****

**COMSATS University Islamabad (CUI)**

Software Requirement Specification  
(SRS DOCUMENT)

for

**VARDROBE**  
Version 2.0

***By***

**Ali Tahir CIIT/FA17-BSE-017/ISB**

**Ashhad Ahsan Rehman CIIT/ FA17-BSE-027/ISB**

***Supervisor*Dr.Syed Sohaib Ali**

*Bachelor of Science in Software Engineering (2017-2021)*

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| Madam Sadia Maqbool | 29-10-2020 | The use case diagram is very long, and the normal flows and alternative flows should use better wordings and should be written in the form of points. | 1.0 |
| Madam Sadia Maqbool | 29-10-2020 | Add Functional requirements related to the AR and Virtual try on the module. | 1.0 |
| Dr. Uzair Iqbal Janjua | 29-10-2020 | Add only realistic nonfunctional requirements within the stipulated time frame. | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Application Evaluation History

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
| Divide one large use case into small and manageable use cases. | The large use cases are broken down into 5 smaller use cases and the image is much clearer now. |
| AR and Virtual try module FR should be mentioned in detail. | The steps needed to perform in doing AR and Virtual try on are included. |
| Reduce the non-functional requirements to meet the scope of application. | Only achievable nonfunctional requirements are kept. |
|  |  |
|  |  |

Supervised by

**Dr. Syed Sohaib Ali**

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Table of Contents**

**Contents**

[1. Introduction 5](#_Toc58174668)

[1.1 Purpose 5](#_Toc58174669)

[1.2 Scope 5](#_Toc58174670)

[2. Overall Description 5](#_Toc58174671)

[2.1 Product Perspective 5](#_Toc58174672)

[2.2 Operating Environment 5](#_Toc58174673)

[2.3 Design and Implementation Constraints 5](#_Toc58174674)

[3. Requirement Identifying Technique 6](#_Toc58174675)

[3.1 Use Case Diagram 6](#_Toc58174676)

[3.2 Use Case Description 10](#_Toc58174677)

[4. Functional Requirements 21](#_Toc58174678)

[4.1 Functional Requirement – I 21](#_Toc58174679)

[4.2 Functional Requirement – II 21](#_Toc58174680)

[4.3 Functional Requirement-III 22](#_Toc58174681)

[4.4 Functional Requirement – IV 22](#_Toc58174682)

[4.5 Functional Requirement – V 23](#_Toc58174683)

[4.6 Functional Requirement – VI 23](#_Toc58174684)

[4.7 Functional Requirement – VII 24](#_Toc58174685)

[4.8 Functional Requirement – VIII 24](#_Toc58174686)

[4.9 Functional Requirement – IX 25](#_Toc58174687)

[4.10 Functional Requirement – X 25](#_Toc58174688)

[4.11 Functional Requirement – XI 26](#_Toc58174689)

[4.12 Functional Requirement – XII 26](#_Toc58174690)

[4.13 Functional Requirement – XIII 26](#_Toc58174691)

[4.14 Functional Requirement – XIV 27](#_Toc58174692)

[4.15 Functional Requirement – XV 27](#_Toc58174693)

[4.16 Functional Requirement – XVI 28](#_Toc58174694)

[4.17 Functional Requirement – XVII 28](#_Toc58174695)

[4.18 Functional Requirement – XVIII 29](#_Toc58174696)

[4.19 Functional Requirement – XIX 29](#_Toc58174697)

[4.20 Functional Requirement – XX 29](#_Toc58174698)

[4.21 Functional Requirement – XXI 30](#_Toc58174699)

[4.22 Functional Requirement – XXII 30](#_Toc58174700)

[4.23 Functional Requirement – XXIII 31](#_Toc58174701)

[4.24 Functional Requirement – XXIV 31](#_Toc58174702)

[4.25 Functional Requirement – XXV 31](#_Toc58174703)

[4.26 Functional Requirement – XXVI 32](#_Toc58174704)

[4.27 Functional Requirement – XXVII 32](#_Toc58174705)

[4.28 Functional Requirement – XXVIII 33](#_Toc58174706)

[4.29 Functional Requirement – XXIX 33](#_Toc58174707)

[4.30 Functional Requirement – XXX 34](#_Toc58174708)

[4.31 Functional Requirement – XXXI 34](#_Toc58174709)

[4.32 Functional Requirement – XXXII 34](#_Toc58174710)

[4.33 Functional Requirement – XXXIII 35](#_Toc58174711)

[4.34 Functional Requirement – XXXIV 35](#_Toc58174712)

[4.35 Functional Requirement – XXXV 36](#_Toc58174713)

[4.36 Functional Requirement – XXXVI 36](#_Toc58174714)

[4.37 Functional Requirement – XXXVII 37](#_Toc58174715)

[4.38 Functional Requirement – XXXVIII 37](#_Toc58174716)

[4.39 Functional Requirement – XXXIX 38](#_Toc58174717)

[4.40 Functional Requirement – XL 38](#_Toc58174718)

[4.41 Functional Requirement – XLI 39](#_Toc58174719)

[4.42 Functional Requirement – XLII 39](#_Toc58174720)

[4.43 Functional Requirement – XLIII 40](#_Toc58174721)

[4.44 Functional Requirement – XLIV 40](#_Toc58174722)

[4.45 Functional Requirement – XLV 40](#_Toc58174723)

[4.46 Functional Requirement – XLVI 41](#_Toc58174724)

[4.47 Functional Requirement – XLVII 41](#_Toc58174725)

[4.48 Functional Requirement – XLVIII 42](#_Toc58174726)

[4.49 Functional Requirement – XLIX 42](#_Toc58174727)

[5. Non-Functional Requirements 43](#_Toc58174728)

[5.1 Usability 43](#_Toc58174729)

[5.2 Performance 43](#_Toc58174730)

[5.3 Security 43](#_Toc58174731)

[5.4 Scalability 43](#_Toc58174732)

[5.5 Integrity 44](#_Toc58174733)

[*5.6* Maintainability 44](#_Toc58174734)

[5.7 Installibility 44](#_Toc58174735)

[6. External Interface Requirements 44](#_Toc58174736)

[6.1 User Interfaces 44](#_Toc58174737)

[6.2 Software Interfaces 45](#_Toc58174738)

[6.3 Hardware Interfaces 45](#_Toc58174739)

[6.4 Communication Interfaces 45](#_Toc58174740)

[7. References 46](#_Toc58174741)

[8. Plagiarism Report 47](#_Toc58174742)

# Introduction

## Purpose

The purpose of this document is to provide an understanding of VARDROBE which stands for Virtual Augmented Reality Droid for Better E-Commerce which aims at providing a better e-commerce solution to the masses.

## Scope

The idea of VARDROBE is inspired by virtual reality and virtually try on’s. This system will aim to provide an e-commerce platform to the users where they will be able to try out the products which they wish to buy before those products even reach their doorstep through the concept of virtually trying on their product through their devices, furthermore this system also aims to implement augmented reality to provide a realtime view of the product through the android device of the customer.

# Overall Description

## Product Perspective

This system is related to already existing e-commerce platforms, although this specific platform of ours aims at a higher customer satisfaction rate by improving the idea of e-commerce through the introduction of Virtual Try on’s now achievable by the use of machine learning. VARDROBE aims to improve e-commerce by the implementation of Virtual Try on and Augmented Reality.

## Operating Environment

The System will have support for Android Version 6.0(Marshmallow) up to Android Version 10 and maybe Android 11 as it is new and there is less information about it right now, furthermore VARDROBE will also support iOS Version 11,12 and 13. The system will be intended for mobile devices and iPads and android tablet devices also.

## Design and Implementation Constraints

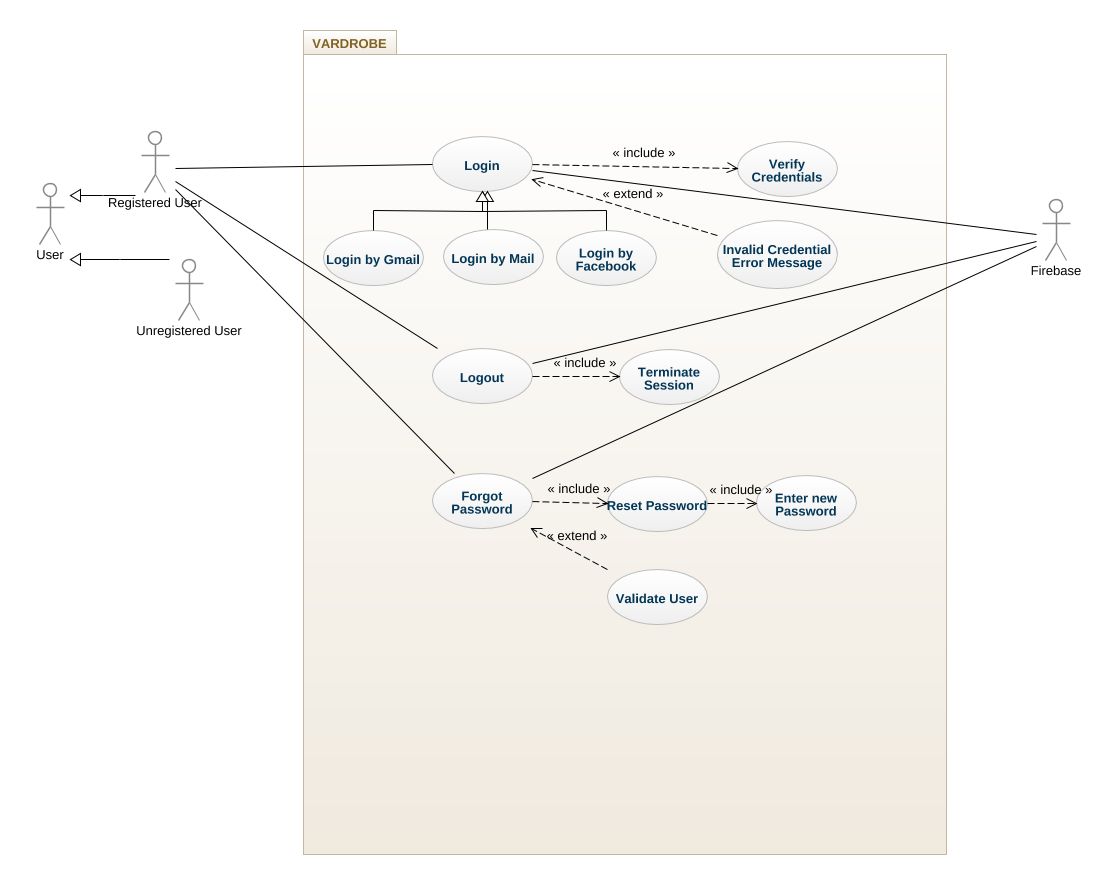
* The System will be developed using flutter which is an open-source user interface software development kit.
* The System will be built using dart language along with flutter.
* The System will use firebase as a backend database and also as a cloud hosting service.
* The System will also have machine learning models implemented using the python language and TensorFlow or PyTorch libraries.
* The System will be developed using Android Studio as a developing environment.

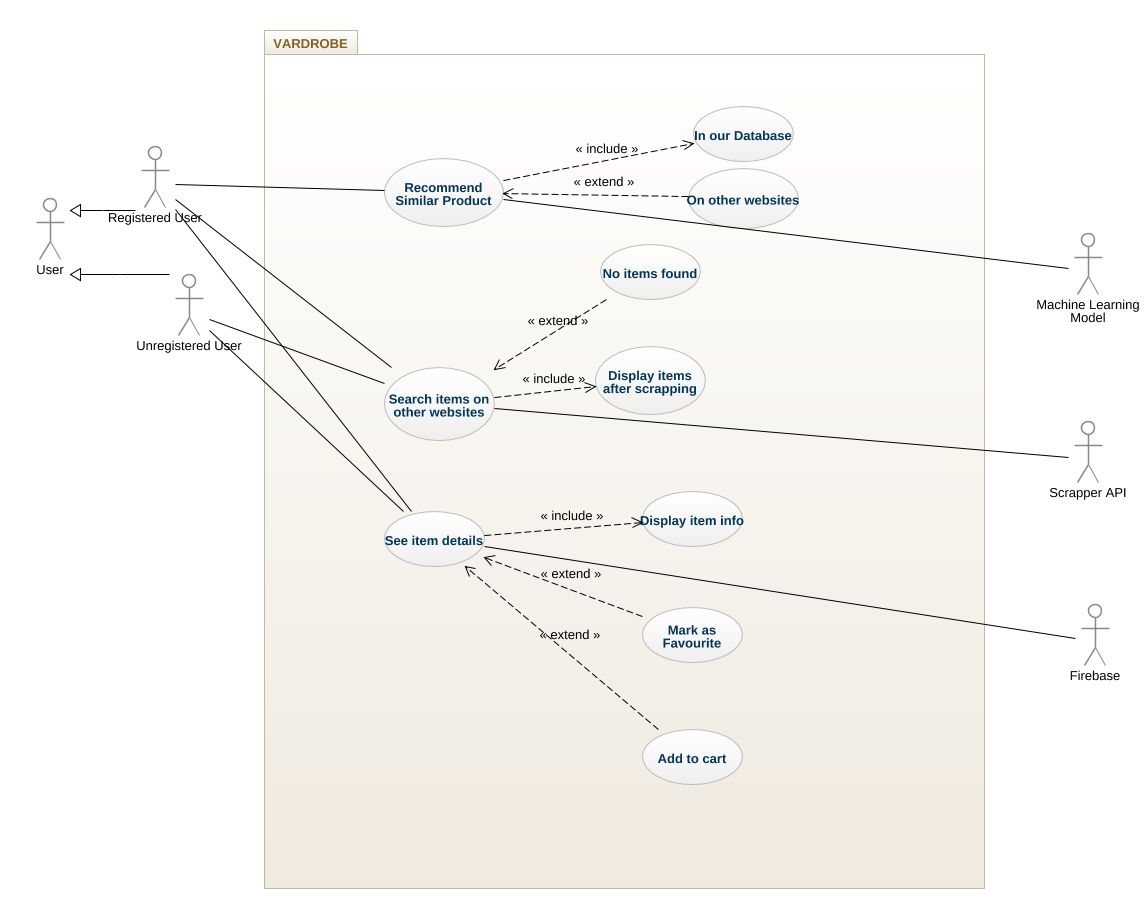
# Requirement Identifying Technique

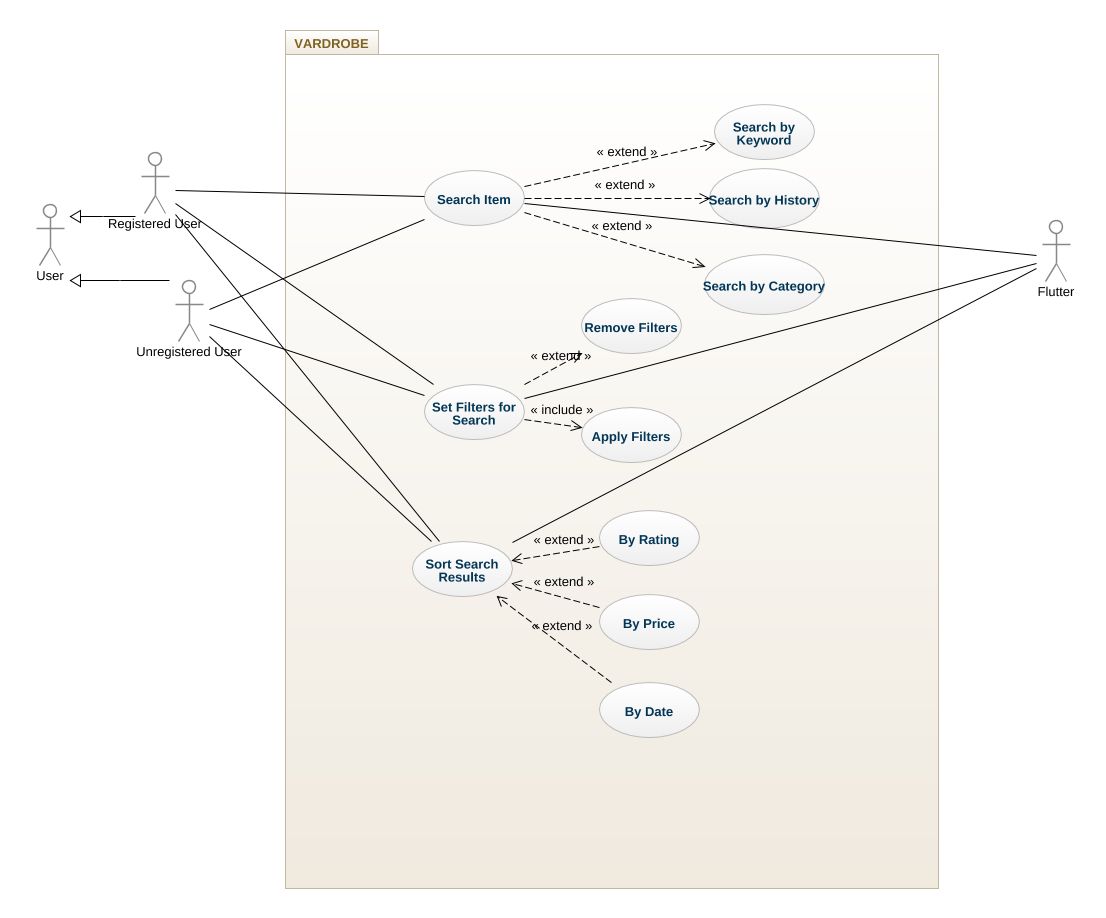
The use case technique will be applied to discover requirements for this system as this system highly focuses on user interaction with the system.

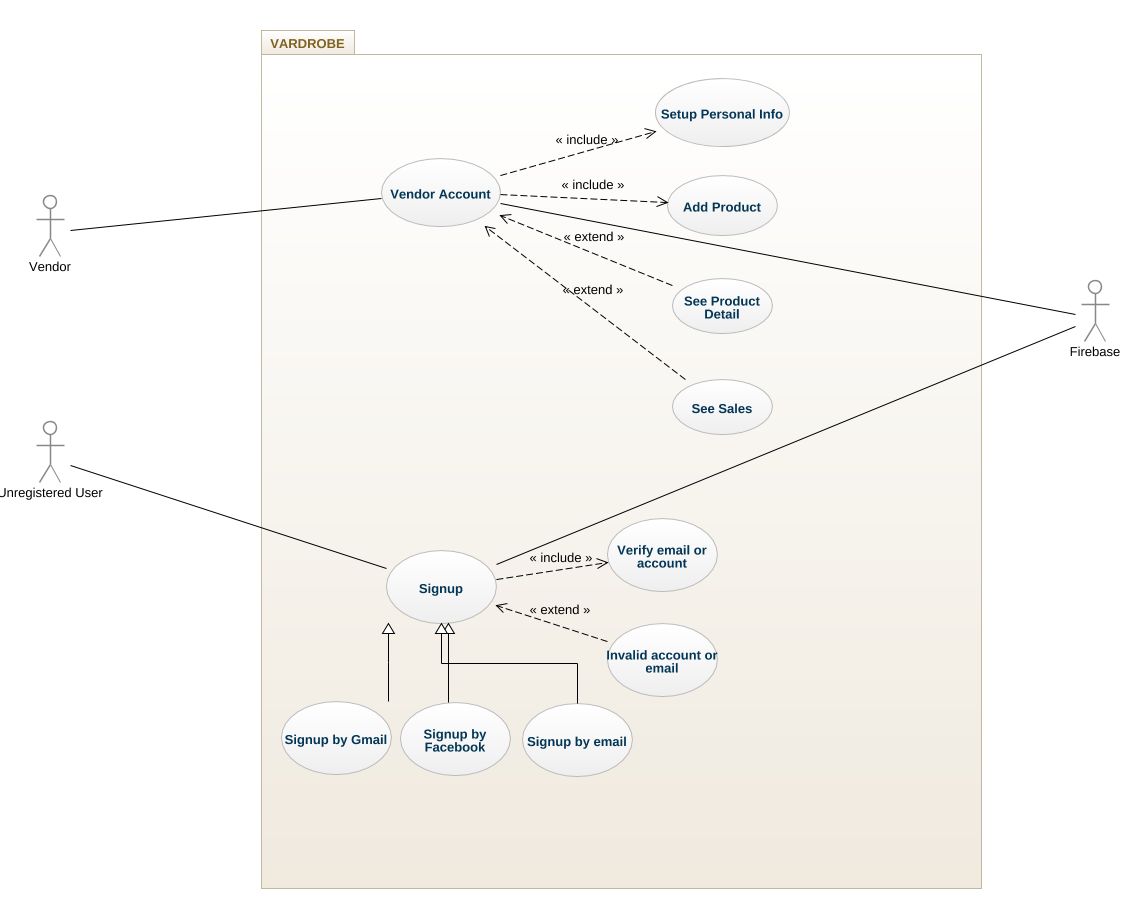
## Use Case Diagram

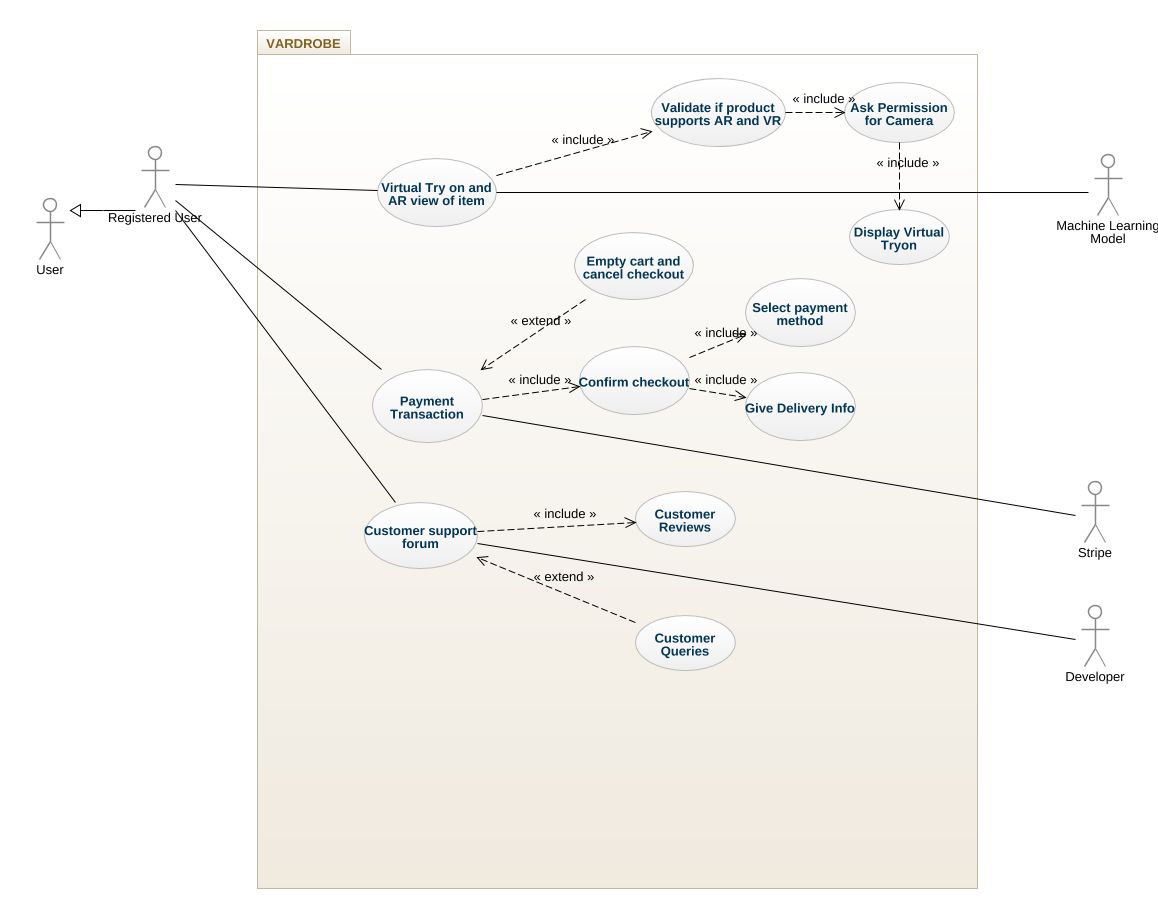
Note(First Diagram is Visio file if it doesn’t open zoom in the second diagram as it is png)











## Use Case Description

Table 1: Textual Description of Login Use Case

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1 |
| **Use Case Name:** | Login to the VARDROBE account |
| **Actors:** | Primary Actor: User, Secondary Actor: Firebase |
| **Description:** | A user tries to log in to his/her account using either their Facebook account or Gmail or by other email and password. The outcome should be a successful login if the credentials are correct and an error message if the credentials are invalid. |
| **Trigger:** | User types in his/her credentials and presses the login button. |
| **Preconditions:** | Users should have the app installed and should have an account registered with the app. |
| **Postconditions:** | VARDROBE access is provided to the user based on their account and the session is maintained. |
| **Normal Flow:** | * The user enters his/her username and password in the field. * The user presses the login button. * Firebase Auth verifies credentials. * App provides access to the user account. |
| **Alternative Flows:** | Users enter his/her credentials and prompt to login, the credentials are verified as invalid by firebase authentication, and access is denied to the user with an error message. |
| **Exceptions:** | * The user does not have a registered account * The user does not have a stable internet connection * The user enters invalid credentials |
| **Business Rules** | The time it takes for the user to log in should be less than 30 seconds. |
| **Assumptions:** | None |

Table 2: Textual Description of Logout Use Case

|  |  |
| --- | --- |
| **Use Case ID:** | UC-2 |
| **UseCase Name:** | Logout of VARDROBE account |
| **Actors:** | Primary Actor: User ,Secondary Actor:Firebase |
| **Description:** | A user tries to log out of his/her account. The outcome should be a successful logout from the user session and the system should take the user back to the login page. |
| **Trigger:** | The user presses the logout button in the user interface. |
| **Preconditions:** | User should be logged into their account. |
| **Postconditions:** | VARDROBE system logs the user out of their account and terminates the session. |
| **Normal Flow:** | * The user presses the logout button. * Firebase Auth logs out of the user account. * Firebase Auth terminates the user session. |
| **Alternative Flows:** | None |
| **Exceptions:** | * The user has not logged into the account. * The user does not have a stable internet connection. * The user prompts to logout and then closes the app. |
| **Business Rules** | The time it takes for the user to logout should be less than 15 seconds. |
| **Assumptions:** | None |

Table 3: Textual Description of Forgot Password

|  |  |
| --- | --- |
| **Use Case ID:** | UC-3 |
| **Use Case Name:** | The user forgot his/her credentials and now clicks the forgot password button. |
| **Actors:** | Primary Actor: User, Secondary Actor: Firebase |
| **Description:** | If a user forgets the credentials of the account they can prompt to reset the account password by clicking on the forgot password button, the system should then ask for the user email and verify that email and send a code to reset the password to that email. |
| **Trigger:** | The user clicks the forgot password button. |
| **Preconditions:** | User should have an email which he/she associated with an account to access this feature. |
| **Postconditions:** | When the user clicks the forgot password button the system takes them to a separate screen where it asks for email and then verifies it to reset the password. |
| **Normal Flow:** | * The user clicks on the forget password button. * The app takes the user to another screen. * The user enters a new password and confirms it. * The app updates the user password. |
| **Alternative Flows:** | User prompts to reset the password and then cancels the verification in email thus canceling the task of resetting the password |
| **Exceptions:** | * The user does not have an email associated with that account * The user does not have a stable internet connection |
| **Business Rules** | Users can only ask to reset the password twice a day. User if prompts for customer support, in this case, should be replied to in 24 hours. |
| **Assumptions:** | The user has an account related to the system |

Table 4: Textual Description of Search Item

|  |  |
| --- | --- |
| **Use Case ID:** | UC-4 |
| **Use Case Name:** | User Searches for an item by keyword |
| **Actors:** | Primary Actor: User, Secondary Actor: Flutter |
| **Description:** | The user types in a keyword in the search box to find an item, the system should display relevant results to that keyword. |
| **Trigger:** | The user types in a keyword in the search box and then presses the search icon button. |
| **Preconditions:** | User should be logged into their account and should type in a keyword at least 3 characters long. |
| **Postconditions:** | The system searches for items which are similar or are related to the keyword and displays the result in the view embedded in the user interface, |
| **Normal Flow:** | * User types in a keyword in the search bar. * The app searches the database for related products. * The app displays related products to the user. |
| **Alternative Flows:** | The user enters numeric in the search box and an error message is displayed on the screen.  The user enters a keyword for too long and no result returns.  The user enters a meaningless keyword that matches no results and no result found message is displayed on the screen. |
| **Exceptions:** | * The user enters numbers and special characters in the search box * The user does not have a stable internet connection |
| **Business Rules** | None |
| **Assumptions:** | There are products in the database |

Table 5: Textual Description of Set Filters for Search

|  |  |
| --- | --- |
| **Use Case ID:** | UC-5 |
| **Use Case Name:** | Set Filters related to items for the search query |
| **Actors:** | Primary Actor: User, Secondary Actor: Flutter |
| **Description:** | The user types in a keyword in the search box to find an item and then sets filters according to his/her liking. The system should display relevant results to that keyword and the filters set. |
| **Trigger:** | User types in a keyword in the search box then set filters and then presses the search icon button |
| **Preconditions:** | Users should be logged into their account and should type in a keyword at least 3 characters long and should press the set filter button to set filters. |
| **Postconditions:** | The system searches for items which are similar or are related to the keyword and the filters set for that search query and displays the result in the view embedded in the user interface, |
| **Normal Flow:** | * The user selects the filter option. * The user sets the filter types. * The user presses the go button. * The app displays results according to the filters set. |
| **Alternative Flows:** | The user enters numerics in the search box and an error message is displayed on the screen.  The user enters a keyword for too long and no result returns.  The user enters a meaningless keyword that matches no results and no result found message is displayed on the screen. |
| **Exceptions:** | * The user enters numbers and special characters in the search box * The user does not have a stable internet connection |
| **Business Rules** | None |
| **Assumptions:** | Filters are appropriate according to items. |

Table 6: Textual Description of Sort Search Result

|  |  |
| --- | --- |
| **Use Case ID:** | UC-6 |
| **Use Case Name:** | Sort the returned results according to filters |
| **Actors:** | Primary Actor: User, Secondary Actor: Flutter |
| **Description:** | When the user inputs a search phrase, then they can change how to sort the returned results either by ascending or descending price or by the best match to their search. |
| **Trigger:** | The user selects the sort button in the result screen |
| **Preconditions:** | User should be logged in to their account and has searched for the products they want to browse |
| **Postconditions:** | The system applies the sorting filter and then correct sorted results would be fed into the screen. |
| **Normal Flow:** | * The user clicks the sort button and the app displays a sort type list. * User sets the option according to which they want the items sorted. * The app displays items according to the sorting method set. |
| **Alternative Flows:** | User search for such a product that is not offered by the platform at the time, so no results and sort would be displayed |
| **Exceptions:** | * The user does not have a registered account * The user does not have a stable internet connection * The user enters invalid credentials * The record does not exist in the system. |
| **Business Rules** | None |
| **Assumptions:** | None |

Table 7: Textual Description of Recommend Similar Products

|  |  |
| --- | --- |
| **Use Case ID:** | UC-7 |
| **Use Case Name:** | Similar products recommended by the recommendation system |
| **Actors:** | Primary Actor: User, Secondary Actor: Machine Learning Model |
| **Description:** | When the user clicks on a product from the search results, the details about the product is displayed, when the user scroll downs to the end of the page they can find the recommendation about the similar products. |
| **Trigger:** | The description page is open and the user is currently browsing the details about the products. |
| **Preconditions:** | The user is logged in and clicks a product to see the item details. |
| **Postconditions:** | The recommended products that most accurately match the description would be classified and the top results are displayed. |
| **Normal Flow:** | * The user presses the checkout button when buying a product. * The app processes the user order and displays related items on the screen for the user. |
| **Alternative Flows:** | The item does not appear in search results because the platform does not offer such items. |
| **Exceptions:** | * The user does not have a registered account * The user does not have a stable internet connection * The user enters invalid credentials. * The platform does not sell the item. |
| **Business Rules** | The system must also have more products like the one selected. |
| **Assumptions:** | None |

Table 8: Textual Description of search on other websites

|  |  |
| --- | --- |
| **Use Case ID:** | UC-8 |
| **Use Case Name:** | Search on the other websites for the products |
| **Actors:** | Primary Actor: User, Secondary Actor: Scrapper API |
| **Description:** | A user wants to search the products on platforms other than VARDROBE. |
| **Trigger:** | The user clicks on any tab with the name of the platform, at the search results screen. |
| **Preconditions:** | The user is logged in and at the search screen result and clicks on the desired website results. |
| **Postconditions:** | The top matched products from another website should be displayed on the search results screen under the correct tab. |
| **Normal Flow:** | * The user enters a keyword related to the item they want. * Scrapper API searches for the related items on other platforms * The app displays results on the screen containing items related to the keyword. |
| **Alternative Flows:** | The other platforms do not have the searched products. |
| **Exceptions:** | * The user does not have a registered account * The user does not have a stable internet connection * The user enters invalid credentials. * VARDROBE does not support those platforms. |
| **Business Rules** | None. |
| **Assumptions:** | The other platforms have the desired products. |

Table 9: Textual Description of see item details

|  |  |
| --- | --- |
| **Use Case ID:** | UC-9 |
| **Use Case Name:** | Browse details about the desired product. |
| **Actors:** | Primary Actor: User ,Secondary Actor:Firebase |
| **Description:** | The user found the products they are looking for and wants to explore further about the product to see its description. |
| **Trigger:** | The user clicks on the search result box, related to the desired product. |
| **Preconditions:** | The user should be logged in and is currently on the search results page, the records are retrieved from the database and is displayed on the screen, |
| **Postconditions:** | The complete description of the product is grabbed from the database and is displayed on the item details screen. |
| **Normal Flow:** | * User presses on the item they are interested in. * The app takes the user to the product description screen. * App displays item description, price, and reviews along with item photos. |
| **Alternative Flows:** | The user enters an invalid search phrase and the record for that product does not exist in the system. |
| **Exceptions:** | * The user does not have a registered account * The user does not have a stable internet connection * The user enters invalid credentials. * The product does not exist in the system. |
| **Business Rules** | None |
| **Assumptions:** | None |

Table 10: Textual Description of Virtual Try on and AR view

|  |  |
| --- | --- |
| **Use Case ID:** | UC-10 |
| **Use Case Name:** | AR and Virtual Try one of the supported products. |
| **Actors:** | Primary Actor: User, Secondary Actor: Machine Learning model |
| **Description:** | A user wants to have to Augmented Reality view of the product or the user wants to have virtually tried the item in the comfort of their home. |
| **Trigger:** | The user clicks on the virtual try-on or AR view button for supported products. |
| **Preconditions:** | The user is logged in and the product support either Virtual try on or AR view or both in some cases. |
| **Postconditions:** | The system returns the AR view or Virtual try on for the product from the models stored on the cloud. |
| **Normal Flow:** | * The user clicks on the VR and AR view section of the app. * The app displays items that support AR and VR. * The user selects an item and is asked for camera access. * The user provides permission and is required to hold a camera to display a VR or AR view. |
| **Alternative Flows:** | User searches for the desired product but the product is either not from VARDROBE or the requested Virtual try on is not supported for the selected product. |
| **Exceptions:** | * The user does not have a registered account * The user does not have a stable internet connection * The user enters invalid credentials * The product is not offered by Vardrobe. * The product can not be virtually tried. |
| **Business Rules** | The virtual try-on and Augmented Reality view of the product should be of very high quality. |
| **Assumptions:** | None |

Table 11: Textual Description of Add to cart

|  |  |
| --- | --- |
| **Use Case ID:** | UC-11 |
| **Use Case Name:** | Add products to your cart. |
| **Actors:** | Primary Actor: User, Secondary Actor: Flutter |
| **Description:** | The user decides to buy a product from VARDROBE and click on the add to cart button, so the products they want to buy can proceed to checkout. |
| **Trigger:** | The user clicks on the add to cart button. |
| **Preconditions:** | The user is logged into the system. |
| **Postconditions:** | The product is added to the cart. |
| **Normal Flow:** | * Users click on the cart button on the product info screen. * Cart icon changes color and is filled now. * App increments the cart by and the user can continue to shop. |
| **Alternative Flows:** | When the user clicks on add to cart but the system runs out of stock for the current product. |
| **Exceptions:** | * The user does not have a registered account * The user does not have a stable internet connection * The user enters invalid credentials. |
| **Business Rules** | The stock must be maintained. |
| **Assumptions:** | The product is in stock. |

Table 12: Textual Description of Payment

|  |  |
| --- | --- |
| **Use Case ID:** | UC-12 |
| **Use Case Name:** | Select the payment method and add the details |
| **Actors:** | Primary Actor: User, Secondary Actor: Stripe API |
| **Description:** | A user wants to place the order from our platform and wants to proceed to the selection of the payment method. |
| **Trigger:** | The user clicks the button proceed to the payment. |
| **Preconditions:** | The user is logged into the system and the product is currently in the cart and the quantity and other user details are already in the system. |
| **Postconditions:** | The system creates a record of the order into the database, and the order is proceeded for delivery, a message showing that their method of payment has been updated is shown for confirmation. |
| **Normal Flow:** | * The user presses the button labeled as “ proceed to checkout”. * The app takes the user to the checkout screen asking for payment details. * Various payment methods are displayed to the user. * The user enters payment details and presses ok. * A secure transaction is made and the user is issued an order script. |
| **Alternative Flows:** | User searches for the product find the desired product, add the product into the cart, and does not have a valid ATM card. |
| **Exceptions:** | * The user does not have a registered account * The user does not have a stable internet connection * The user enters invalid credentials * The user does not have a valid ATM Card |
| **Business Rules** | Make contact with Stripe Payment gateway to integrate their services to the VARDROBE. |
| **Assumptions:** | None |

Table 13: Textual Description of the Customer Support forum

|  |  |
| --- | --- |
| **Use Case ID:** | UC-13 |
| **Use Case Name:** | Contact customer support |
| **Actors:** | Primary Actor: User, Secondary Actor: Developer |
| **Description:** | A user have some issues with their order or they have some query related to the platform, or they want to give their feedback to the developers so they decide to talk to the customer support. |
| **Trigger:** | The user selects the customer support option from their profile. |
| **Preconditions:** | A user is logged into the system and have some issue with the products and want to contact the VARDROBE team. |
| **Postconditions:** | A screen saying that your query has been submitted is displayed to the customer and the request is forwarded to the customer support representative and tries to resolve their issue |
| **Normal Flow:** | * The user presses the chat icon on the bottom of the screen. * The user enters a message containing a query or question. * The customer support team contacts the user and replies with a suitable answer. |
| **Alternative Flows:** | The user goes to the customer support section and they could not find their issue listed there so they decide to write an email to the support. |
| **Exceptions:** | * The user does not have a registered account * The user does not have a stable internet connection * The user enters invalid credentials |
| **Business Rules** | Provide top-notch customer support to their clients. |
| **Assumptions:** | The customer has a legit issue or feedback. |

Table 14: Textual Description of Managing the products

|  |  |
| --- | --- |
| **Use Case ID:** | UC-14 |
| **Use Case Name:** | Product and stock management |
| **Actors:** | Primary Actor: Admin, Secondary Actor: Firebase |
| **Description:** | The administration manages the products and keeps an eye on the stock so that the products are available to the customer for ordering most of the time. |
| **Trigger:** | The system runs low on the products or a new product is added. |
| **Preconditions:** | Admin is logged into the system, is currently on the admin panel to add new products to the website. |
| **Postconditions:** | Admin performs the addition or updates operation of a product and the updated data is visible to the customers, the records are added to the firebase. |
| **Normal Flow:** | * Admins or developers are responsible for the addition of products. * Admin logs into their account using the admin panel. * Admin can add a product or remove or update it. * Admin can manage vendors. |
| **Alternative Flows:** | Admin is logged into the system but the firebase server is not responding. |
| **Exceptions:** | * The admin is not logged into the account * The firebase server has some issues. |
| **Business Rules** | Have better relationships with the vendor and keep the stock filled. |
| **Assumptions:** | None |

Table 15: Textual Description of Manage vendor accounts

|  |  |
| --- | --- |
| **Use Case ID:** | UC-15 |
| **Use Case Name:** | Managing the vendor's account |
| **Actors:** | Primary Actor: Admin, Secondary Actor: Firebase |
| **Description:** | A user tries to log in to his/her account by the use of either their Facebook account or Gmail or by other email and password. The outcome should be a successful login if the credentials are correct and an error message if the credentials are invalid. |
| **Trigger:** | Admin adds a new vendor account or changing details related to vendors. |
| **Preconditions:** | Admin is logged in to their accounts and currently on the manage vendors page. |
| **Postconditions:** | The updated values are saved into the vendors' account and are added to the firebase. |
| **Normal Flow:** | * Vendor logs into their account using an email or an account. * Vendors can see their products, * Vendors can see their sales. |
| **Alternative Flows:** | Admin is logged into the system, but the firebase server is not responding. |
| **Exceptions:** | * The admin is not logged into the account * The firebase server has some issues. |
| **Business Rules** | None |
| **Assumptions:** | None |

# Functional Requirements

## Functional Requirement – I

|  |  |
| --- | --- |
| **Identifier** | FR-1 |
| **Title** | Search for item |
| **Requirement** | The user shall be able to search for the item from the search option by entering the relevant search query given that their input is within the correct search query range/ |
| **Source** | User |
| **Rationale** | If the user wants to buy a product from the platform, at first, they must search for the product, so a search functionality is the backbone of the system. |
| **Business Rule** | None |
| **Dependencies** | FR- XIV |
| **Priority** | High |

Table 16: Description of FR-1

## Functional Requirement – II

|  |  |
| --- | --- |
| **Identifier** | FR-II |
| **Title** | Sort search result |
| **Requirement** | The user shall be able to apply various sorting filter to the search results that are returned, the sorting techniques must be used to sort. |
| **Source** | User |
| **Rationale** | It would provide the user with a more personalized experience to the platform so they can quickly look up and find the products that they want. |
| **Business Rule** | None |
| **Dependencies** | FR-I |
| **Priority** | Medium |

Table 17: Description of FR-II

## Functional Requirement-III

|  |  |
| --- | --- |
| **Identifier** | FR-III |
| **Title** | Search Filters |
| **Requirement** | The user shall be able to add the filters, the filters shall be predefined in the system and should be able to run a query at the backend and then retrieve the results from the database. |
| **Source** | User |
| **Rationale** | The search personalization would help the user to get a more in-depth look from the search results that would help them in purchasing the best-matched product from the platform |
| **Business Rule** | Any restriction, policy, the rule that the requirement must be fulfilled through its functional behavior |
| **Dependencies** | FR-II |
| **Priority** | Medium |

Table 18: Description of FR-III

## Functional Requirement – IV

|  |  |
| --- | --- |
| **Identifier** | FR-IV |
| **Title** | Similar products recommendation |
| **Requirement** | The user shall be able to get the recommendation about the similar products that they are currently browsing, the products must accurately match the product descriptions, the recommendation should be made by the machine learning model running in a cloud. |
| **Source** | User |
| **Rationale** | To implement the most widely used and appreciated feature on an e-commerce website that would boost the sales for the vendors. |
| **Business Rule** | Focus on adding more similar products so that every vendor is happy with the platform. |
| **Dependencies** | FR-II |
| **Priority** | High |

Table 19: Description of FR-IV

## Functional Requirement – V

|  |  |
| --- | --- |
| **Identifier** | FR-V |
| **Title** | Scrape data from other websites |
| **Requirement** | Description of requirement which may be written either from the user or system perspective e.g.  The user shall be able to view the data from other supported platforms from Vardrobe when they click on the respective tab of that platform, the product must be loaded in real-time and it should be the latest products in the websites’ catalog |
| **Source** | User |
| **Rationale** | Vardrobe wants to provide the AR view for the products from the other websites as well, so the user won’t have to open multiple tabs in their browser rather than they can view the details about the product under one umbrella. |
| **Business Rule (if required)** | Scrape the data from the publicly available website, so it comes within the legal data. |
| **Dependencies** | FR-II |
| **Priority** | High |

Table 20: Description of FR-V

## Functional Requirement – VI

|  |  |
| --- | --- |
| **Identifier** | FR-VI |
| **Title** | Items details |
| **Requirement** | The user shall be able to view the details of the desired product when they click on the card that contains the information about the specific product. |
| **Source** | User |
| **Rationale** | If the user buys the product it would generate revenue for Vardrobe that would boost the business, so this feature is most important, among others. |
| **Business Rule** | Any restriction, policy, the rule that the requirement must be fulfilled through its functional behavior |
| **Dependencies** | FR-II |
| **Priority** | High |

Table 21: Description of FR-VI

## Functional Requirement – VII

|  |  |
| --- | --- |
| **Identifier** | FR-VII |
| **Title** | Augmented Reality view of the product |
| **Requirement** | The user must be able to get the Augmented Reality view of the supported product when the user clicks on the related button. |
| **Source** | User |
| **Rationale** | This is the backbone of the ideology that led us to create Vardrobe to give the user a sense of the real experience from a virtual environment. |
| **Business Rule** | Attract the customer to change their way of online shopping, to move from the traditional mode of shopping toward the future of shopping. |
| **Dependencies** | FR-II, FR-VI |
| **Priority** | High |

Table 22: Description of FR-VII

## Functional Requirement – VIII

|  |  |
| --- | --- |
| **Identifier** | FR-VIII |
| **Title** | Virtual try-on |
| **Requirement** | The user shall be able to get the virtual try-on from their camera of the supported products from the platform. |
| **Source** | User |
| **Rationale** | This is the main concept behind the VARDROBE, which will allow its customers to virtually try on the products in real-time without going to the shop physically. |
| **Business Rule** | Attract the customers to be able to virtually try on their favorite products from the ease of their home. |
| **Dependencies** | FR-II, FR-VI |
| **Priority** | High |

Table 23: Description of FR-VIII

## Functional Requirement – IX

|  |  |
| --- | --- |
| **Identifier** | FR-IX |
| **Title** | Adding item to the cart |
| **Requirement** | The user should be able to add the products that they want to purchase from the platform inside their cart so that they can proceed to the checkout, the right product should be added to the cart if it is in stock and the cart should be updated after every product entry. |
| **Source** | User |
| **Rationale** | If the user adds the item to the cart so then we are one step closer to generating revenue by selling our products on VARDROBE. |
| **Business Rule** | Keep an eye on the inventory status. |
| **Dependencies** | FR-II |
| **Priority** | Medium |

Table 24: Description of FR-IX

## Functional Requirement – X

|  |  |
| --- | --- |
| **Identifier** | FR-X |
| **Title** | Adding payment gateway for ATM transactions |
| **Requirement** | The user should be able to choose from multiple options of the payment, among them if a user wants to pay via ATM card they should be able to perform the transactions, their account should be verified |
| **Source** | User |
| **Rationale** | If the user places an order from the platform it would lead towards the success of the application. |
| **Business Rule** | Keep good terms with Stripe. |
| **Dependencies** | FR-II |
| **Priority** | Medium |

Table 25: Description of FR-X

## Functional Requirement – XI

|  |  |
| --- | --- |
| **Identifier** | FR-XI |
| **Title** | Customer Support |
| **Requirement** | The user should be able to register their complaint by using the feature of customer support. |
| **Source** | User |
| **Rationale** | The success of any platform includes getting the trust of the customer to build the customer base, so this feature should be unprecedented to gain the loyalty of the customer. |
| **Business Rule** | Strive to have to trust the customer over VARDROBE and its products. |
| **Dependencies** | FR-XIV |
| **Priority** | High |

Table 26: Description of FR-XI

## Functional Requirement – XII

|  |  |
| --- | --- |
| **Identifier** | FR-XII |
| **Title** | Manage the inventory |
| **Requirement** | The admin should be able to manage the inventory, add new products, and keep an eye on the stock. |
| **Source** | Admin |
| **Rationale** | The inventory should be maintained to be consistent in the competitive world of online shopping. |
| **Business Rule** | None |
| **Dependencies** | FR-IX |
| **Priority** | High |

Table 27: Description of FR-XII

## Functional Requirement – XIII

|  |  |
| --- | --- |
| **Identifier** | FR-XIII |
| **Title** | Manage the vendor’s accounts |
| **Requirement** | The admin should be able to manage the vendor account by getting logged into their admin account, with ease so that the products are readily available and new products keep getting updated in the inventory. |
| **Source** | Admin |
| **Rationale** | Keep the inventory updated so the customers keep coming back to the platform from repeating their buying practice. |
| **Business Rule** | Keep good relations with the vendors. |
| **Dependencies** | FR-IX |
| **Priority** | High |

Table 28: Description of FR-XIII

## Functional Requirement – XIV

|  |  |
| --- | --- |
| **Identifier** | FR-XIV |
| **Title** | Login to the account |
| **Requirement** | The user should be able to log in to their account so that they can use the services provided by VARDROBE, there should be multiple options to log in to the account either by registering with their email, connecting their Google account or by using their Facebook account. |
| **Source** | User |
| **Rationale** | The login process is the first step toward having a potential customer so this step must be performed with great care and ease. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

Table 29: Description of FR-XIV

## Functional Requirement – XV

|  |  |
| --- | --- |
| **Identifier** | FR-XV |
| **Title** | Logout |
| **Requirement** | The user should be able to log out from their account just by clicking a button, their session should be deactivated from the server. |
| **Source** | User |
| **Rationale** | To protect user privacy, the user should be able to log out so that their details are not seen by some unwanted person. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Low |

Table 30: Description of FR-XV

## Functional Requirement – XVI

|  |  |
| --- | --- |
| **Identifier** | FR-XVI |
| **Title** | Email verification |
| **Requirement** | When the user sign-ups for the system using their email address an email should be sent to their inbox for account confirmation and they should only be able to use the system after confirming their email. |
| **Source** | Firebase |
| **Rationale** | To protect the system from getting spam users and bots so that system should be available to legit users only |
| **Business Rule** | Try to get the maximum legit user for the platform. |
| **Dependencies** | FR-XIV |
| **Priority** | Medium |

Table 31: Description of FR-XVI

## Functional Requirement – XVII

|  |  |
| --- | --- |
| **Identifier** | FR-XVII |
| **Title** | SMS verification |
| **Requirement** | When the user sign-ups for the system using their email address a message with a confirmation code should be sent to their registered mobile number to get the confirmation of their account. |
| **Source** | Firebase |
| **Rationale** | To protect the system from getting spam users and bots so that system should be available to legit users only |
| **Business Rule** | Try to get the maximum legit user for the platform. |
| **Dependencies** | FR-XIV |
| **Priority** | Medium |

Table 32: Description of FR-XVII

## Functional Requirement – XVIII

|  |  |
| --- | --- |
| **Identifier** | FR-XVIII |
| **Title** | Unique email for every user |
| **Requirement** | Every user should only be allowed to have one account email address so every user can avail of the services of the platform. |
| **Source** | Firebase |
| **Rationale** | To make sure that the system meets the Non-Functional requirements of the availability. |
| **Business Rule** | Try to get the maximum legit user for the platform. |
| **Dependencies** | None |
| **Priority** | High |

Table 33: Description of FR-XVIII

## Functional Requirement – XIX

|  |  |
| --- | --- |
| **Identifier** | FR-XIX |
| **Title** | Unique mobile number for every user |
| **Requirement** | Each user should only be allowed to have one account per mobile number so that the system will not have to bear the extra load of bogus users. |
| **Source** | Firebase |
| **Rationale** | To make sure that the system is available for the legit users for the maximum period. |
| **Business Rule** | Try to get the maximum legit user for the platform. |
| **Dependencies** | None |
| **Priority** | High |

Table 33: Description of FR-XIX

## Functional Requirement – XX

|  |  |
| --- | --- |
| **Identifier** | FR-XX |
| **Title** | Keeping the password secure |
| **Requirement** | Every user password should not be stored in plain text rather than the hashed function should be used to encrypt the password. |
| **Source** | Firebase |
| **Rationale** | To protect the privacy of the user and get their trust on the platform |
| **Business Rule** | None |
| **Dependencies** | FR-XIV |
| **Priority** | High |

Table 34: Description of FR-XX

## Functional Requirement – XXI

|  |  |
| --- | --- |
| **Identifier** | FR-XXI |
| **Title** | Account verification before resetting the password |
| **Requirement** | When the user wants to reset their password for any reason, they should first verify their identity by entering their old password before resetting the password. |
| **Source** | User |
| **Rationale** | To make sure that only legit user is resetting the password. |
| **Business Rule** | None |
| **Dependencies** | FR-XIV |
| **Priority** | Medium |

Table 35: Description of FR-XXI

## Functional Requirement – XXII

|  |  |
| --- | --- |
| **Identifier** | FR-XXII |
| **Title** | User with the unique id |
| **Requirement** | Every user should have a unique id so that every user should be identified, and they should be able to use the application without any interference. |
| **Source** | Firebase |
| **Rationale** | Simplify the management of the users. |
| **Business Rule** | To make sure that any action that needs to be taken is being taken for the correct user. |
| **Dependencies** | None |
| **Priority** | Low |

Table 36: Description of FR-XXII

## Functional Requirement – XXIII

|  |  |
| --- | --- |
| **Identifier** | FR-XXIII |
| **Title** | Product with the unique id |
| **Requirement** | Every product should be identified by a unique id so that the complaints and order related to that specific product should be resolved. |
| **Source** | Firebase |
| **Rationale** | Simplify the management of the products. |
| **Business Rule** | To make sure that any action that needs to be taken is being taken against the correct product. |
| **Dependencies** | None |
| **Priority** | Low |

Table 37: Description of FR-XXIII

## Functional Requirement – XXIV

|  |  |
| --- | --- |
| **Identifier** | FR-XXIV |
| **Title** | The vendor with the unique id |
| **Requirement** | Every vendor should be assigned a separate id so that the issues and products of the specific vendor could be resolved. |
| **Source** | Admin |
| **Rationale** | Simplify vendor management. |
| **Business Rule** | To make sure that any action that needs to be taken is being taken for the right vendor. |
| **Dependencies** | None |
| **Priority** | Low |

Table 38: Description of FR-XXIV

## Functional Requirement – XXV

|  |  |
| --- | --- |
| **Identifier** | FR-XXV |
| **Title** | Categorizing the products |
| **Requirement** | Every product should be stored under the right category so if the user filters the search by product type then the correct product would be displayed. |
| **Source** | Flutter |
| **Rationale** | Make the management of the product as simple as it can be. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Low |

Table 39: Description of FR-XXV

## Functional Requirement – XXVI

|  |  |
| --- | --- |
| **Identifier** | FR-XXVI |
| **Title** | Quality of AR |
| **Requirement** | When the user wants to get the AR view of the product the rendered quality of the product should be the highest possible so that the user would have the best idea of the product's actual image. |
| **Source** | Augmented Reality models/ Flutter |
| **Rationale** | If the quality of the AR view is good m it would attract the user and leads to the success of the system. |
| **Business Rule** | Make sure to use high-quality AR. |
| **Dependencies** | FR-VII |
| **Priority** | High |

Table 40: Description of FR-XXVI

## Functional Requirement – XXVII

|  |  |
| --- | --- |
| **Identifier** | FR-XXVII |
| **Title** | Management of Virtual try on based products |
| **Requirement** | When the user selects a product for which the Virtual try-on is not supported then the option to get the virtual try-on should not be visible on the screen. |
| **Source** | Virtual try-on model |
| **Rationale** | Products whose Virtual try on is not possible should be separated to avoid any confusion. |
| **Business Rule** | None |
| **Dependencies** | FR-VIII |
| **Priority** | Medium |

Table 41: Description of FR-XXVII

## Functional Requirement – XXVIII

|  |  |
| --- | --- |
| **Identifier** | FR-XXVIII |
| **Title** | Availability of the Augmented reality |
| **Requirement** | When the user clicks on the specific product the AR view of the product should be ready to be displayed on the screen. |
| **Source** | User |
| **Rationale** | This is the basic idea behind Vardrobe this functionality should be at the highest quality possible. |
| **Business Rule** | Attract the maximum customers by offering an AR view of the product |
| **Dependencies** | FR-VII |
| **Priority** | Low |

Table 42: Description of FR-XXVIII

## Functional Requirement – XXIX

|  |  |
| --- | --- |
| **Identifier** | FR-XXIX |
| **Title** | Ratings of the products |
| **Requirement** | When the user searches for the product the rating for that product should be listed along with the product details. |
| **Source** | User |
| **Rationale** | The rating should be available for every product and it should be listed among other details, so the user won’t have to click the item and scroll down to get information. |
| **Business Rule** | None |
| **Dependencies** | FR-XIV |
| **Priority** | Low |

Table 43: Description of FR-XXIX

## Functional Requirement – XXX

|  |  |
| --- | --- |
| **Identifier** | FR-XXX |
| **Title** | Review of the products |
| **Requirement** | The user should be able to see the review of the products added by other users. |
| **Source** | User |
| **Rationale** | The reviews should be accurate so that the user can make their mind while buying the product. |
| **Business Rule** | None |
| **Dependencies** | FR-VI |
| **Priority** | Low |

Table 44: Description of FR-XXX

## Functional Requirement – XXXI

|  |  |
| --- | --- |
| **Identifier** | FR-XXXI |
| **Title** | Images with reviews |
| **Requirement** | The user should be able to add images of the received products in their product review. |
| **Source** | User |
| **Rationale** | Organize the display of the reviews accurately. |
| **Business Rule** | None |
| **Dependencies** | FR-XXX |
| **Priority** | Low |

Table 45: Description of FR-XXXI

## Functional Requirement – XXXII

|  |  |
| --- | --- |
| **Identifier** | FR-XXXII |
| **Title** | Order placement |
| **Requirement** | The user should be able to place an order for the product they want to buy. |
| **Source** | User |
| **Rationale** | Product ordering is one way to generate revenue for the system so this step must be implemented correctly. |
| **Business Rule** | Facilitates the users as much as is possible. |
| **Dependencies** | FR-IX |
| **Priority** | High |

Table 46: Description of FR-XXXII

## Functional Requirement – XXXIII

|  |  |
| --- | --- |
| **Identifier** | FR-XXXIII |
| **Title** | Cash on delivery |
| **Requirement** | The user should also be able to select the most widely used model of paying the cash on delivery within the application by placing their details. |
| **Source** | User |
| **Rationale** | The user order must be recorded correctly and should be notified to the vendor. |
| **Business Rule** | Make deal with the courier service to allow the smooth delivery of the products |
| **Dependencies** | FR-XXXII |
| **Priority** | Medium |

Table 47: Description of FR-XXXIII

## Functional Requirement – XXXIV

|  |  |
| --- | --- |
| **Identifier** | FR-XXXIV |
| **Title** | Store personal information |
| **Requirement** | The personal information like address and the contact number of the user should be stored for the safekeeping. |
| **Source** | Firebase |
| **Rationale** | The information must be stored correctly against every user. |
| **Business Rule** | Personal information should be kept private from any third party. |
| **Dependencies** | None |
| **Priority** | Medium |

Table 48: Description of FR-XXXIV

## Functional Requirement – XXXV

|  |  |
| --- | --- |
| **Identifier** | FR-XXXV |
| **Title** | Order history |
| **Requirement** | The user should be able to see their order history in their profile. |
| **Source** | User |
| **Rationale** | The profile of the user should reflect the correct information. |
| **Business Rule** | None |
| **Dependencies** | Non |
| **Priority** | Low |

Table 49: Description of FR-XXXV

## Functional Requirement – XXXVI

|  |  |
| --- | --- |
| **Identifier** | FR-XXXVI |
| **Title** | Track order status |
| **Requirement** | When the user places their order they should be able to track the status of the order under their profile section. |
| **Source** | User |
| **Rationale** | The order status must update after every cycle of delivery. |
| **Business Rule** | Get better connections with the courier services. |
| **Dependencies** | None |
| **Priority** | Medium |

Table 50: Description of FR-XXXVI

## Functional Requirement – XXXVII

|  |  |
| --- | --- |
| **Identifier** | FR-XXXVII |
| **Title** | Add products to favorite |
| **Requirement** | The user should be able to make a collection of their favorite products so that they can look at it later. |
| **Source** | User |
| **Rationale** | The favorites of the user would provide a faster way to find a product that they are looking for, |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Low |

Table 51: Description of FR-XXXVII

## Functional Requirement – XXXVIII

|  |  |
| --- | --- |
| **Identifier** | FR-XXXVIII |
| **Title** | Delete product from the cart |
| **Requirement** | The user should be able to delete the products that they do not want to buy from their cart before checkout |
| **Source** | User |
| **Rationale** | The products that got deleted should no longer be visible in the cart and the checkout price. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Low |

Table 52: Description of FR-XXXVII

## Functional Requirement – XXXIX

|  |  |
| --- | --- |
| **Identifier** | FR-XXXIX |
| **Title** | Delete products from favorites |
| **Requirement** | User should be able to delete products from the favorites whenever they want |
| **Source** | User |
| **Rationale** | The products that got deleted should no longer be visible in favorites. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Low |

Table 53: Description of FR-XXXIX

## Functional Requirement – XL

|  |  |
| --- | --- |
| **Identifier** | FR-XL |
| **Title** | Notify vendor of new orders |
| **Requirement** | The vendor should be notified whenever their product gets a new order from a customer within 30 minutes. |
| **Source** | Admin |
| **Rationale** | If the vendor does not get the notification of the delivery on time then it can cause the system loss of a potential client. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

Table 54: Description of FR-XL

## Functional Requirement – XLI

|  |  |
| --- | --- |
| **Identifier** | FR-XLI |
| **Title** | Notify vendor of low stock |
| **Requirement** | If the product is running low on supply, then the admin must notify the vendors to fill their stock. |
| **Source** | User |
| **Rationale** | If the vendor does not get the notification of low stock on time, then it can cause the system loss of a potential client. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

Table 55: Description of FR-XLI

## Functional Requirement – XLII

|  |  |
| --- | --- |
| **Identifier** | FR-XLII |
| **Title** | Stock status |
| **Requirement** | The stock status of a specific product must be displayed to the user with a clear message stating whether the product is available or not. |
| **Source** | Firebase |
| **Rationale** | The status must be accurate and must update every instance from the database, so the user won’t face difficulty in placing the order. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Low |

Table 56: Description of FR-XLII

## Functional Requirement – XLIII

|  |  |
| --- | --- |
| **Identifier** | FR-XLIII |
| **Title** | Checkout Calculation |
| **Requirement** | The user should be able to see their total bill before placing the order. |
| **Source** | Firebase/Flutter |
| **Rationale** | The calculations should be accurate and must not include any hidden charges |
| **Business Rule** | Noe |
| **Dependencies** | None |
| **Priority** | Low |

Table 57: Description of FR-XLIII

## Functional Requirement – XLIV

|  |  |
| --- | --- |
| **Identifier** | FR-XLIV |
| **Title** | Checkout |
| **Requirement** | The user must be able to able to perform checkout after selecting their payment method this should not take longer than 1 minute. |
| **Source** | Firebase/ Flutter |
| **Rationale** | The final goal of any business is to get the orders, this functionality must be implemented correctly. |
| **Business Rule** | None |
| **Dependencies** | FR-XLIII |
| **Priority** | Medium |

Table 58: Description of FR-XLIV

## Functional Requirement – XLV

|  |  |
| --- | --- |
| **Identifier** | FR-XLV |
| **Title** | Human pose Representation |
| **Requirement** | The system should be able to estimate the different poses of the body of every user irrespective of their physique and should be able to map the poses. |
| **Source** | Machine Learning model |
| **Rationale** | The poses must be calculated accurately so that the virtual try-on module can work as intended. |
| **Business Rule** | None |
| **Dependencies** | FR-XXVII |
| **Priority** | Medium |

Table 59: Description of FR-XLV

## Functional Requirement – XLVI

|  |  |
| --- | --- |
| **Identifier** | FR-XLVI |
| **Title** | Generative Adversarial Network |
| **Requirement** | The GAN should be able to train the virtual try-on model to generate the images with the use of Variational Autoencoder loss. |
| **Source** | Machine Learning model |
| **Rationale** | GAN architecture must have a substantial number of Input and hidden layers with the relevant activation functions and losses. |
| **Business Rule** | None |
| **Dependencies** | FR-XXVII |
| **Priority** | Medium |

Table 60: Description of FR-XLVI

## Functional Requirement – XLVII

|  |  |
| --- | --- |
| **Identifier** | FR-XLVII |
| **Title** | Clothing representation |
| **Requirement** | The selected dress should be able to correctly present by the accurate dimensions. |
| **Source** | Machine Learning model |
| **Rationale** | The dress representation should be aligned with the human pose representation. |
| **Business Rule** | None |
| **Dependencies** | FR-XXVII |
| **Priority** | High |

Table 61: Description of FR-XLVII

## Functional Requirement – XLVIII

|  |  |
| --- | --- |
| **Identifier** | FR-XLVIII |
| **Title** | Fitting of the item on the image |
| **Requirement** | The model should be able to fit the dress on the picture uploaded by the users at the highest accuracy possible. |
| **Source** | Machine Learning model |
| **Rationale** | This is the backbone feature of the application and should be implemented with great care. |
| **Business Rule** | None |
| **Dependencies** | FR-XXVII |
| **Priority** | High |

Table 62: Description of FR-XLVIII

## Functional Requirement – XLIX

|  |  |
| --- | --- |
| **Identifier** | FR-XLIX |
| **Title** | 3D model rendering |
| **Requirement** | The 3d model of the product should be fetched and should be rendered. |
| **Source** | Augmented Reality model |
| **Rationale** | This is another important feature of the application and the model rendering should be handled with care. |
| **Business Rule** | None |
| **Dependencies** | FR-XXVII |
| **Priority** | High |

Table 63: Description of FR-XLIX

# Non-Functional Requirements

## Usability

*USE-1: The system should be very easy to use for people even with low or no previous experience with the application use.*

*USE-2: The system shall allow the user to see the details of their previous order with a single interaction.*

*USE-3: The system users should be able to recover their password with great ease.*

*USE-4: The system user should find themselves comfortable with using the system in their first interaction with the application.*

## Performance

*PER-1: The system should be able to retrieve the search queries in a matter of seconds while keeping in mind the connection speed of the customer.*

*PER-2: The system’s output of the Augmented Reality view should be at it’s highest depending upon the hardware capabilities of the user’s device.*

*PER-3: The system’s virtual try-on feature should produce high quality depending upon the hardware of the user’s device.*

*PER-4: The system should not consume the hardware capabilities unnecessarily.*

## Security

*SEC-1: The system should be secure from any kind of intrusion from any unwanted entity.*

*SEC-2: The user should be able to reset their password safely without getting worried that their information.*

*SEC-3: The system should only request to access the required permission from the user that is needed for the system to work.*

## Scalability

*SCAL-1: The system must be able to accompany any further expansion to accommodate more and more users.*

## Integrity

*INT-1: The system must represent accurate data to the user.*

*INT-2: The data must be consistent in every part of the application; this also implies to the database.*

## Maintainability

*MAINT-1: The system should be developed in such a manner that the bugs can be fixed.*

*MAINT-2: The system should be organized in a pattern so that each component is independent of each other yet connected.*

## Installibility

*INS-1: Vardrobe should be easy to install, any updates should be installed with ease without the interaction of the user.*

# External Interface Requirements

## User Interfaces

|  |  |
| --- | --- |
| **Application Theme** | Material Theme would be used for the designing of the mobile application the buttons would [3] |
| **Fonts** | Font Awesome, Google Fonts, Custom Fonts |
| **Icons** | Font Awesome Icons, Cupertino Icons, and more font packages |
| **GUI** | Flutter development kit |
| **Navigation** | Material page routes (Flutter) |
| **Layout** | Material App containing safe area and scaffolds and many more widgets, all provided by flutter. |
| **Buttons** | Flat Button, drop-down button, Button bar, Icon button, floating action button, and much more using flutter |
| **Message displays** | Alert dialog and Cupertino Alert dialog, plus custom created alert dialog. |

Table 64:User Interface requirements

## Software Interfaces

* VARDROBE would be developed on a hybrid based mobile development environment developed and supported by Google called Flutter. The latest version of Flutter is 1.17 (stable).[1]
* VARDROBE would be developed on Windows 10 version 2004.
* VARDROBE would be tested both physically and virtually on the Android devices running on the latest release of Android 10.
* VARDROBE would be tested on a Physical iOS device running the latest iOS version.
* Firebase would be used as a NoSQL cloud database to store and retrieve the information in real-time, the flutter and firebase both are developed and maintained by Google and they go hand in hand with each other, all the required data of the user would be stored on Google’s powerful cloud servers to minimize the downtime of the application.
* Selenium version 4.0+, a web testing framework would be used, to scrape the javascript-based website data and the data would be displayed in the search results screen of VARDROBE, scrapy can also be used If needed.
* Stripe API would be used as a payment gateway.[3]
* AWS Sage maker would be used as a cloud service to deploy the machine learning models, the data from the AWS servers would be grabbed by the API and would be displayed at the places needed in the VARDROBE.

## Hardware Interfaces

The following are the hardware requirement of a mobile application for the application to run.

* **Storage: 16 GB+**
* **RAM: 4GB+, 8 recommended**
* **CPU: Octa-core processor is recommended**
* **WIFI 802.11**
* **GPU: Adreno 618 or equivalent**
* **Main Camera**
* **Front camera**
* **Network Interface card**

## Communication Interfaces

* Google OAuth would be used for the users who want to login to the Vardrobe using their Google accounts.
* Facebook OAuth would be used for the users so that they can log in with their Facebook account if they want.
* If a user opts to login by email his mobile number would be verified by sending a message to their mobile number and prompting them to add the received pin.
* HTTPS POST protocol would be used to POST data onto the server.
* HTTPS GET protocol would be used to get the information from a server.
* The application like any other application would be vulnerable to outside attacks but SHA256 encryption and HTTPS protocol would be used to secure the communication layer.

# References

* 1. Flutter Team, Flutter documentation <https://flutter.dev/docs>
  2. Stripe team, Stripe documentation, <https://stripe.com/docs>
  3. Material Theme, https://material.io/resources/build-a-material-theme

# Plagiarism Report

