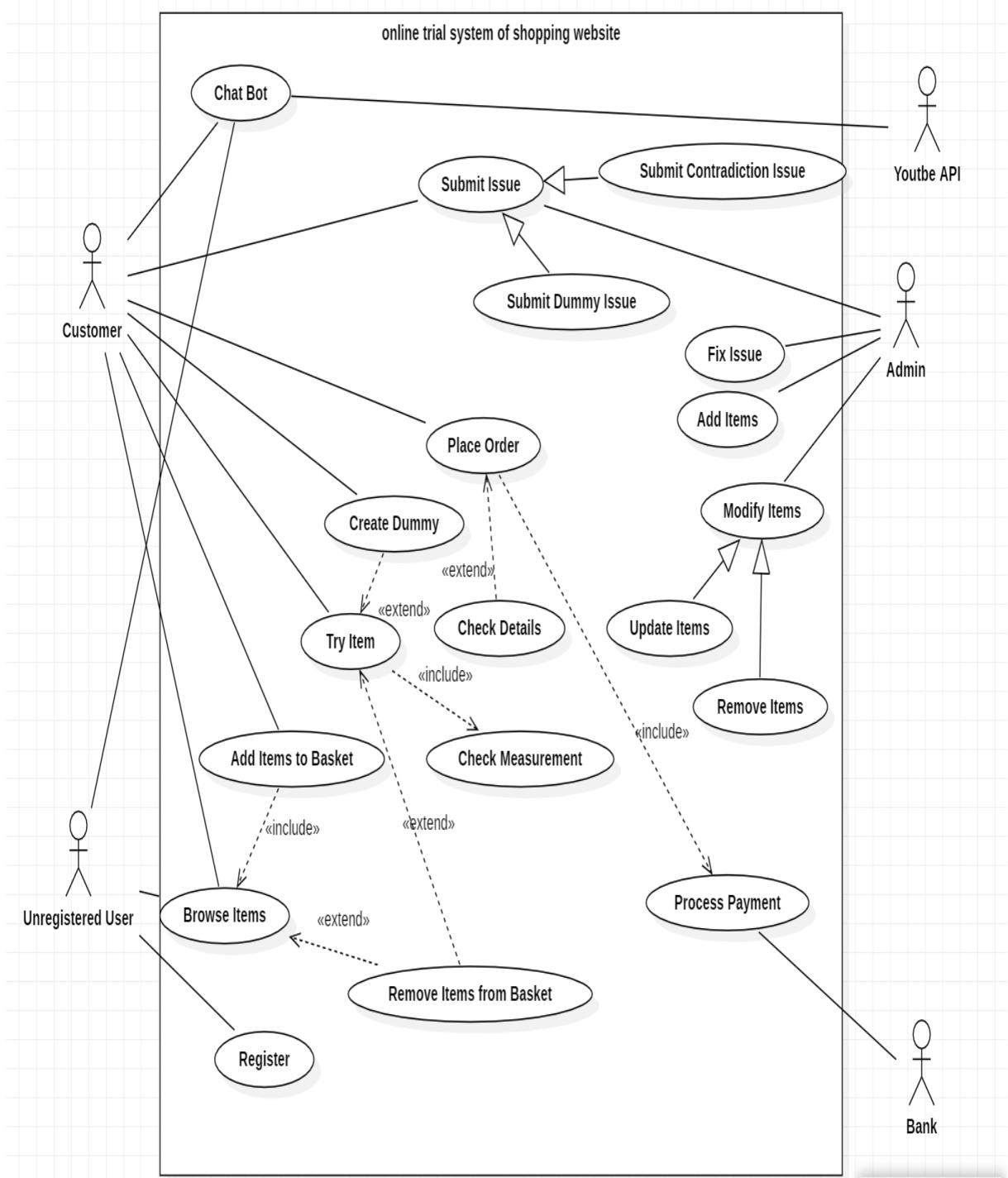
## 2 Specific requirements

### 2.1 External interfaces

System Interface	Functionality	Input	Output
Youtube API	It will be given a video ID, and it will fetch the video with that ID, so that the website can play the video.	1. Video_ID: String	Video_Playback: Byte_Stream In case the video is not found, the byte stream will be null.
Bank	It will be given bank card details, amount to charge, and it will process the payment, and will present the transaction id, or error codes, in case of failure.	1.First_Name: String 2.Last_Name: String 3.Card_Num: int 4.Card_CVV: int 5.Card_Exp: int 6. Amount: float	Transaction_ID: int  If transaction id is a 9 digit number, the transaction is successful and the 9 digit number is the id. else: if its 0, insufficient balance. if its 1, invalid First or last name. if its 2, invalid card number. if its 3, invalid expiry. if its 4, invalid cvv if its 5, other reason for failure.

## 2.2 Functions

### 2.2.1 UML Use Case Diagram



## 2.2.2 Detailed Use Case Descriptions:

### 2.2.2.1 Register

<b>Use Case</b>	Register
<b>Actors</b>	Unregistered User
<b>Cross references</b>	
<b>Typical Course of Events</b>	
<b>Actor Intentions</b>	<b>System Responsibility</b>
1. User clicks on register button	
	2. System opens register page/form
3. User enters their details such as, Name, Lastname, email address and password	
4. User clicks on submit button	
	5. System validates the information provided, email in correct format, password matching the criteria.
	6. System verifies no previous account exist on the email provided.
	7. System adds an entry for the account in the database.
	8. System shows confirmation message.
9. User clicks on "return to home page" link	
<b>Alternative Courses</b>	
Step 5: If email is not in correct format, or password does not match criteria, User will be presented to enter the details again, Goes to Step 2. Step 6: If an account already exists on the email provided, System shows a message stating such, and cancels the operation.	

### 2.2.2.2 Create Dummy

<b>Use Case</b>	Create Dummy
<b>Actors</b>	Customer
<b>Cross references</b>	
<b>Typical Course of Events</b>	
<b>Actor Intentions</b>	<b>System Responsibility</b>
1. User Clicks on add dummy button	
	2. System opens a form for adding Measurements
3. User enters their measurements, such as chest size, shoulder length waist, hip size, height and age.	
4. User clicks on submit button	
	5. System validates the measurements, and verifies they are not something that is not possible, such as o.
	6. System creates a 3D model with the measurements provided
	7. Systems displays the 3D model.
8. User is shown the dummy, and provided buttons to Save or Create new Model.	
9. User clicks on Save Dummy	
	10. System adds the measurements to the database.
	11. System shows a confirmation message, and goes back to Step 7.
<b>Alternative Courses</b>	
Step 5: If the measurements are not reasonable, System asks user to enter the measurements again, goes to Step 2.	
Step 8: User can click on Create new Model, in which case System goes to Step 2.	

### 2.2.2.3 Browse Items

<b>Use Case</b>	Browse Items
<b>Actors</b>	Unregistered User, Customer
<b>Cross references</b>	Add Items to Basket
<b>Typical Course of Events</b>	
<b>Actor Intentions</b>	<b>System Responsibility</b>
1. User clicks on Browse Items button on Homepage	
	2. System fetches all available items from the db.
	3. System displays all the available items, and "add to basket" and "remove items from basket" button at the bottom.
4. Customer selects 1 or more items.	
5. Customer clicks on "add to basket" button.	
	6. System creates a new entry in customer's basket in the db.
	7. System adds the selected items to Customer's basket
	8. System displays confirmation message.
	9. System displays all items again, goes to Step 2.
10. User clicks on "Return to homepage" button	
<b>Alternative Courses</b>	
Step 4: Customer clicks on "remove items from basket" button instead, so System goes to Remove Items from Basket use case.	

#### 2.2.2.4 Remove Items from Basket

<b>Use Case</b>	Remove Items from Basket
<b>Actors</b>	Customer
<b>Cross references</b>	Browse Items, Try Item
<b>Typical Course of Events</b>	
<b>Actor Intentions</b>	<b>System Responsibility</b>
1. Customer selects the item to remove.	
2. Customer clicks on remove items from basket button.	
	3. System fetches all items in Customer's basket from the db.
	4. System verifies if the selected item exist in the user's basket.
	5. System removes the item from the basket and updates it in the db.
	6. System shows a confirmation message.
7. Customer clicks on "go to homepage" button.	
<b>Alternative Courses</b>	
Step 4: If the selected item does not exist in Customer's basket, system shows an error to the customer explaining such.	

### 2.2.2.5 Try Items

<b>Use Case</b>	Try Items
<b>Actors</b>	Customer
<b>Cross references</b>	Remove Items from Basket
<b>Typical Course of Events</b>	
<b>Actor Intentions</b>	<b>System Responsibility</b>
1. Customer opens their basket	
2. Customer selects an item to try	
	3. System fetches the item details from the db.
	4. System creates a 3D model of the selected item.
	5. System presents Customer with list of all available dummies.
6. Customer selects a dummy to try the item on.	
	7. System gets the selected dummy model from db.
	8. System merges the dummy model and the item model together, and shows the user a 3D model of dummy with the item.
	9. System compares the measurements of the item and the dummy.
	10. If measurements don't match system will prompt customer stating such.
11. If customer does not like the items, user can remove the items from basket	
<b>Alternative Courses</b>	
Step 6: If no dummy exist for customer, then system gives an option to create a dummy. Step 11: Customer may not remove any items from basket.	

### 2.2.2.6 Place Order

<b>Use Case</b>	Place Order
<b>Actors</b>	Customer, Bank
<b>Cross references</b>	Check details
<b>Typical Course of Events</b>	
<b>Actor Intentions</b>	<b>System Responsibility</b>
1. Customer opens their basket.	
	2. System calculates and displays the total price.
	3. System adds "proceed with payment" button, to allow payment processing.
4. Customer clicks on "proceed with payment" button	
	5. System opens a form to receive customer's credit/debit card details.
6. Customer enters their card details, such as card number, expiry and CVV.	
7. Customer clicks on "Pay" button	
	8. System sends the card details and price to Bank system.
9. Bank performs the transaction and sends confirmation message.	
	10. System shows a confirmation message with the items details.
11. Customer clicks on "return to homepage" button.	
<b>Alternative Courses</b>	
Step 9: If the bank transaction is not successful, then the bank will send an error message related to the problem, and the system shows the error message and goes back to Step 5.	

### 2.2.2.7 Check Details

<b>Use Case</b>	Check Details
<b>Actors</b>	Customer
<b>Cross references</b>	
<b>Typical Course of Events</b>	
<b>Actor Intentions</b>	<b>System Responsibility</b>
1. Customer clicks on "view order details" button.	
	2. System fetches customer's order from the db with order id.
	3. System receives the items in the order user added the items.
	4. System calculates total price including VAT
	5. System displays all the items and details in a specific format
	6. System adds a "print details" button at the bottom.
7. Customer clicks on "print details" button	
	8. System prepares a document with order details in a printable format.
<b>Alternative Courses</b>	
Step 8: Customer may not click on "print details" button, and hence System will keep showing the details.	

### 2.2.2.8 Chat Bot

<b>Use Case</b>	Chat Bot
<b>Actors</b>	Unregistered User, Customer, Youtube API
<b>Cross references</b>	
<b>Typical Course of Events</b>	
<b>Actor Intentions</b>	<b>System Responsibility</b>
1. User clicks on chatbot link.	
	2. System gives the option to select which tutorial to watch, either for creating a dummy, or outline of website's operational cycle.
3. User selects the tutorial they want to watch.	
	4. System fetches selected tutorial's video ID from the db.
	5. System sends the video ID to Youtube API, to receive the video.
6. Youtube API receives the video ID, and provides the link.	
	7. System opens an embedded video of the video link provided by Youtube API.
8. User watches the video.	
9. User clicks on "close video button"	
	10. System gives the option to select which tutorial to watch again, or to return to homepage.
11. User clicks on "return to homepage" button.	
<b>Alternative Courses</b>	
Step 11: Instead of clicking selecting on "return to homepage" button, user selects another tutorial, and hence System goes to Step 4.	

### 2.2.2.9 Submit Dummy Issue

<b>Use Case</b>	Submit Dummy Issue
<b>Actors</b>	Customer, Admin
<b>Cross references</b>	
<b>Typical Course of Events</b>	
<b>Actor Intentions</b>	<b>System Responsibility</b>
1. Customer clicks on "Help" button.	
	2. System opens a selection for user to select their issue. Either with creating a dummy, or contradiction in item's measurements.
3. Customer selects issue with creating a dummy.	
	4. System opens a form to enter details of the issue
5. Customer enters the details of the issue.	
	6. System takes the selection, the details of issue and the details of customer and adds it to the table of dummy issues in the db with a new issue ID.
	7. System shows a confirmation message and the issue ID.
	8. System sends message to Admin, to notify him.
9. Customer clicks on "return to homepage" button.	
<b>Alternative Courses</b>	
.	

### 2.2.2.10 Submit Contradiction Issue

Use Case	Submit Contradiction Issue
Actors	Customer
Cross references	
Typical Course of Events	
Actor Intentions	System Responsibility
1. Customer clicks on "Help" button.	
	2. System opens a selection for user to select their issue. Either with creating a dummy, or contradiction in item's measurements.
3. Customer selects issue with contradiction in item's measurements	
	4. System opens a form to take order ID, item Name, and the details
5. Customer enters the details of the issue, with the item name, order id.	
	6. System validates the entries, if the order id and item name are correct.
	10. System adds the details of issue, including order id, and item name and the details of customer and adds it to the item contradiction issues table in db with a new issue ID.
	11. System shows a confirmation message and the issue ID.
	12. System sends message to Admin, to notify him.
13. Customer clicks on "return to homepage" button.	
Alternative Courses	
.Step 6: System, is unable to validate the entries, either wrong order id or item name is provided, goes back to Step 5.	

### **2.2.2.11 Fix Issue**

<b>Use Case</b>	Fix Issue
<b>Actors</b>	Admin
<b>Cross references</b>	
<b>Typical Course of Events</b>	
<b>Actor Intentions</b>	<b>System Responsibility</b>
1. Admin clicks on "Issues List" button	
	2. System fetches all the unresolved issues from the db.
	3. System displays all the issues, and asks for selection of an issue.
4. Admin selects an issue.	
	5. System fetches the details of the selected issue from the db.
	6. System displays the details of the selected issue.
7. Admin works on the issue, and marks it as fixed.	
	8. System marks the selected issue as fixed in the db.
9. Admin clicks on "return to homepage button".	
<b>Alternative Courses</b>	
.Step 7: Admin is not able to fix the issue for now, and marks the issue with unresolved, then goes to Step 9.	

### 2.2.2.12 Add Items

<b>Use Case</b>	Add Items
<b>Actors</b>	Admin
<b>Cross references</b>	
<b>Typical Course of Events</b>	
<b>Actor Intentions</b>	<b>System Responsibility</b>
1. Admin clicks on "Add Item" button.	
	2. System opens a form to enter the item details such as name, price, picture, and measurements.
3. Admin enters the details of the item.	
	4. System takes the details and adds it to the db.
	5. System shows a confirmation message.
6. Admin clicks on "return to homepage" button.	
<b>Alternative Courses</b>	
Step 6: Admin clicks on "add more items" button instead, goes to Step 2.	

### 2.2.2.13 Update Items

<b>Use Case</b>	Update Item
<b>Actors</b>	Admin
<b>Cross references</b>	
<b>Typical Course of Events</b>	
<b>Actor Intentions</b>	<b>System Responsibility</b>
1. Admin clicks on "Update Item" button.	
	2. System fetches all available items from db.
	3. System displays all the items, and allows selection of an item.
4. Admin selects an item.	
	5. System opens a form to change the item details such as name, price, picture, and measurements.
6. Admin enters the new details of the item.	
	7. System takes the details and updates it to the db according to the item id.
	8. System shows a confirmation message.
9. Admin clicks on "return to homepage" button.	
<b>Alternative Courses</b>	
Step 9: Admin clicks on "update more items" button instead, goes to Step 2.	

#### **2.2.2.14 Remove Item**

<b>Use Case</b>	Remove Item
<b>Actors</b>	Admin
<b>Cross references</b>	
<b>Typical Course of Events</b>	
<b>Actor Intentions</b>	<b>System Responsibility</b>
1. Admin clicks on "Remove Item" button.	
	2. System fetches all available items from db.
	3. System displays all the items, and allows selection of 1 or more items.
4. Admin selects 1 or more items.	
	5. System deletes all the selected items from db according to their item id.
	6. System shows a confirmation message.
7. Admin clicks on "return to homepage" button.	
<b>Alternative Courses</b>	

## **2.3 Usability Requirements**

### **2.3.1 Performance**

We need to keep track of the performance metrics for a task to ensure that users are able to complete that task successfully without errors, if there are errors we need to think about a better approach to implement the task, otherwise users will not be able to use our product. For this, we can use the Error Rate metric.

### **2.3.2 Productivity**

We need to keep track of the productivity metrics for a task to make sure that the task is completed within an acceptable range of time, otherwise users may get frustrated and impatient, and stop using our product. For this, we can use the Task Time metric.

### **2.3.3 Accessibility**

We need to keep track of accessibility metrics to ensure that all users regardless of their technical knowledge be easily able to complete a task easily, otherwise we will be limiting certain groups of users from accessing our product. For this, we can use Single Ease Question (SEQ).

### **2.3.4 User Friendliness**

We need to keep track of user-friendliness or satisfaction, to ensure that users enjoy or are satisfied with the overall layout and the flow of the tasks in our product, otherwise users will prefer not to use our product. For this, we can use System Usability Scale (SUS).

## **2.4 Performance requirements**

### **2.4.1 Concurrent Users**

Our product should be able to handle atleast 100 users at the same time, since we expect our product to atleast be used by 100 users at the same time. If this is not maintained users will experience downtime, and very high response time, which will make them irritated and they will stop using our product.

### **2.4.2 Response Time**

Our product's response time for every action should be less than 1 second, the only expection is the dummy creation and the try item task, which can take max 10s, this is done so to ensure that users experience a responsive website, and not a laggy product, which can often frustate users, and make them stop using our product.

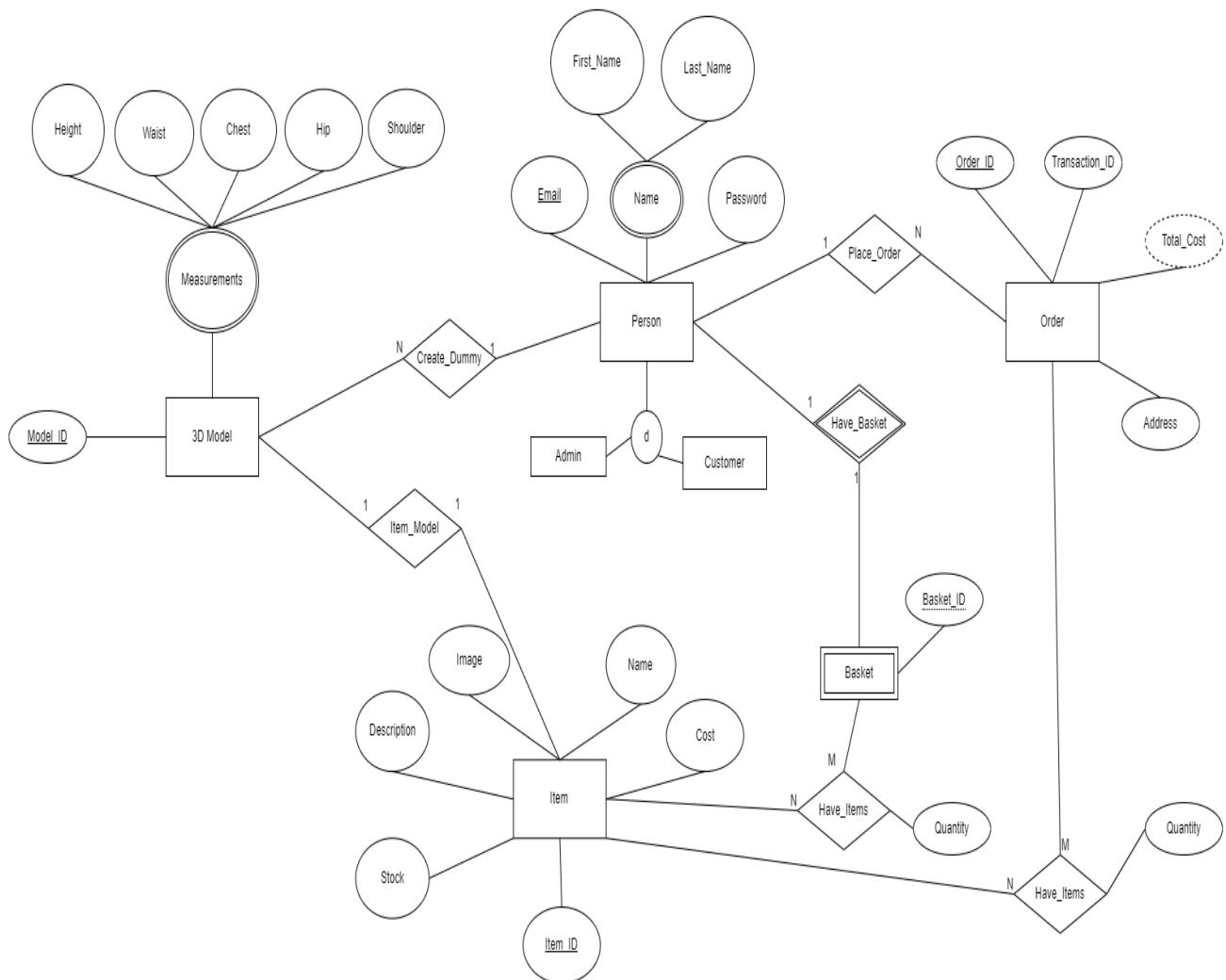
### **2.4.3 Data Access**

Our product's data access time should be less than 1 second, this means that we should implement a database that can match this requirement, similarly we should ensure that the data stored in the database is also stored in such a way to minimize data access time. This is important since if the loading of data takes huge time, users won't be able to correctly use our product, an example is of while browsing items, the items should load from database quickly so the user can browse all the items, if items take too much time to load, users won't be able to browse all the items, and may leave our product thinking that the item they need is not present.

### **2.4.4 Scalability**

Our product should be easily scalable to handle an increase in concurrent users or data storage needs. We anticipate an increase in user traffic and product usage over time by 10% every month, so we need to ensure that our product can handle this growth without experiencing any significant downtime or degradation in performance. This can be achieved by implementing a scalable architecture and using cloud-based infrastructure, which can dynamically adjust to changing demands.

## 2.5 Logical database requirements



## **2.6 Software system attributes**

### **2.6.1 Reliability**

- 2.6.1.1 The product should be reliable in sense that, whatever task the user wants to do is actually done, and there should be no unexpected results. There should be no errors or unexpected results. We need this because unexpected results can lead to loss of progress or overall frustration of user which will lead the user to quit using the product.
- 2.6.1.2 The product should be reliable, that there are no memory leaks, as memory leaks can cause the software to crash, which will lead to higher downtime or even loss of data, such as transactions, which will cause issues such as missing orders.
- 2.6.1.3 The product should be able to store huge number of data in the database, since the more and more users are expected to register on the software, if we cannot handle huge data, this may result in higher access time, or even corruption of data while accessing, this will cause incorrect orders and operations to start occurring in the product.

### **2.6.2 Availability**

- 2.6.2.1 The product should be available for most time, meaning that the uptime should be atleast 99%, so that every user can use the product without any interruption. If the product is experiencing frequent downtime, user's will prefer not to use the product.
- 2.6.2.2 The product should restart in under a minute in-case of system failure, or restart. If the product takes too long to restart, or doesn't restart without human intervention, then this will lead to missed sales, or even loss of transactions. User's may face order issues, which will lead to overall bad image for the product.

### **2.6.3 Security**

- 2.6.3.1 The product should securely send the card details to bank system by encrypting the data, so that no hacker can retrieve the information, as leaking of card details will lead to legal issues for the product, and product owners, which might lead to shutdown of product.
- 2.6.3.2 The product should save the passwords of registered users in a secure way, such as by using hashing technique, such as SHA-256. This is done so in case of data leak, the hacker isn't able to identify the password of user's, otherwise the product owners can face lawsuits, which will destroy the image of the product, drive off customers and may even lead to shutdown.

## **2.6.4 Maintainability**

- 2.6.4.1 The product should store data in database in simplest form possible, by applying normalization, so in future it is easier to add more features, and also to go through data easily if there are any issues.
- 2.6.4.2 The product should log all the actions, this can help when finding an error, and will make it much easier and less time consuming to find faults.

## **2.6.5 Portability**

- 2.6.5.1 The product should not depend on the hardware or operating system, such as being able to run on both x64 and x86 hardware, similarly it should also be able to run on linux or Windows server. By doing this we can ensure that when the time for upgrade comes we don't face any issue shifting to the newer hardware. As problems in shifting can result in lost data, or high downtime.
- 2.6.5.2 The product should be written in a language that is supported by all operating systems, such as Java, similarly the database service used should also be compatible on all operating systems, this can be taken care of by choosing popular database services. If we don't adhere to these we might lose data in case we need to service our current hardware and have to run the software on another hardware.

## **2.7 Supporting Information**

N/A

## **3 References**

- 3.1 Başeğmez, & Tuncali Yaman. (2022, March). The Role of Virtual Try-On Technology in Online Purchasing Decision. *Journal of Research in Business*, 7(IMISC 2021 Special Issue), 165–176. <https://doi.org/10.54452/jrb.1023619>
- 3.2 Hwangbo, H., Kim, E. H., Lee, S. H., & Jang, Y. J. (2020, September). Effects of 3D Virtual “Try-On” on Online Sales and Customers’ Purchasing Experiences. *IEEE Access*, 8. <https://doi.org/10.1109/access.2020.3023040>

## 4 Appendices

### 4.1.1 Acronyms and abbreviations

4.1.1.1 API: Application Programming Interface

4.1.1.2 VTO: Virtual Try On

Grading Item	Standard
Introduction – Purpose	4
Introduction – Scope	4
Introduction – Related Work	4
Introduction – Product Overview – Context Model and System Interfaces	8
Introduction – Product Overview – User Interfaces	8
Introduction – Product Overview – Product Functions	8
Introduction – Product Overview – User Characteristics	2
Introduction - Product Overview – Limitations	2
Introduction - Product Overview – Assumptions and Dependencies	4
Introduction – Definitions	*
Specific Requirements – External Interfaces	4
Specific Requirements – Functions – UML Use-case Diagram	16
Specific Requirements – Functions – Use case descriptions	16
Specific Requirements – Usability Requirements	4
Specific Requirements – Performance Requirements	4
Specific Requirements – Logical Database Requirements	8
Specific Requirements – Software System Attributes	4
Specific Requirements – Supporting Information	*
References	*
Appendices - Acronyms and abbreviations	*

\* Marks will be deducted from other related sections if they are not provided where necessary.