Ain Shams University
Faculty of Computer & Information Sciences
Computer Science Department

TRY FIT

July 2025

Ain Shams University
Faculty of Computer & Information Sciences
Computer Science Department

TRY FIT: VTON Egyptian Brands

By:

Genyveyav Raafat Louka [CS]
Hassnaa Hassan Saied [CS]
Habiba Mohammed Yahia [CS]
Toqa Osama Ali [CS]
Monica Adel Lotfy [CS]
Maria George Kamel [CS]

Under Supervision of:

Prof. Dr Abeer Mahmoud
Professor of Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

T.A. Mohamed Essam
Bioinformatics Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

Acknowledgement

First and foremost, all praise and gratitude are due to **Allah**, whose boundless mercy and guidance have empowered us throughout this journey. Without His blessings, we would not have had the strength, patience, or clarity to complete this work. We pray that this project serves a meaningful purpose

We would like to express our deepest appreciation to our **families**, especially our parents, for their constant encouragement, prayers, and unconditional love. Their unwavering belief in us and sacrifices behind the scenes have played a crucial role in our ability to reach this milestone. Words fall short of truly capturing the impact they have had on our academic and personal growth.

We are also sincerely thankful to our dedicated supervisors, **Prof. Dr. Abeer Mahmoud** and **T.A. Mohamed Essam**, whose mentorship, constructive feedback, and consistent support have guided us through each stage of this project. Their patience, clarity, and commitment to our development helped transform challenges into opportunities for learning.

Additionally, we extend our gratitude to all our **friends and colleagues** who walked alongside us through long nights, tight deadlines, and moments of doubt. Your support, whether through brainstorming, motivation, or even lighthearted distractions, made this journey more bearable and memorable.

To everyone who contributed to this project in any way—directly or indirectly—please accept our heartfelt thanks. This achievement is not ours alone; it is built upon the support, guidance, and encouragement we've received from each of you.

Abstract

In the era of online retail, one of the major challenges faced by fashion consumers is the inability to accurately visualize how clothing items would appear on their own bodies. This limitation is even more pronounced in regions where modest fashion is prevalent, such as Egypt and the broader MENA region, where existing VTON systems often overlook cultural and privacy considerations. To address this gap, **TryFit** was developed — an intelligent VTON system designed with cultural awareness and inclusivity at its core.

TryFit allows users to virtually try on upper, lower, or full-body garments, including hijabs and veiled outfits, simply by uploading their personal images. The system integrates several advanced computer vision techniques, including semantic segmentation, pose estimation, cloth warping, and image synthesis using a customized LaDI-VTON backend. Additionally, a classical preprocessing module was introduced to selectively black out facial regions based on label color segmentation, thereby enhancing privacy for veiled users and improving alignment focus during generation.

The platform is implemented using a modular backend powered by Flask and PyTorch, connected to a Firebase cloud database and presented through a user-friendly FlutterFlow mobile interface. The final system supports real-time try-on with high garment fidelity, accurate alignment to body pose, and realistic texture preservation. Experimental evaluation across diverse datasets demonstrated strong performance in handling a wide variety of garment types and user poses, with particular success in veiled dress synthesis. The results confirm TryFit's ability to produce high-quality, culturally respectful virtual try-on outputs — ultimately improving user confidence, reducing return rates, and supporting local fashion brands through inclusive digital experiences.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	What the abbreviation stands for
CLIP	Contrastive Language-Image Pretraining
CNN	Convolutional Neural Network
EMASC	Enhanced Mask-Aware Skip Connections
GMM	Geometric Matching Module
LDM	Latent Diffusion Model
MENA	Middle East and North Africa
MLP	Multi-Layer Perceptron
NN	Neural Network
SCHP	Self-Correction Human Parsing
TPS	Thin Plate Spline
VAE	Variational Autoencoder
VTON	Virtual Try-On

1- Introduction

1.1 Motivation

Our team is driven by the growing challenges faced in online shopping, particularly the inability of users to visualize how clothing items would appear on them. To address this limitation, we are developing an intelligent VTON system designed to enhance the user experience by enabling more informed and confident purchasing decisions. By integrating advanced image processing and personalization techniques, our solution aims to bridge the gap between physical and digital shopping, ultimately improving customer satisfaction and reducing return rates.

In doing so, we also aim to support underrepresented user groups, such as veiled women, by offering modesty-conscious try-on features. Our goal is to create an inclusive, culturally respectful platform that reflects the diverse needs of modern fashion consumers.

1.2 Problem Definition

In Egypt's rapidly expanding online fashion market, customers often face considerable difficulty in accurately assessing the fit and style of clothing through conventional e-commerce platforms. This challenge is particularly pronounced among veiled women and individuals who prioritize privacy, as current VTON solutions are either insufficient or entirely lacking in culturally appropriate options. These limitations contribute to increased dissatisfaction, elevated return rates, and a growing reliance on time-consuming in-store shopping experiences. Recognizing these challenges, our team is developing a culturally aware, intelligent VTON system that enhances fit accuracy, respects personal and cultural preferences, and delivers a more engaging and efficient online shopping experience.

The lack of localized, privacy-conscious digital fitting solutions highlights a significant technological gap in regional e-commerce infrastructure. Addressing this gap has the potential to not only boost consumer confidence but also empower online retailers to better serve diverse customer needs.

1.3 Objective

Create a Smart VTON System:
Design and implement an AI-driven platform that allows users to virtually try on clothing, offering realistic fit visualization to improve confidence in online purchases and reduce product returns.

Support Modest Fashion Through Hijab Integration:
Ensure the system accommodates the needs of veiled women by incorporating features that accurately represent hijab styles, promoting cultural inclusivity in digital fashion.

Enhance Image Processing for Realistic Results:
Tackle common visual challenges such as inconsistent lighting, pose variations, body shape diversity, and occlusions through advanced image alignment and processing techniques.

Promote Egyptian Designers and Brands:
Incorporate a localized database of fashion items from Egyptian brands to support homegrown talent and provide users with culturally relevant fashion options.

Deliver a User-Friendly and Accessible Experience:
Focus on designing an intuitive and responsive user interface that streamlines navigation, garment selection, and the VTON process for a wide range of users.

Continuously Refine Based on Real-World Use:
Evaluate the system's effectiveness through user testing, gathering feedback on fit accuracy and satisfaction to iteratively enhance performance and usability.

1.4 Time Plan

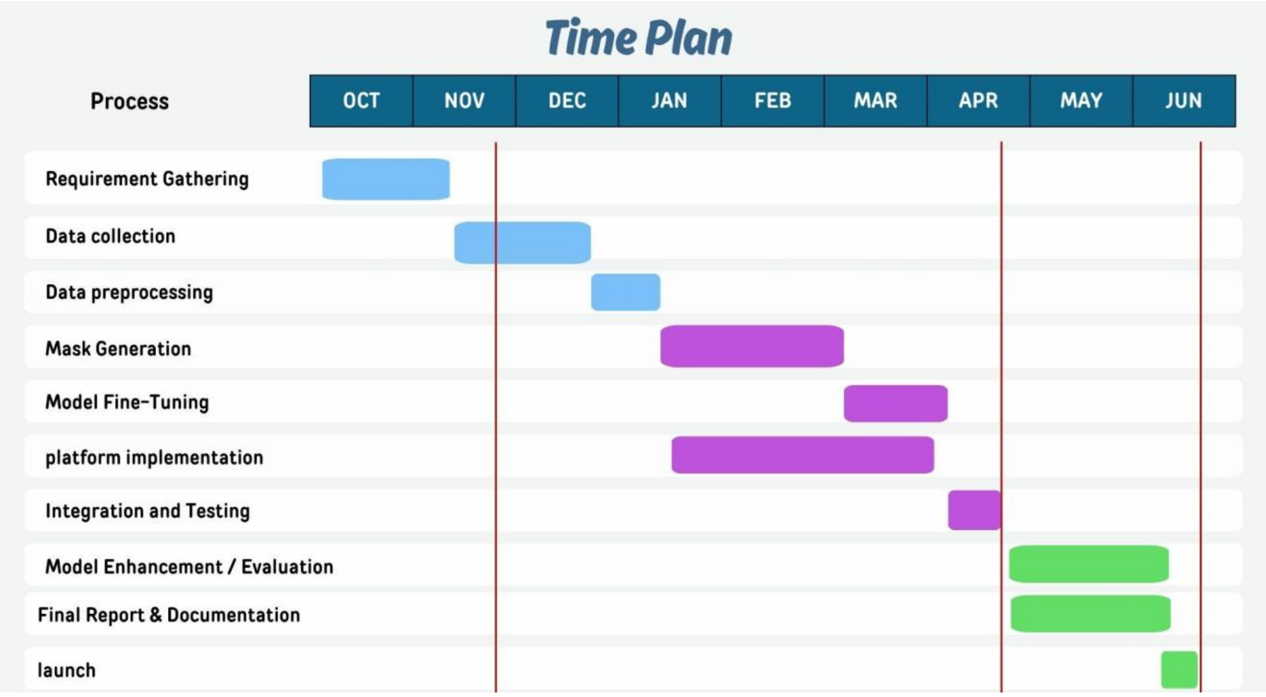


Figure 1.4.1 Time Plan

1.5 Document Organization

● **Chapter 2: Background**

This chapter presents an overview of the VTON domain, highlighting key concepts such as semantic segmentation, pose estimation, and diffusion-based generation. It also reviews existing systems like VITON-HD, discussing their strengths and limitations, particularly their lack of support for modest fashion and cultural inclusivity.

● **Chapter 3: Analysis and Design**

This chapter analyzes the problem context in detail, focusing on the specific needs of veiled users and the modest fashion market. It outlines system requirements, both functional (e.g., support for upper/lower/full outfits) and non-functional (e.g., privacy, usability), and presents the architectural design of the proposed system. It includes component diagrams, data flow, and user interaction models.

● **Chapter 4: Implementation and Testing**

This chapter explains the development process of the TryFit system. It covers the technical choices such as the use of LaDI-VTON, TPS, EMASC, and SCHP. It also details the preprocessing pipeline for segmentation and warping, and how the system integrates with Flutter and Firebase. The chapter concludes with testing methodologies, evaluation metrics (FID, LPIPS), and experimental results validating realism, cultural accuracy, and system performance.

● **Chapter 5: User Manual**

This chapter provides a step-by-step guide for end-users and providers. It explains how to upload personal images, select garments, view try-on results, and download outputs. It also includes instructions for providers on managing clothing inventory and user access through the Firebase backend.

● **Chapter 6: Conclusion and Future Work**

This chapter summarizes the project's key outcomes, including its effectiveness in generating culturally appropriate try-on results and supporting modest fashion. It also proposes future enhancements such as real-time processing, support for more cultural styles, and integration with size recommendation systems.

2- Background

1. Detailed Description of the Field of the Project: VTON

VTON is an innovative and rapidly growing field at the intersection of computer vision, deep learning, and fashion technology. It enables users to digitally try on clothing in a realistic manner using only images or videos, thereby bridging the gap between physical fitting rooms and online shopping platforms. The core aim is to allow users to visualize how garments will look and fit on their bodies without needing to physically wear them.

VTON systems utilize advanced image processing and deep learning techniques to realistically overlay or replace clothing on a person's image or video feed. This has major applications in e-commerce, enhancing customer experience, reducing return rates, and expanding accessibility to fashion.

The focus of this project is on culturally inclusive VTON systems specifically tailored to modest fashion, with special consideration for veiled women in Egypt and the MENA region. This niche addresses a significant gap in current VTON research and commercial systems, which largely cater to Western fashion and often fail to support traditional or religious dress codes such as hijabs, long dresses, and loose garments. By integrating components that respect cultural norms and privacy concerns, this project aims to democratize VTON technology for underrepresented communities.

2. Scientific Background Related to the Project

Developing realistic and culturally aware VTON systems involves combining multiple advanced computer vision and deep learning techniques:

1. **Human Parsing:**
   This refers to the semantic segmentation of human images into meaningful regions such as head, torso, arms, legs, hijab, and shoes. Accurate human parsing is critical for aligning clothing realistically and managing occlusions (e.g., ensuring a hijab overlays properly without unnatural overlaps).

2. **Pose Estimation:**
   Pose estimation algorithms detect and extract key human body joints and skeleton structure from images. This spatial information allows the system to warp garments accurately onto the user's posture, making sure the cloth fits naturally when the body is in different poses.

3. **Deep Learning Models:**

- **U-Net:** A convolutional encoder-decoder network architecture widely used for image segmentation tasks. U-Net extracts semantic masks from person images, segmenting regions needed for precise overlay.

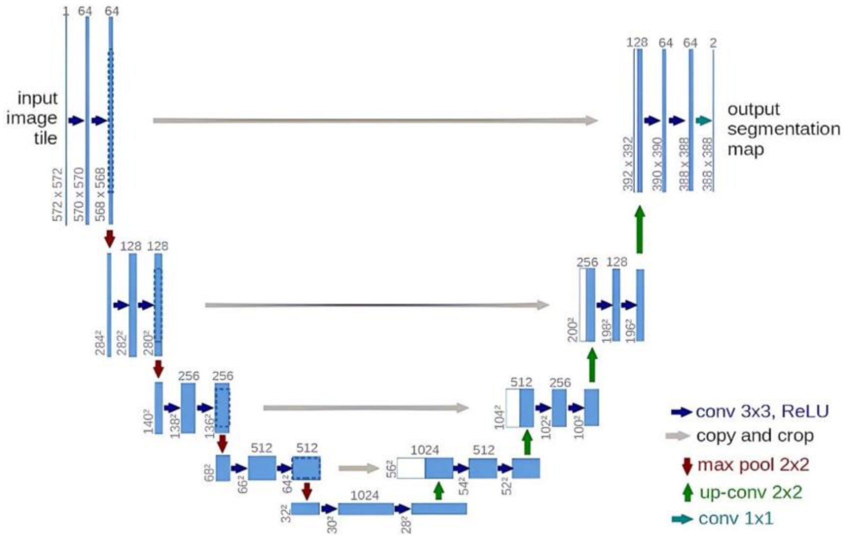


Figure 2.1 UNet architecture

- **CNN:** A convolutional neural network that extracts hierarchical visual features through localized filtering operations, used to analyze garment textures and structural patterns for accurate warping and alignment.

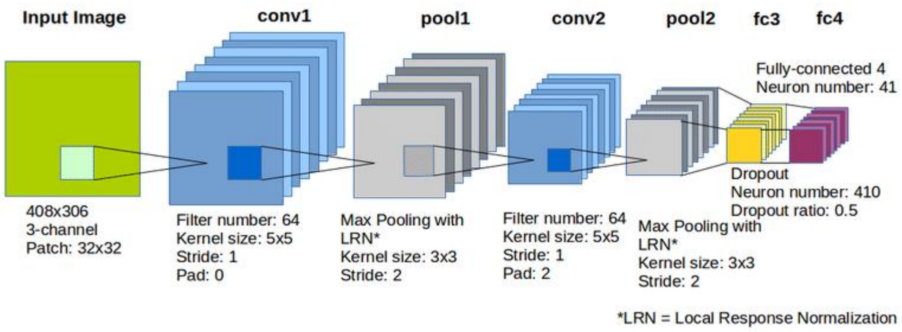


Figure 2.2 CNN architecture

- **VGG16**: A deep convolutional network with a 16-layer convolutional neural network that extracts hierarchical image features, used to compute perceptual loss for maintaining garment texture and structural fidelity during virtual try-on generation.

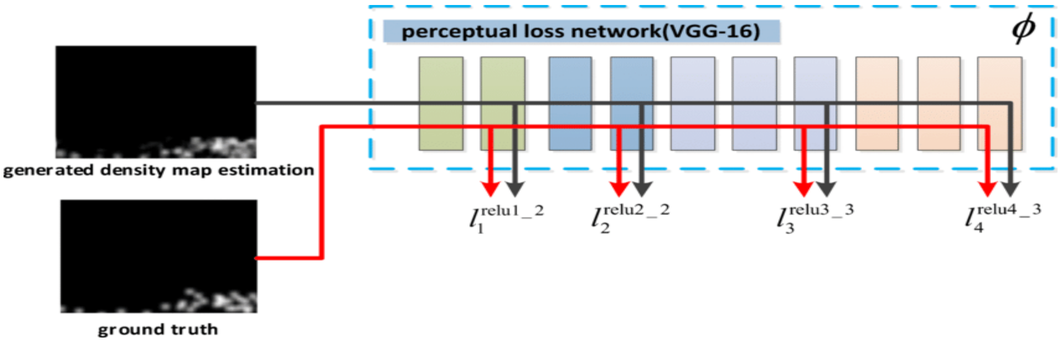


Figure 2.3 VGG16 architecture

- **VAEs:**
  VAEs are used to encode person and clothing features into a latent space and then decode them into photorealistic try-on images. They provide structured and controllable generation, allowing smoother transitions and better disentanglement of appearance attributes such as clothing texture, lighting, and body pose. Unlike GANs, VAEs emphasize reconstruction quality and stability, making them suitable for generating realistic outputs while preserving identity and modesty-aware visual consistency.

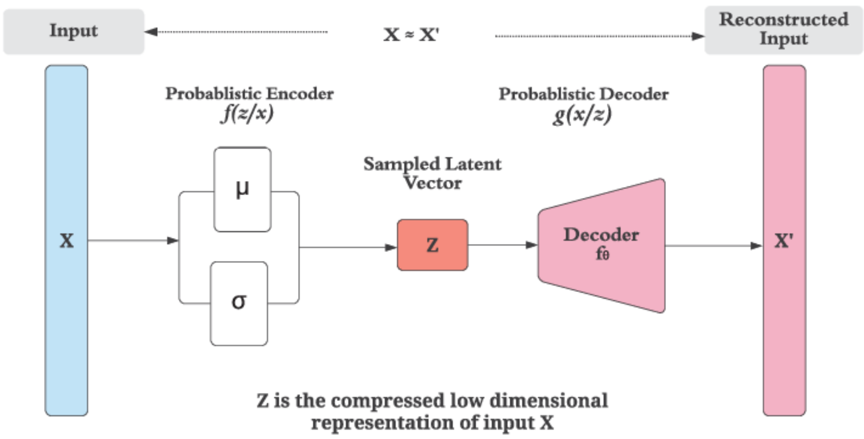


Figure 2.4 VAE architecture

- **ResNet18:**
  An 18-layer residual convolutional network that extracts hierarchical garment features, used in garment encoder to capture detailed textures and shapes for accurate warping and conditioning.

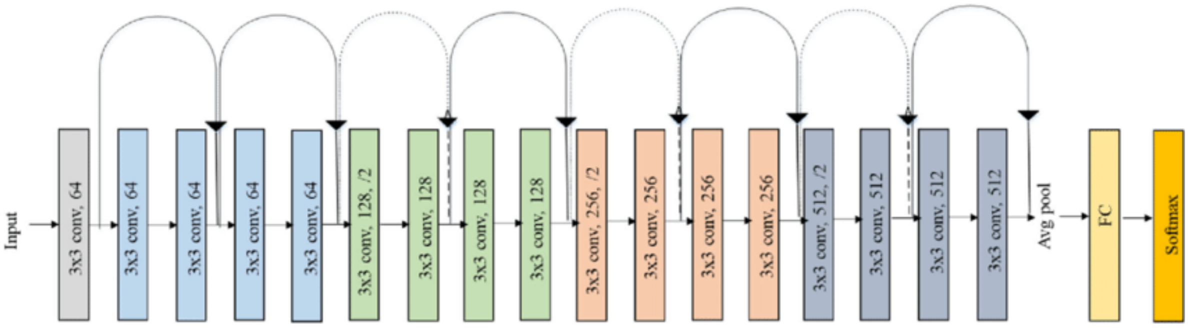


Figure 2.5 ResNet18 architecture

- **CLIP module:** A vision-language model that aligns visual garment features with textual descriptions, used to generate semantic embeddings that guide the diffusion process for style-consistent virtual try-on results.

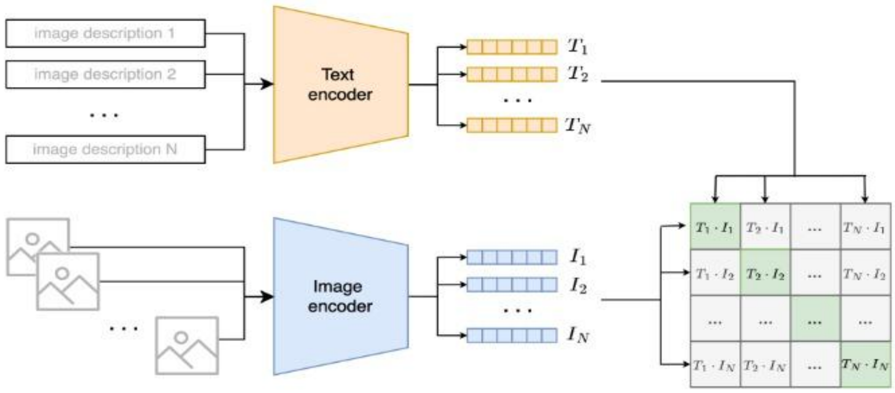


Figure 2.6 CLIP architecture

- **Refinement Networks and Attention Mechanisms:** These components improve the final output by correcting misalignments, occlusion artifacts, and lighting mismatches, producing seamless integration of garment and body.

4. **Cultural-Specific Modeling:**
   This project incorporates enhanced parsing for specific modest fashion elements such as hijab, hair, and neck regions. Such cultural-aware segmentation ensures that veiled users receive respectful and realistic try-on results that conform to modesty standards.

3. Survey of Related Work in the Field

Several pioneering research projects and commercial systems have laid the foundation for VTON technology:

- **VITON / VITON-HD[1]:**
  Early works that introduced encoder-decoder architectures with garment warping modules to transfer clothing onto personal images. They utilized thin-plate spline transformations for garment deformation but were mostly designed for Western-style clothing and lacked cultural adaptability.

- **CatVTON[2]:**
  Improved upon VITON by integrating attention-guided warping modules that better align garments with body poses and reduce distortions. However, it still lacked support for hijab or modest fashion-specific garments.

- **LaDI-VTON[3]:**
  A recent advancement that combines pose estimation methods with layered decomposition techniques for improved garment-body fusion and generalization across poses and clothing types. LaDI-VTON serves as the backend model foundation for this project, with extensions added to handle hijabs and modest fashion constraints.

Despite these advances, most existing models do not sufficiently address cultural diversity or privacy concerns inherent in modest fashion. This project extends these frameworks to fill that gap by incorporating region-specific garment types and respectful representation.

4. Description of Existing Similar Systems

In addition to academic research, several commercial platforms provide virtual try-on features:

- **Kolors Virtual Try-On (HuggingFace)[8]:**
  Offers generative try-on with clean overlays but lacks support for layered

garments or hijabs. Does not include privacy features like face masking or culturally specific segmentation.

- **VTry.io[9]:**
  Provides fast virtual try-on using user-uploaded images and pre-trained models. Limited to Western outfits, with no adaptation for modest fashion or veiled users.
- **Flux Virtual Try-On (Flux1.ai)[10]:**
  Delivers high-fidelity AI-generated try-on for modern clothing trends. Fails to represent long dresses, loose styles, or head coverings needed in MENA fashion.

These commercial systems generally do not provide nuanced modeling of modest clothing styles, do not respect privacy concerns (e.g., masking the face or hijab properly), and do not support long, loose garments or layered dressing typical in the MENA region.

This project aims to bridge these gaps by embedding cultural awareness into the model architecture and system pipeline, supporting modesty with dedicated parsing of hijabs and relevant clothing items.

5. Description of Technologies Used

The following technologies are leveraged in this project:

- **PyTorch:**
  The primary deep learning framework used to implement all neural network models including U-Net, warping networks, and refinement modules.

- **Flask:**
  A lightweight Python web framework used to build the backend API server. Flask manages user requests, runs model inference pipelines, and returns try-on results.

- **Firebase:**
  A cloud platform that stores user images, clothing items, and generated outputs. It also provides real-time database functionality and seamless integration with the frontend.

- **Flutter:**
  A low-code mobile app development platform. Flutter enables rapid creation of the mobile app frontend with real-time interactions to the Flask backend through restful API calls.

3- Analysis and Design

3.1 System Overview

3.1.1 System Architecture

Our VTON system is designed to simulate the experience of trying on clothes without physically wearing them. The system takes a full-body image of a person and a desired clothing item as input and generates a highly realistic output image of that person wearing the selected garment.

Unlike traditional virtual try-on systems, our system combines semantic understanding from natural language prompts (using learnable V⋆ tokens) and image-based understanding (via an Inversion Adapter) to guide a latent diffusion model for image synthesis. It ensures garment alignment, person identity preservation, and high visual realism.

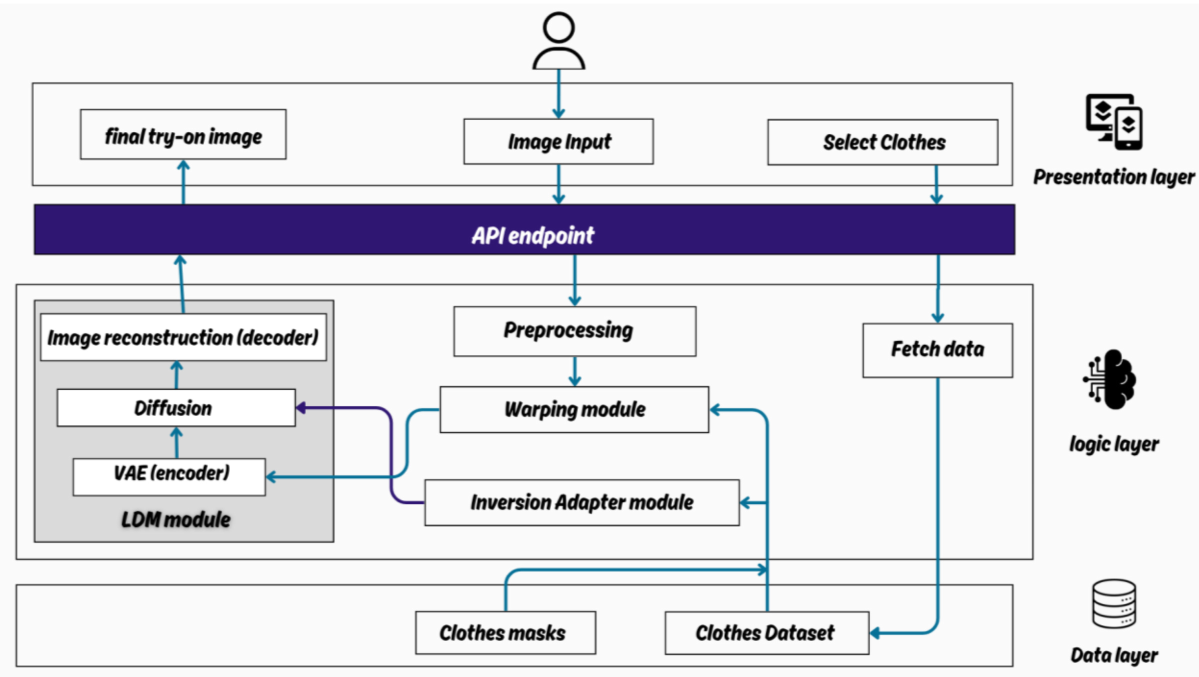


Figure 3.1.1.1 System architecture

The system is divided into three major layers:

1. **Presentation Layer** – Handles user interaction.
2. **Logic Layer** – Core processing and generation modules.
3. **Data Layer** – Contains datasets and precomputed assets used during inference.

Below is a detailed description of the system architecture and its components, based on the provided diagram.

Presentation Layer

Image Input

The user uploads a full-body photo.
This image will be used to extract body pose, segmentation maps, and keypoints for clothing alignment.

Select Clothes

User browses and selects a garment (e.g., shirt, dress) from the database.
The selected item is passed into the logic layer for try-on.

Logic Layer

Fetch Data

Retrieves:
The selected garment image
Its corresponding clothing mask (binary/segmented region of the clothing)
Other metadata if needed (e.g., category tags or textual labels for conditioning)

Preprocessing

Extracts structural information from the input human image:
Pose keypoints / skeletons using a model like OpenPose
Segmentation maps for preserving body parts like the face, arms, ..etc.
Outputs are passed to the Warping and Inversion Adapter modules.

Warping Module

Takes:
The garment image
The pose/skeleton/keypoints of the target human

Outputs a warped garment, aligning the clothing shape and position to the user's body pose using geometric transformations (e.g., TPS or flow-based warping).

Inversion Adapter Module
Encodes the garment appearance (e.g., texture, color, style) into pseudo-token embeddings.
These tokens represent the semantic and visual information of the garment for conditioning the diffusion model.

VAE (Encoder)
Compresses the input human image and warped garment into a latent representation.
VAE ensures lower-dimensional efficiency and smooth latent space transitions for downstream diffusion.

Diffusion
Refines the latent representation by incorporating:
Warped clothing
Pseudo-token embeddings
Pose and segmentation maps
Denoises and enhances alignment and realism during multi-step sampling.

Image Reconstruction (Decoder)
Decodes the refined latent from the diffusion module back into a full-resolution image.
Output is a realistic try-on image showing the user wearing the selected garment.

Data Layer

Clothes Dataset
A curated dataset of clothing images (tops, dresses, pants, etc.), optionally categorized.
Includes metadata and labels for garment type and attributes.

Clothes Masks
Binary or segmentation masks that isolate clothing from the background.
Used in warping and embedding to extract clothing regions accurately.

3.1.2 System Users

A. Intended Users:

The VTON system is designed to serve two primary categories of users: **Normal Users** and **Providers**, each interacting with the system in different ways based on their roles and permissions.

1. Normal Users

These are the end users who utilize the application to virtually try on clothing items. Their primary interactions with the system include:

- **Browsing and selecting clothes** from various categories.
- **Uploading personal photos** to preview how selected clothing items would appear on them using the try-on feature.
- **Viewing generated results** after processing try-on sessions and saving them.
- **Adding clothing items to favorites** for quick access in future visits.
- **Managing their personal profiles**, including updating their names, changing their passwords, and modifying their profile image.

This group benefits from the system's AI-driven try-on capabilities, which aim to enhance the online shopping experience by providing a realistic visual preview of clothing items.

2. Providers

Providers are administrative users, such as brand owners or store managers, who are responsible for managing the clothing inventory within the application. Their main functionalities include:

- **Adding new clothing items** to the system under appropriate categories.
- **Removing outdated or unavailable items** from the database to ensure the catalog remains current and accurate.

By managing the clothing dataset, providers ensure the system offers a diverse and up-to-date selection of fashion items for normal users to interact with.

B. User Characteristics

The VTON system is designed to be user-friendly and accessible, requiring minimal technical expertise from its users. The two main user groups: **Normal Users** and **Providers** have slightly different expectations in terms of skill level and interaction with the system.

1. Normal Users

Normal users are general consumers who want to try on clothes virtually. No prior technical knowledge is required. However, to operate the system effectively, users should possess the following basic skills:

- **Familiarity with mobile applications**, such as navigating screens, clicking buttons, and uploading images.
- **Basic familiarity with VTON interfaces**, including browsing clothing categories, selecting items to try on, uploading images, and managing user profiles.
- **Ability to register and log in** using an email and password.

The system is designed with an intuitive user interface to support a wide range of age groups and technical backgrounds.

2. Providers

Providers are administrative users responsible for managing the clothing inventory. While the system remains user-friendly, providers are expected to have:

- **Basic digital literacy**, including file management and image handling.

- **Understanding of product categorization**, allowing them to properly assign clothing items to relevant categories.
- **Access control awareness**, ensuring secure and appropriate use of administrative privileges (e.g., adding or removing items).

Providers may receive minimal training or documentation to support these tasks, but no advanced technical background is required.

3.2 Design Diagrams

3.2.1 Use Case Diagram

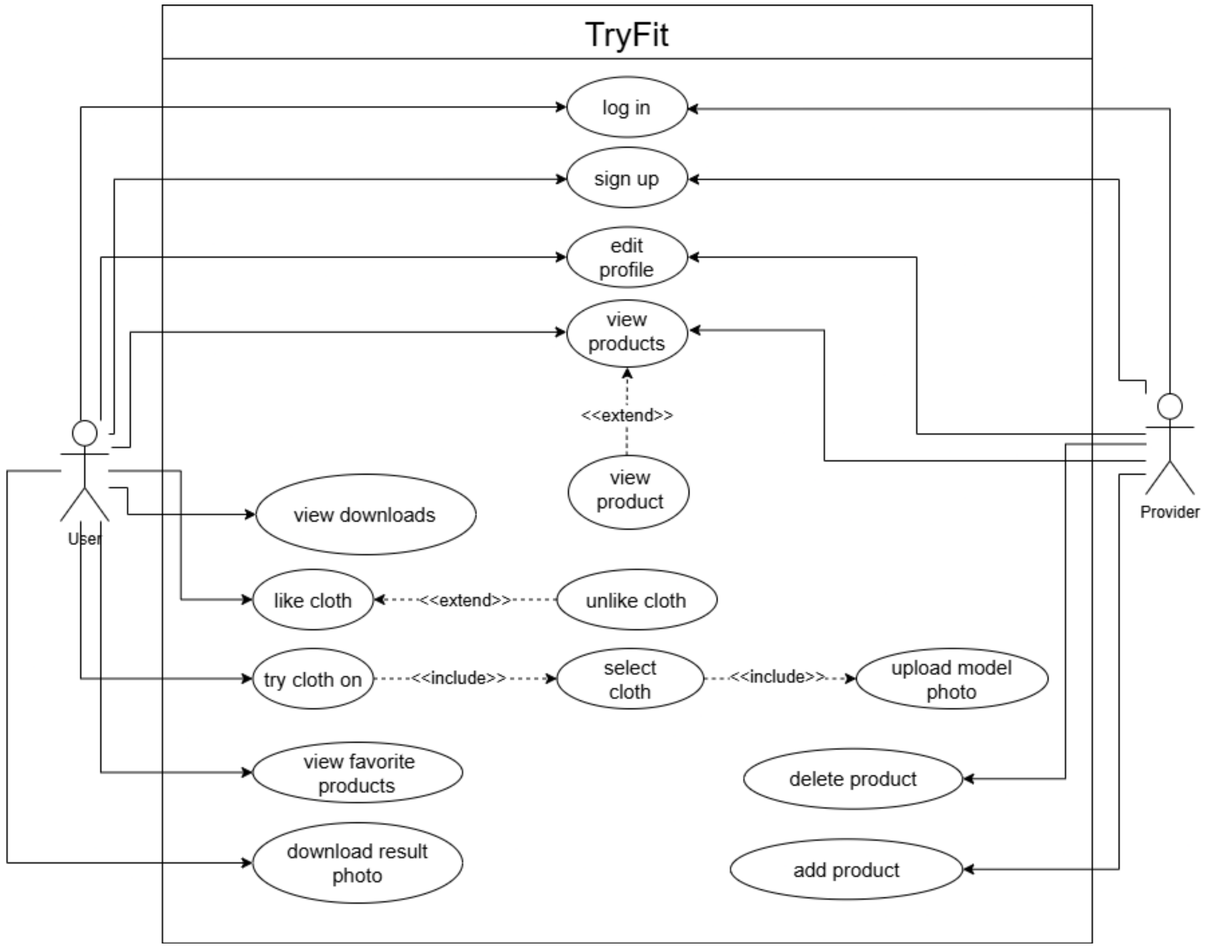


Figure 3.2.1.1 Use Case Diagram

Use Case Description

Actor 1: User (Normal User)

Users are individuals who use the TryFit app to visualize how clothes look on them virtually. Their functionalities include:

- **Sign Up / Log In:**
  Users can register for an account and log in to access the application features.

- **Edit Profile:**
  Users can update their personal details such as name, password, and profile picture.

- **View Products:**
  Displays all clothing items available in the system.

- **View Product** (Extended from View Products)**:**
  Shows a larger view of the selected clothing item in a dedicated screen, including only the image for better visual clarity.

- **Like Cloth / Unlike Cloth:**
  Allows users to favorite or unfavorite clothing items.

- **View Favorite Products:**
  Displays all items that the user has marked as favorite.

- **Try Cloth On** (Includes Select Cloth and Upload Model Photo)**:**
  Enables users to try on a selected item by uploading their model image. The selected clothing image is then applied to the user's image using the virtual try-on system.

- **Download Result Photo:**
  Users can download the final generated image resulting from the try-on process.

- **View Downloads:**
  Users can review and access previously downloaded results.

Actor 2: Provider

Providers are administrative users responsible for managing clothing content in the application. Their functionalities include:

- **Sign Up / Log In:**
  Authenticate to access provider-specific controls.
- **Edit Profile:**
  Update the provider's personal information and profile image.
- **Add Product:**
  Allows uploading a new clothing item to the system, including its image and category.
- **Delete Product:**
  Remove outdated or unwanted clothing items from the app.
- **View Products:**
  Shows the entire catalog of clothing items, similar to the user view.
- **View Product** (Extended from View Products)**:**
  Displays a larger image of a selected clothing item in a dedicated view-only screen, allowing providers to review item visuals clearly.

3.2.2 Class Diagram

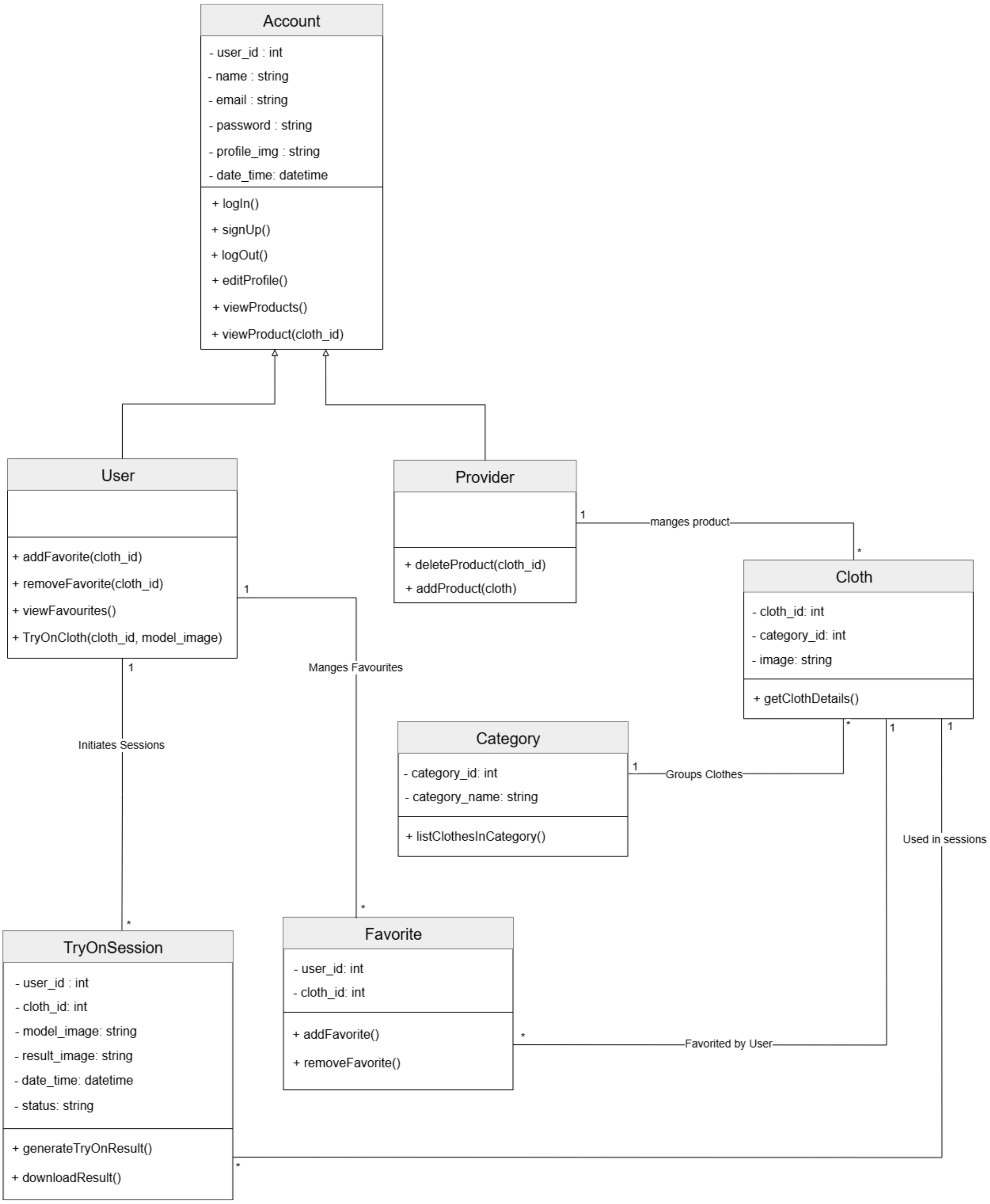


Figure 3.2.2.1 Class Diagram

Class Description

1. Account (Abstract Base Class)

This class defines shared attributes and methods between **User** and **Provider.**

- **Attributes:**
  - **user_id**: int
  - **name**: string
  - **email**: string
  - **password**: string
  - **profile_img**: string
  - **date_time**: datetime
- **Methods:**
  - login()
  - signUp()
  - logOut()
  - editProfile()
  - viewProducts()
  - viewProduct(cloth_id)

2. User (Inherits from Account)

Represents a regular user who can try on clothes and manage favorites.

- **Methods:**
  - addFavorite(cloth_id)
  - removeFavorite(cloth_id)
  - viewFavourites()
  - TryOnCloth(cloth_id, model_image)

3. Provider (Inherits from Account)

Represents the clothing provider who can manage the product catalog.

- **Methods:**
  - addProduct(cloth)
  - deleteProduct(cloth_id)

4. Cloth

Represents a clothing item.

- **Attributes:**
  - **cloth_id**: int
  - **category_id**: int
  - **image**: string
- **Methods:**
  - getClothDetails()
- **Relationships:**
  - Managed by a **Provider**
  - Belongs to a **Category**
  - Used in **TryOnSession**
  - Can be favorited by many **Users**

5. Category

Defines types of clothing such as "Upper", "Lower", and "Full".

- **Attributes:**
  - **category_id**: int
  - **category_name**: string
- **Methods:**
  - listClothesInCategory()

6. TryOnSession

Represents a virtual try-on session for a specific user and cloth.

- **Attributes:**
  - **user_id**: int
  - **cloth_id**: int
  - **model_image**: string
  - **result_image**: string
  - **date_time**: datetime
  - **status**: string
- **Methods:**
  - generateTryOnResult()
  - downloadResult()

7. Favorite

Maps the many-to-many relationship between **User** and **Cloth**.

- **Attributes:**
  - **user_id**: int
  - **cloth_id**: int
- **Methods:**
  - addFavorite()
  - removeFavorite()

3.2.3 Sequence Diagram

A) user open application and try clothes

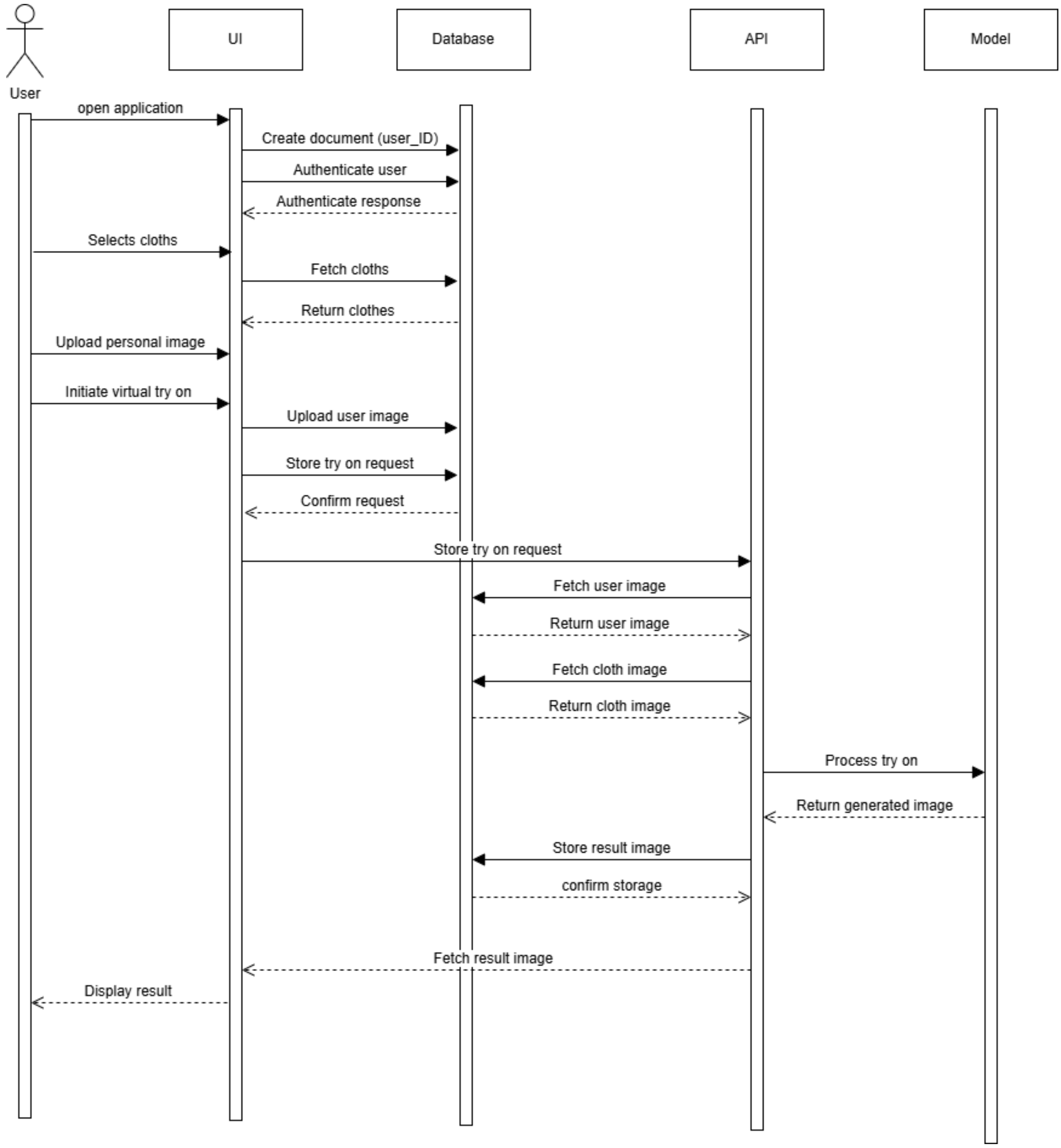


Figure 3.2.3.1 Sequence Diagram for open app and try on

B) user add favorite cloth image

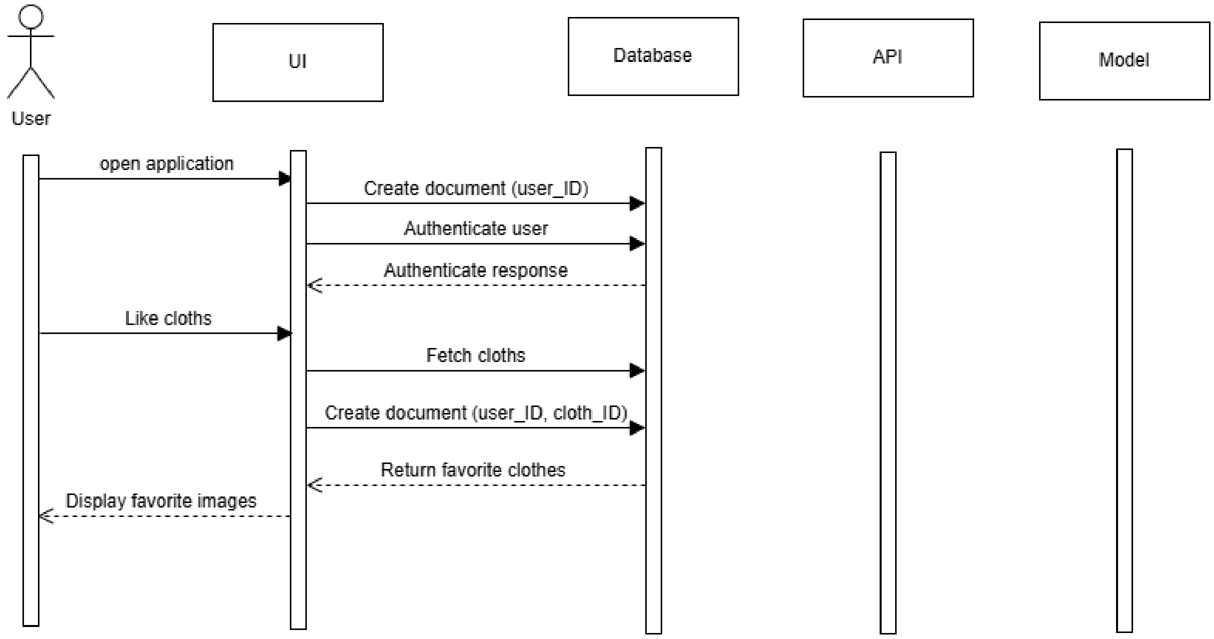


Figure 3.2.3.2 Sequence Diagram for add image to favorite

C) provider add and remove items

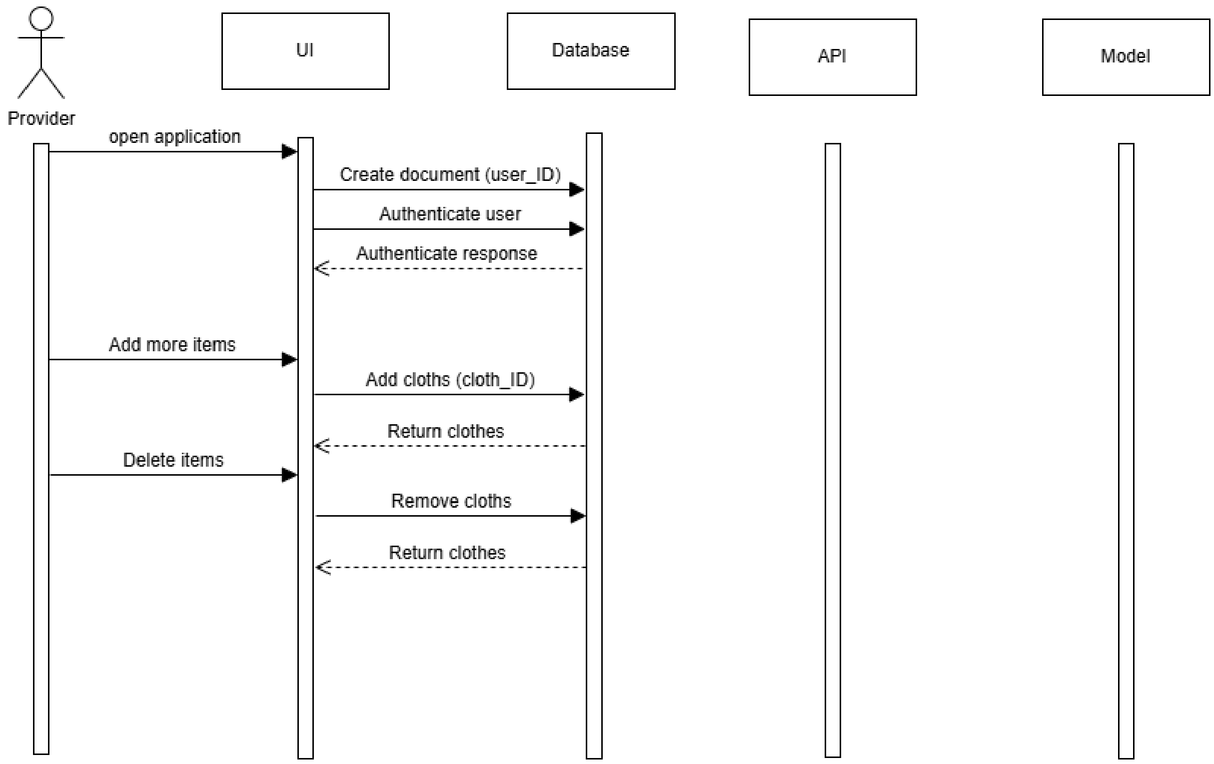


Figure 3.2.3.3 Sequence Diagram provider add and remove item

3.2.4 Database Diagram

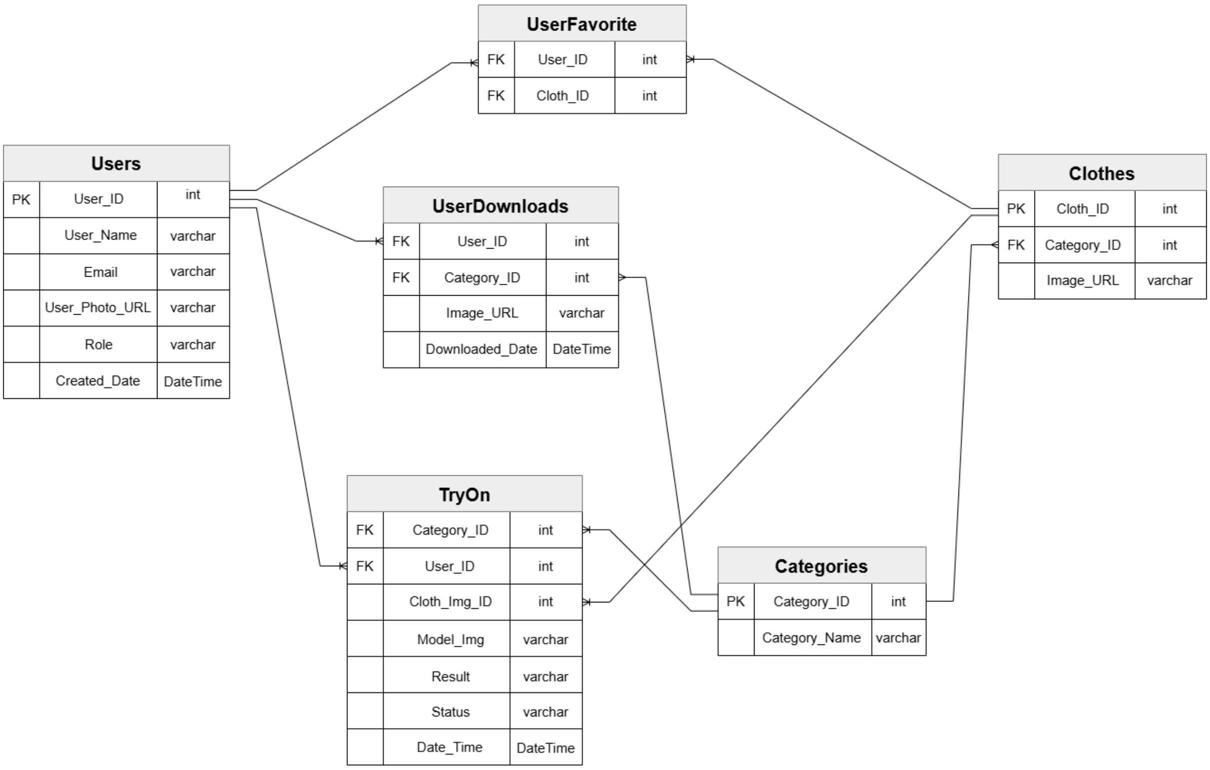


Figure 3.2.4.1 Database diagram

Table Descriptions

- **Users**
    - **User_ID** (PK): Unique identifier for each user.
    - **User_Name**: Name of the user.
    - **Email**: User's email address.
    - **User_Photo_URL**: Link to the user’s profile image.
    - **Role**: Defines user type (e.g., normal user or provider).
    - **Created_Date**: Timestamp of user account creation.

- **Clothes**
  - **Cloth_ID** (PK): Unique identifier for each clothing item.
  - **Category_ID** (FK): Links the item to a specific category.
  - **Image_URL**: Image path for the clothing item.

- **Categories**
  - **Category_ID** (PK): Unique identifier for each clothing category.
  - **Category_Name**: The name of the category (e.g., Upper, Lower, Full).

- **UserFavorite**
  - **User_ID** (FK): References the user who favorited the item.
  - **Cloth_ID** (FK): References the favorited clothing item.
  - This table implements a **many-to-many** relationship between users and clothes.

- **TryOn**
  - **Category_ID** (FK): Category of the selected clothing.
  - **User_ID** (FK): References the user performing the try-on.
  - **Cloth_Img_ID**: ID of the clothing image used.
  - **Model_Img**: Path or reference to the uploaded user photo.
  - **Result**: URL or path of the generated try-on output.
  - **Status**: Processing status (e.g., Done, Not_Done).
  - **Date_Time**: Timestamp of the try-on action.

- **UserDownloads**
  - **User_ID** (FK): References the user who downloaded the result.
  - **Category_ID** (FK): Category of the downloaded item.
  - **Image_URL**: Path to the downloaded result image.
  - **Downloaded_Date**: Timestamp of the download action.

4- Implementation and Testing

4.1 Dataset Description

The dataset used in the TryFit VTON system was compiled through a combination of **web scraping techniques** and the integration of **publicly available datasets**, specifically the **DressCode** and **ANCHOR** datasets. These sources were selected for their relevance and quality in the context of clothing try-on applications.

The data is structured in the form of **paired images**, where each pair includes:

- A standalone image of the clothing item.
- A corresponding image of a model wearing the same item.

This pairing approach ensures the system can learn both garment structure and fit, which is essential for realistic VTON performance.

To ensure diversity and inclusivity, the dataset includes both **veiled** and **unveiled** models, which helps address the cultural and market-specific needs, especially for local fashion consumers.

The clothing items are categorized into **three main categories**:

- **Upper-body garments** (e.g., shirts, blouses)
- **Lower-body garments** (e.g., trousers, skirts)
- **Full-body outfits** (e.g., dresses, abayas)

Dataset

Table 4.1.1 Dataset description

<table>
<tr><th>Name</th><th>Size</th><th>Description</th><th>Resource</th></tr>
<tr><td rowspan="4">Our Dataset</td><td>399 pair images (FULL)
523 pair images (LOWER)
663 pair images (UPPER)</td><td>unveiled models</td><td>DressCode Dataset</td></tr>
<tr><td>1000 pair images (FULL)
865 pair images (UPPER)</td><td>unveiled models</td><td>Anchor Dataset</td></tr>
<tr><td>41 pair images (FULL)
93 pair images (LOWER)
55 pair images (UPPER)</td><td>unveiled models</td><td rowspan="2">web scraping Local brands websites</td></tr>
<tr><td>141 pair images</td><td>veiled models</td></tr>
</table>

Samples of dataset

Dresses

Figure 4.1.1 sample of dresses dataset

Lower body

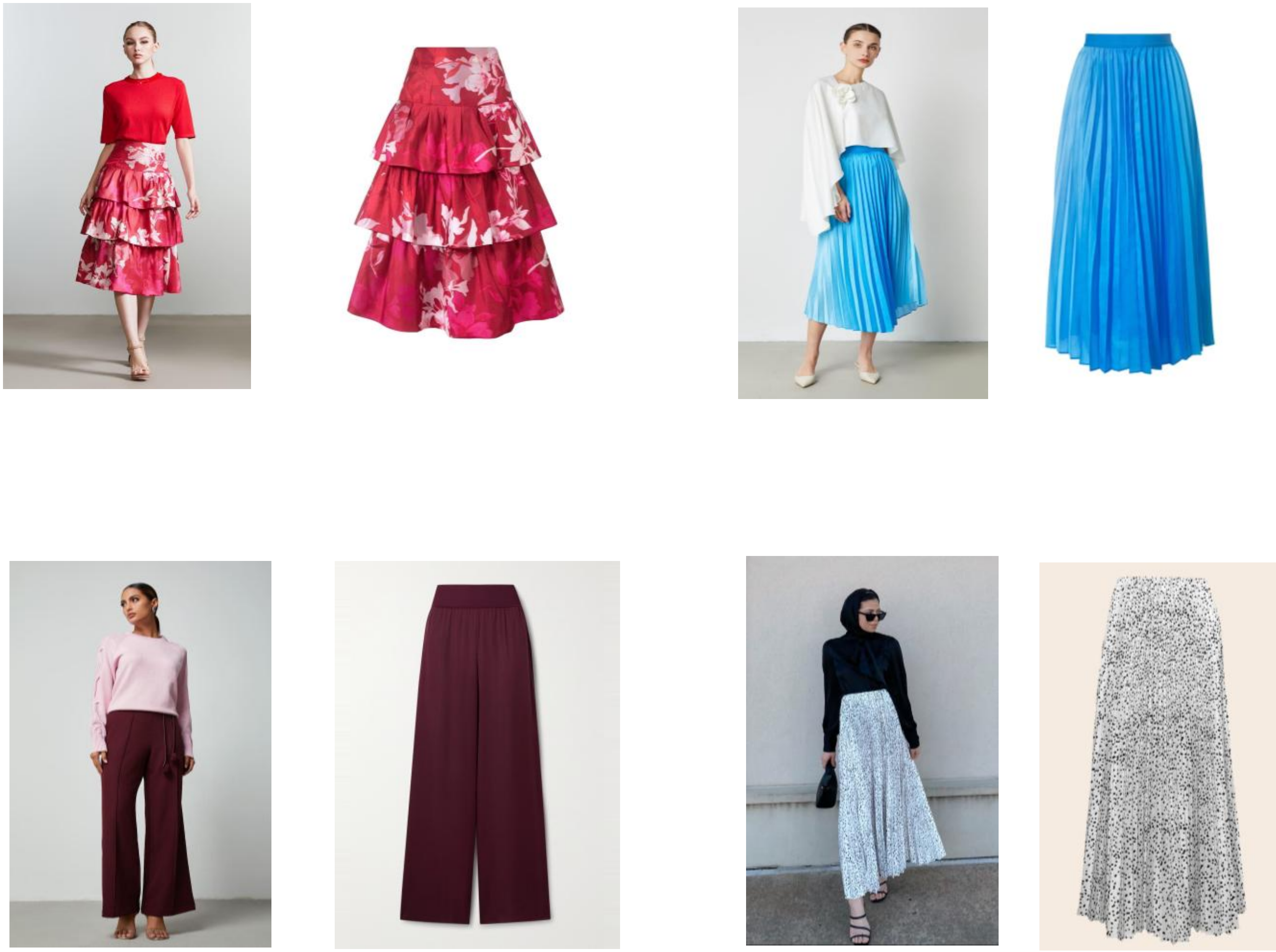


Figure 4.1.2 sample of Lower-body dataset

Upper body

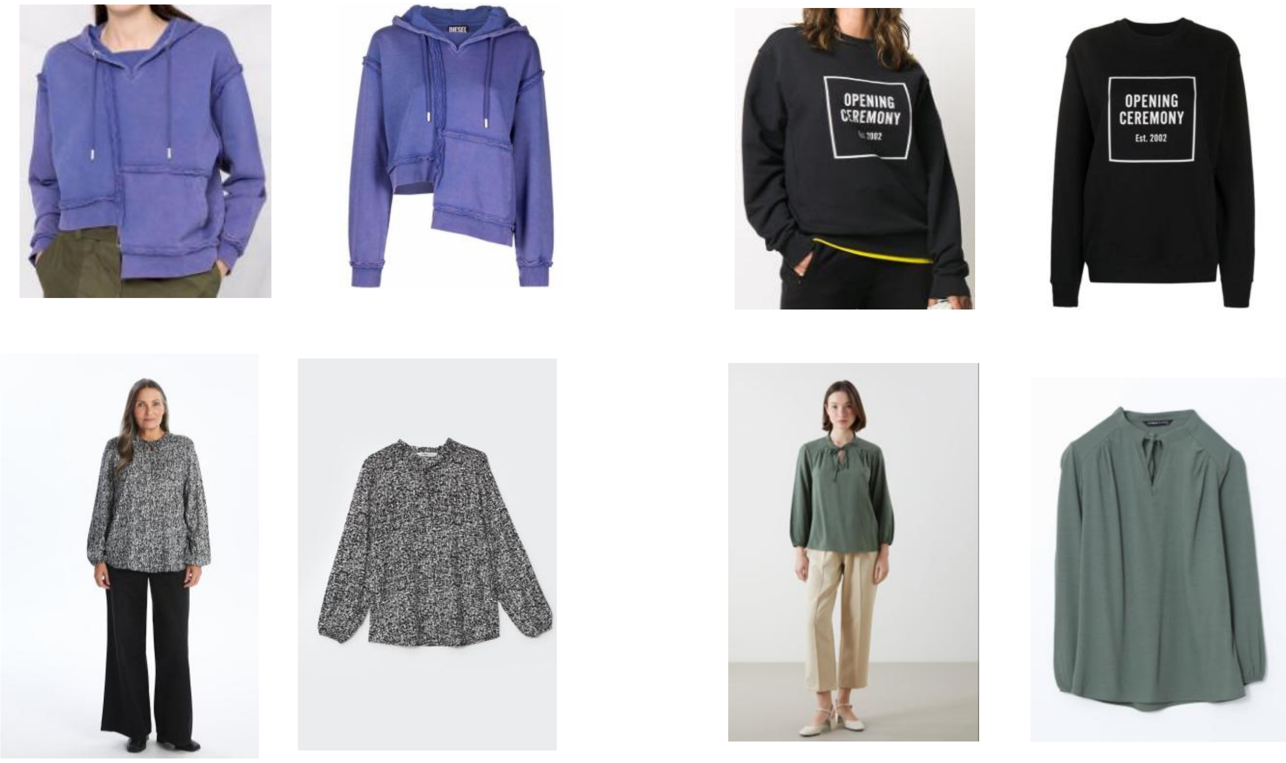


Figure 4.1.3 sample of Upper-body dataset

4.2 Phases Description

Phase 1: Data Preparation & Preprocessing

This phase extracts essential visual and geometric features from the person and garment images to guide the try-on process.

1.1 Human Parsing (Semantic Segmentation)

A pretrained human parsing model (SCHP) segments the person image into approximately 18 body regions (e.g., face, arms, hair, upper clothes, etc.). These segmentation maps enable:

- Identity preservation (by isolating face, hair, and skin)
- Region-wise garment fitting and inpainting
- Parsing-aware skip connections during image decoding

Figure 4.2.1 model label map

1.2 Pose Estimation (Keypoint and skeleton Detection)

A pose estimation model with OpenPose.
Detects 2D body keypoints (e.g., shoulders, knees) and skeleton.

These keypoints are used to:

- Guide garment alignment
- Control the GMM
- Maintain structural consistency in try-on results

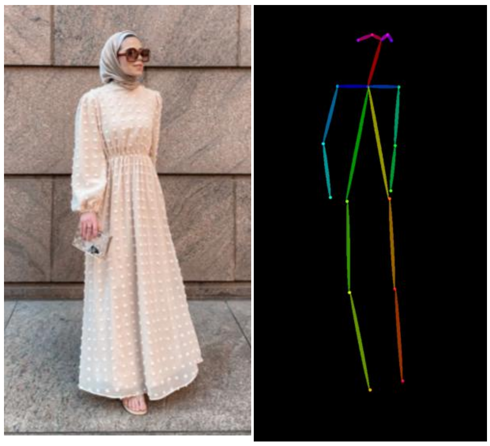


Figure 4.2.2 model skeleton

1.3 Cloth Mask Extraction

A binary segmentation mask of the selected in-shop garment is generated manually or with a segmentation model (e.g., Grounded-SAM). This enables:

- Focused warping on the garment only
- Clear separation between foreground (clothing) and background

Figure 4.2.3 cloth mask

1.4 Face Masking (Privacy and Attention Control)

To avoid privacy issues and enhance training, the face is masked out in the person image using the parsing map. This helps:

- Prevent facial feature memorization during training
- Redirect attention toward garment realism and alignment

Phase 2: LaDi-VTON Try-On Pipeline

This phase includes semantic garment encoding, geometric warping, alignment refinement, and latent encoding.

2.1 Warping Module: Geometric Alignment & Refinement

This module aligns the in-shop garment with the target person's pose and body shape through three key stages:

2.1.1 Feature Extraction

Input Components:

- In-shop garment image (RGB)
- Target person's 2D pose keypoints (from OpenPose)
- Garment segmentation mask (binary)

Feature Processing:

1. A CNN encoder (ResNet-based) extracts multi-level visual features from both:
   - The garment image
   - The target person image
2. Pose heatmaps are generated from keypoints and combined with visual features

2.1.2 Thin Plate Spline (TPS) Warping

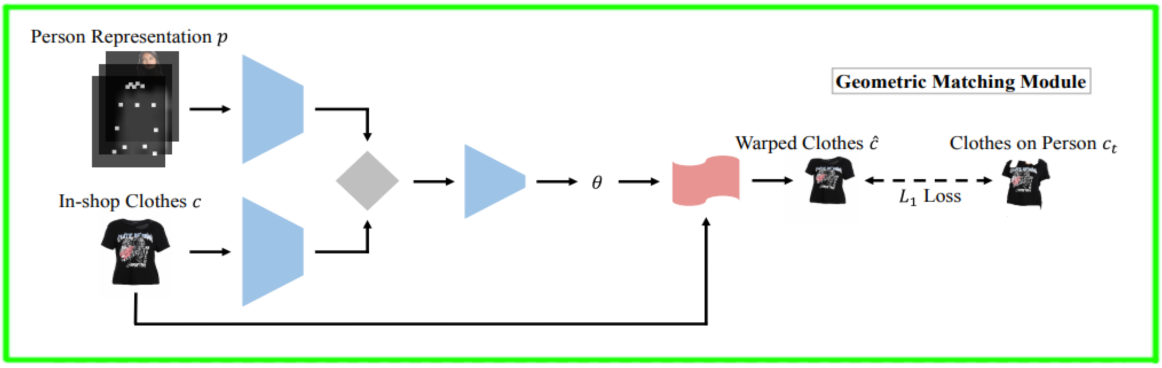


Figure 4.2.4 TPS architecture

Core Function:

- Predicts smooth deformation fields to non-rigidly warp the garment
- Aligns garment shape with target body pose while preserving texture

Key Mechanisms:

1. Control point detection along garment edges
2. Optimal transformation calculation that:
   - Minimizes spatial distortion
   - Maintains fabric integrity
   - Balances precise alignment with natural deformation
3. Implementation via a Geometric Matching Module (GMM) that predicts TPS parameters

Output:

- Coarsely aligned garment image matching the target pose

2.1.3 Refinement Network (U-Net Architecture)

Input Fusion:

- Warped garment image
- Parsed person segmentation (from SCHP)
- Pose heatmap representation

Network Design:

- Encoder-Decoder structure with skip connections
- 4-level downsampling/upsampling blocks
- Each block contains:
    - Convolutional layers
    - Batch normalization
    - ReLU activation

Training Objectives:

1. Pixel-level accuracy (L1 loss)
2. Visual quality preservation (Perceptual loss using VGG-19)
3. Combined loss function balancing both objectives

Key Benefits:

- Corrects residual misalignments
- Enhances garment details
- Maintains realistic fabric appearance
- Preserves important structural elements

2.2 CLIP and Inversion Adapter Module

2.2.1 Module Overview

The CLIP and Inversion Adapter module forms the **cross-modal translation core** of the virtual try-on system, performing bidirectional alignment between:

1. **Visual Domain** (Garment images → CLIP ViT features)
2. **Textual Domain** (Fashion descriptors → CLIP text embeddings)

This enables **semantic-aware garment conditioning** for the diffusion process through:

- Hierarchical visual feature extraction
- Learned projection to text-compatible latent space
- Multi-modal fusion for enhanced detail preservation

2.2.2 Dual-Stream Processing Pipeline

A- Text Processing Branch

Inputs:

- Natural language prompts (e.g., "a striped sleeveless summer dress")
- Category templates ("an upper body garment")

Processing Stages:

1. **Tokenization**:
   - Max sequence length: 77 tokens
   - Special tokens: [SOS], [EOS], [PAD]
   - Token IDs mapped via CLIP's vocabulary (49,408 entries)
2. **Text Encoding**:
   - 12-layer transformer (768-D embeddings)
   - Positional embeddings added
   - Output: text_embeddings (77, 768)

Key Operations:

- Attention masking for padding tokens
- Layer-wise gradient scaling
- Prompt weighting (emphasis on garment descriptors)

B- Visual Processing Branch

Inputs:

- Garment image (3, H, W) → resized/cropped to 224×224
- Optional: Detail preservation masks

Processing Stages:

1. **Patch Embedding**:
   - 14×14 patches → 196 sequences
   - Linear projection to 1280-D
   - [CLS] token initialization
2. **ViT Encoding** (32 layers):
   - Multi-head attention (16 heads)
   - LayerNorm → MLP (GELU, 4:1 expansion)
   - Output: [CLS] (1,1280) + patch_tokens (196,1280)
3. **Feature Aggregation**:
   - Multi-layer fusion (last 4 layers)
   - Spatial attention pooling
4. **output**

   CLIP visual features ([CLS] token + patch tokens)

2.2.3 Cross-Modal Feature Fusion System

A- Inversion Projection

Purpose: Transforms visual garment features into text-compatible pseudo-tokens

Process:

1. **Input:** CLIP visual features ([CLS] token + patch tokens)
2. **Architecture:**
   - 3-layer MLP with bottleneck design
   - Dimensionality: 1280 → 1024 → 768
3. **Output:**
   - 16 pseudo-tokens (16 × 768)
   - Each token encodes distinct garment attributes (texture, color, structure)

Key Innovation:

- Maintains spatial-aware features while aligning with text embedding space

B- Feature Fusion

1. **Early Fusion** (Adapter Input):
   - Concatenate [CLS] + spatial features
   - Weighted by texture saliency
2. **Late Fusion** (Diffusion UNet):
   - Visual tokens attend to text queries
   - Dynamic gating (0.3 text: 0.7 visual)

C- Conditional Generation

- Visual tokens control:
  - Texture details (stripes, prints)
  - Color distribution
- Text embeddings guide:
  - Garment category
  - Style semantics

2.3 LDM module

2.3.1 Variational Autoencoder (VAE)

Purpose:
The VAE serves as a latent space projector, efficiently compressing high-resolution images into a compact representation while preserving essential visual details.

Architecture Details:

- Encoder:
  - Comprises four downsampling blocks, each consisting of 3×3 convolutions, instance normalization, and SiLU activation functions.
  - Outputs a latent tensor, representing mean (μ) and log-variance (σ) for stochastic sampling.
  - Achieves a 16× compression ratio (1024×768×3 → 64×48×4).
- Decoder:
  - Reconstructs images via four upsampling blocks using nearest-neighbor interpolation and 3×3 convolutions.
  - Incorporates skip connections from the encoder to enhance detail recovery.

Key Innovations:

- **Localized feature learning** enhances preservation of fine garment details like textures and patterns.
- **Perceptual compression** optimizes image quality for human vision while maintaining computational efficiency.

2.3.2 Diffusion Process

Noise Schedule:

- Utilizes a 1000-step cosine schedule for noise injection.
- Balances noise ratios.
- Incorporates learned variance prediction for adaptive denoising.

Denoising Dynamics:

- **Forward Process:**
  Progressively corrupts the latent representation with Gaussian noise, governed by a decay factor.
- **Reverse Process**:
  Recovers the clean latent state through iterative noise prediction, conditioned on time-dependent parameters.

2.3.3 Denoising U-Net

- Encoder-Decoder Structure:
  - Four hierarchical levels with channel dimensions [128, 256, 512, 1024].
  - Residual blocks feature group normalization.
- Multi-Scale Self & Cross-Attention
  - Combines 4-head self-attention for spatial feature relationships with cross-attention that aligns garment image features (512D) and text embeddings (768D)
  - Processes features hierarchically across different resolutions (32×24 to 64×48)
- Position-Guided Attention
  - Specifically incorporates body pose information through heatmap-conditioned attention in early network stages
  - Uses DensePose coordinates to spatially guide feature attention without full axial attention in decoder layers

2.3.4 Enhanced Mask-Aware Skip Connections (EMASC)

Architecture:

- Processes parsing masks (18 channels) into 64-channel feature maps via dilated convolutions.
- Dynamically modulates U-Net skip connections using sigmoid-activated gating.
- Injects residual mask features to reinforce spatial alignment.

Functional Properties:

- Applies different resolutions for multi-scale refinement.
- Specifically enhances:
    - Garment-boundary continuity
    - Fabric pattern retention
    - Occlusion handling (e.g., sleeve-body interactions)

2.3.5 Conditioning Framework

Textual Guidance:

- Leverages a CLIP text encoder (ViT-H/14) for 1280D embeddings.
- Employs four trainable V* tokens (1280D) with position-sensitive embedding.
- Projects text features to 768D via an adapter network.

Visual Guidance:

- Extracts 512D garment features using a modified ResNet-18.
- Projects features to 768D for cross-attention compatibility.

Pose Guidance:

- Encodes 2D keypoints into 18-channel heatmaps.
- Applies learned positional encoding for spatial conditioning.

2.3.6 Latent Decoding (Final Output)

Decoding:

After denoising, the final latent $\hat{z}$ is passed through the pretrained VAE decoder to generate the final RGB try-on image. The result is:

- High-resolution (e.g., 1024×768 px)
- Aligned with the person's body and pose
- Faithful to the selected garment's shape and texture

4.3 Technologies & UI

Technologies Used

1. **Machine Learning/Deep Learning:**

    a. Human Parsing: SCHP style segmentation models.

    b. Pose Estimation: OpenPose or equivalent keypoint extractors.

    c. Warping: TPS transformation.

    d. Try-On: LADI-VTON[3] (VAE + LDM + EMASC).

2. **Image Processing:**

    a. Cloth mask extraction using Segment Anything Model.

    b. Face blurring via segmentation masks.

3. **Backend & API Integration:**

    a. Flask-based API endpoints to process model and cloth images.

    b. Firebase/Flutter (future work) for cloud and mobile integration.

UI Design and Wireframes

While detailed wireframes are not included in the presentation, the system includes a front-end interface that allows:

- Image upload (person and clothing).
- Try-on preview generation.
- Interaction flow to choose between different clothing items.

4.4 Experimental Results

Dataset 1: 706 pairs of images / 121 veiled pairs

Dataset 2: 586 pairs of images / 10 veiled pairs

Dataset 3: 1274 pairs of images / 130 veiled pairs

Dataset 4: 1912 pairs of images

Table 4.4.1 Experiments and trails results

#	Dataset	EMASC steps	TPS epochs	refinement epochs	Text	descriptions	Hyper parameter tuning	FID ↓	LPIPS↓
1	dataset 1	2000	5	5	_	_	_	111.35	0.24
2	dataset 1	3000	5	5	_	_	_	78.49	0.147
3	dataset 2	3000	30	30	_	_	_	103.22	0.147
4	dataset 3	5000	50	50	✓	_	_	53.577	0.142
5	dataset 3	7000	50	50	✓	_	✓	**53.283**	**0.140**
6	dataset 4	7000	75	75	✓	✓	✓	57.219	0.168
7	dataset 4	7000	70	70	✓	✓	✓	59.249	0.169
8	dataset 4	7000	70	70	✓	_	✓	57.62	0.168

Visual results

Figure 4.4.1 Qualitative results on our dresses dataset

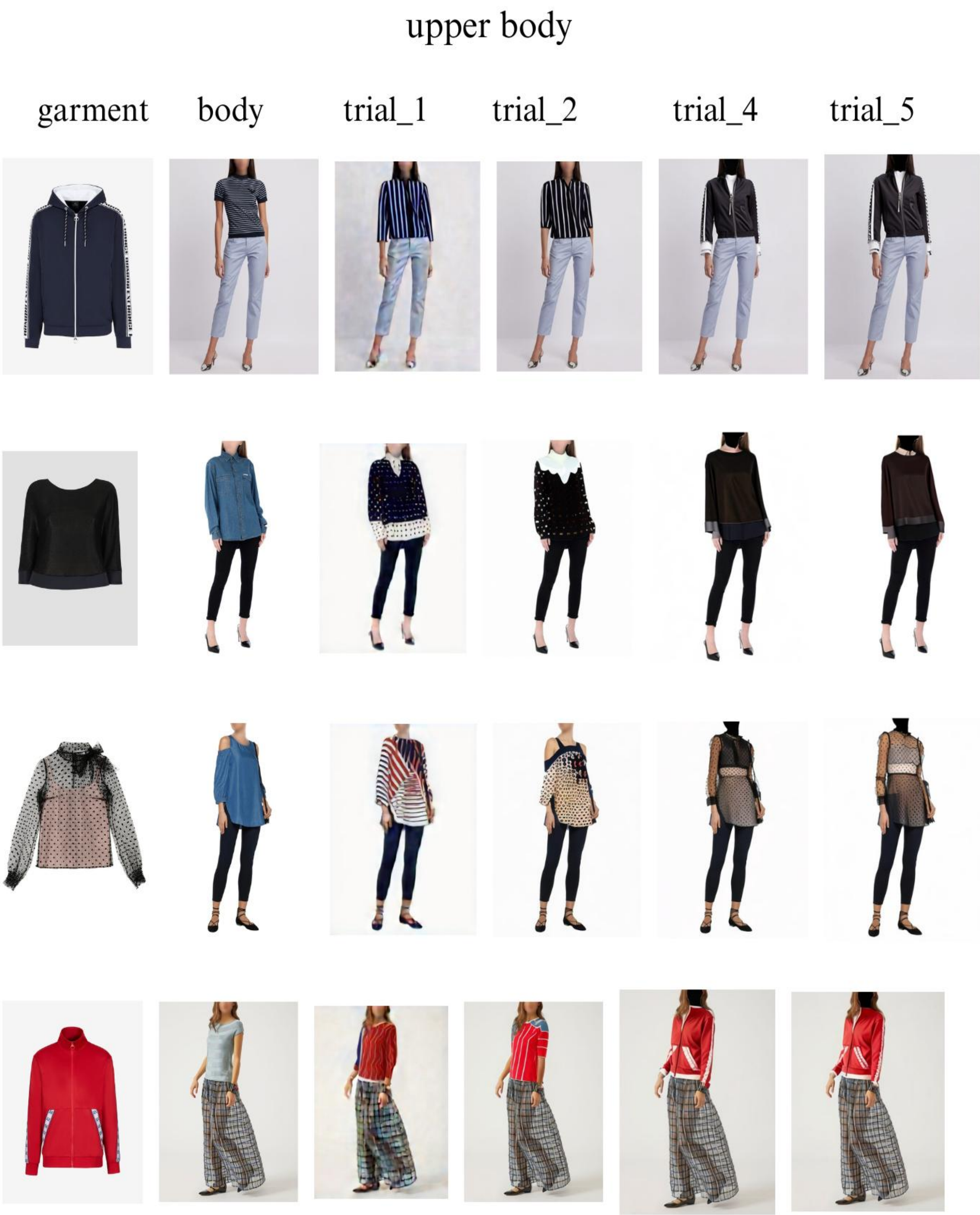


Figure 4.4.2 Qualitative results on our upper dataset

Figure 4.4.3 Qualitative results on our upper dataset

Experiment Objectives

The experimental phase was designed to evaluate the effectiveness of our VTON system across several key dimensions:

1. **Try-On Realism**: Assess how naturally and accurately garments are synthesized onto the model image.

2. **Garment Variety Coverage**: Test the system's ability to handle different clothing categories, including upper wear, lower wear, and full body veiled outfits.

3. **Cultural Relevance**: Ensure the model accommodates culturally specific clothing elements, such as hijabs and modest fashion.

4. **Preprocessing Efficacy**: Analyze the impact of segmentation, mask generation, and face blurring on output quality.

5. **Model Robustness**: Examine system behavior across varied poses, lighting conditions, and clothing styles sourced from multiple datasets.

System Configuration

- **Model Architecture**:
- **Try-On Engine**: LADI-VTON[3]
- **Core Components**:
    - TPS Warping: Performs geometric alignment of garments to body pose.
    - EMASC Module: Refines garment contours and spatial placement.
    - LDM: Generates photorealistic images from latent space.
        - VAE: Compresses and reconstructs images for diffusion processing.

- **Preprocessing Workflow**:
  - Semantic Segmentation: Produces SCHP-style label maps for up to 18 semantic body regions.
  - Pose Estimation: Extracts human keypoints to define body pose and structure.
  - Cloth Mask Extraction: Generates binary masks of garments using GrabCut or SAM to isolate clothing regions.
  - Face Blurring: Blurs the face based on segmentation to reduce identity artifacts and guide focus to the clothing.

- **Dataset Composition**:
  - Images sourced from Egyptian fashion brands including Mamzi, Niswa, Saqhoute, and What's Up.
  - Dataset includes diverse samples: veiled and unveiled models, various lighting conditions, and poses.

- **Hardware & Frameworks**:
  - GPU: Assumed usage of NVIDIA Tesla V100 or A100 for diffusion-based inference.
  - Frameworks: PyTorch, TorchVision, OpenCV.
  - Batch Size: Small (1–2), due to high memory demands of diffusion models.

Results and Analysis

A. Full-Body Veiled Dresses

- **Performance**:
  - The system retained hijabs and head coverings accurately, without distortion or blending artifacts.
  - Long dresses aligned well with body pose and preserved vertical texture consistency.
- **Conclusion**:
  - Demonstrates the model's ability to generate culturally appropriate try-on results.
  - Effective for inclusive fashion applications targeting Middle Eastern consumers.

B. Upper Garments (Blouses, Jackets, Shirts)

- **Performance**:
  - TPS effectively aligned sleeves and shoulder regions.
  - EMASC refinement corrected typical misalignments in sleeve curvature and cuffs.
  - Face blurring reduced focus on identity and improved garment clarity.
- **Key Improvement**:
  - Pre-EMASC outputs often showed detached or floating sleeves; EMASC resolved this by enhancing local spatial coherence.

C. Lower Garments (Pants, Skirts)

- **Performance**:
  - Warping was successful around the legs when pose keypoints were precise.
  - Skirts maintained realistic drape and movement, especially with appropriate cloth masks.
  - Minor artifacts appeared in occluded or crossed-leg scenarios.
- **Insight**:
  - Lower garments are more sensitive to keypoint accuracy; preprocessing quality strongly influences outcome.

Summary of Findings

- The LADI-VTON[3] architecture consistently produced **clear, well-aligned, and photorealistic results**, particularly when the full preprocessing pipeline was applied.
- **Minimal visual artifacts** were observed, and **texture fidelity** of garments was largely preserved.
- The system was especially strong in synthesizing veiled outfits, confirming its suitability for culturally aware applications.

Key Highlights

- **Output Demonstrations**:
  - Comparisons between original and warped garments.
  - Outputs before and after EMASC refinement.
  - Results with and without face blurring.
  - Side-by-side visualizations across garment categories: veiled dresses, upper wear, lower wear.

- **Strengths**:
  - High accuracy in pose-guided garment alignment.
  - Realistic cloth textures and boundaries.
  - Supports culturally modest fashion, including hijabs and long dresses.
  - Stable performance across various body types and clothing styles.
- **Limitations and Future Directions**:
  - Minor misalignments still occur in lower garments during complex poses.
  - Further training on diverse Egyptian datasets (including plus-size and child models) could improve generalization.
  - Exploring newer backbone models (e.g., diffusion transformers) may enhance realism and inference speed.

5- User Manual

1. Normal Users

I. Start Screen

The **Start Screen** is the first interface users encounter when opening the **TryFit** application. It displays the app name, a short tagline, and a welcoming illustration that highlights the virtual try-on concept.

How to use: When the app opens,tap the **"Get Started"** button at the bottom of the screen to proceed to the **Login** or **Registration** options. This screen serves as a gateway into the core features of the application.

Figure 5.1 Start screen

II. Login Screen

The **Login Screen** allows users to securely access their accounts. It includes fields to enter an email and password, along with helpful navigation options for password recovery and account registration.

How to use:

- Enter your registered email address in the **"Email"** field.
- Enter your password in the **"Password"** field.
- Click the **"Login"** button to access your account.
- If you forgot your password, click on **"Forget password?"** to reset it.
- If you don't have an account yet, click on **"Create account"** to register.

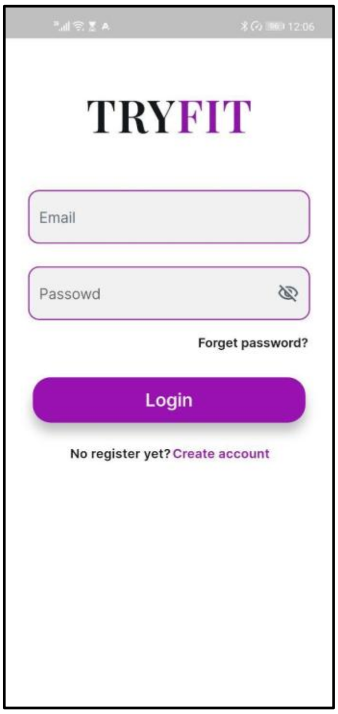


Figure 5.2 Login screen

III. Register Screen

The **Register Screen** allows new users to create an account to access the application's features. It includes fields for entering the user's full name, email, password, and password confirmation. It also provides a navigation option for users who already have an account.

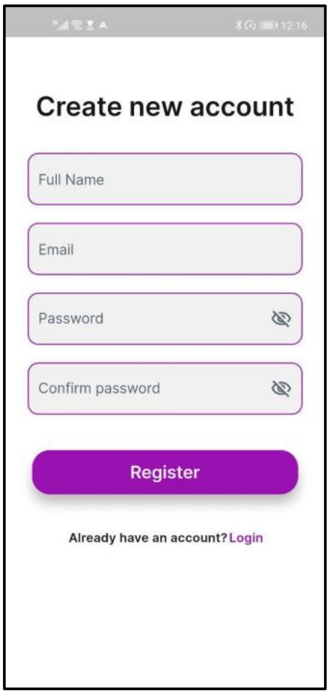


Figure 5.3 Register screen

How to use:

- Enter your full name in the **"Full Name"** field.
- Enter a valid email address in the **"Email"** field.
- Enter a secure password in the **"Password"** field.
- Re-enter the same password in the **"Confirm Password"** field.
- Click the **"Register"** button to create your account.
- If you already have an account, click on **"Login"** to return to the login screen.

IV. Home Page

The **Home Page** serves as the central navigation hub for the application. It provides quick access to key sections: **Upper**, **Lower**, and **Full**, each designed to streamline user workflows.

Figure 5.4 Home screen

How to Use:

- Navigate to Upper Section:
  Click the "**Upper**" button to view condensed data or primary controls.
- Navigate to Lower Section:
  Click the "**Lower** " button to reveal supplementary options.
- Navigate to Full Section:
  Click the "**Full** " button to switch to a detailed view.

V. Navigation Bar

The **Navigation Bar** is located at the bottom of the screen and allows users to quickly switch between the main sections of the application. It remains visible across all primary screens for easy and consistent access. includes the following options:

1-Home

2-Profile

3-Search

4-Favorites

5-Downloads

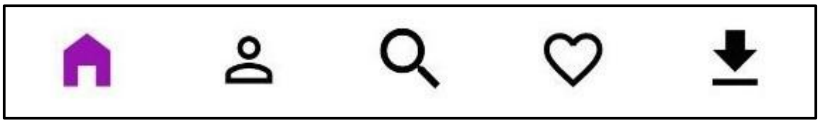


Figure 5.5 Navigation bar

How to Use:

- Switch Screens:
  - Tap any icon (e.g., **Search**) to jump directly to that page.
- Active Tab Indicator:
  - The highlighted icon (e.g., colored differently) shows your current screen.

VI. Category Page

The Category Page displays all clothing items under the selected category (**Upper**, **Lower**, or **Full**). Users can browse products, view individual items in detail, add them to favorites, or try them on virtually.

How to use:

- After selecting a category from the Home Screen, a list of relevant clothing items will be displayed.
- Tap on a **product image** to view the item in a separate full-screen view.
- To add an item to your **favorites**, tap the **heart icon** on the item.
- To try on the clothing item, tap the **"Try On Now"** button.

- Tap the **back icon** in the top corner to return to the Home Screen at any time.

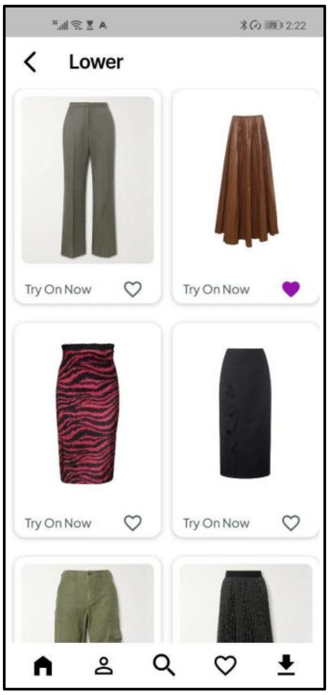


Figure 5.6 Lower category screen

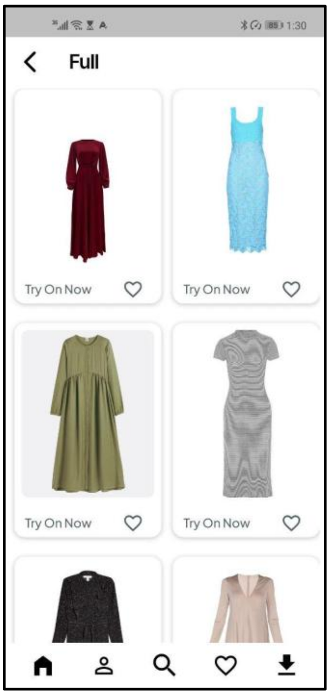


Figure 5.7 Full category screen

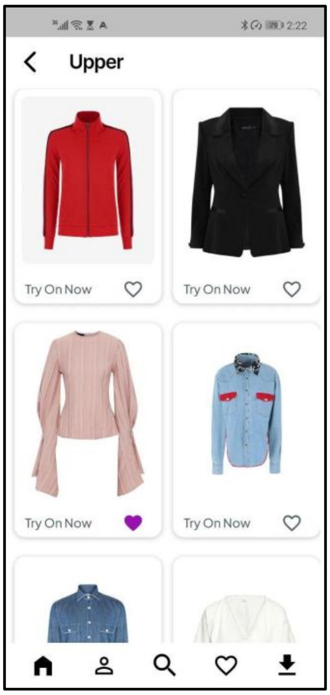


Figure 5.8 Upper category screen

VII. Profile Page

The **Profile Page** allows users to manage their personal information and access additional features such as downloads and app details. It also provides an option to securely log out of the application.

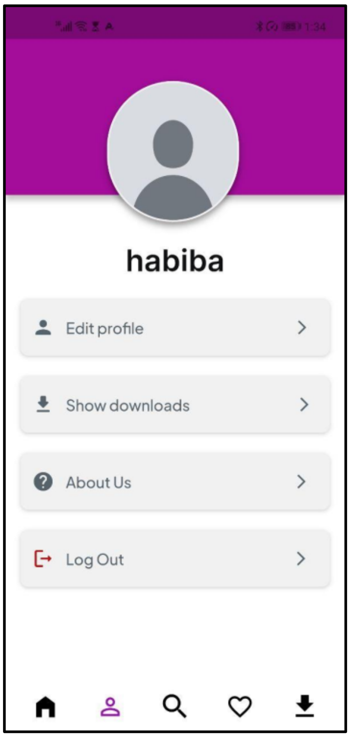


Figure 5.9 User profile screen

How to use:

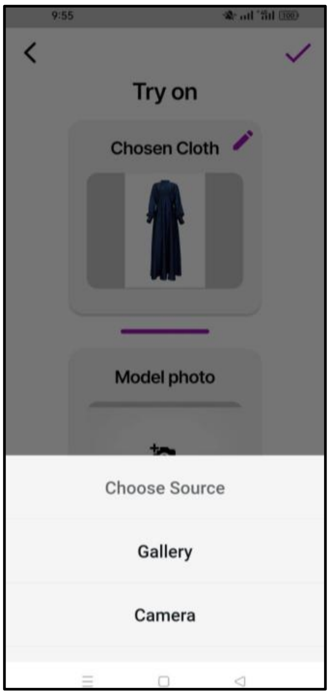
- Tap **"Edit Profile"** to update your name, password, or profile image.
- Tap **"Show Downloads"** to view all previously saved try-on results.
- Tap **"About Us"** to learn more about the app and its purpose.
- Tap **"Logout"** to securely sign out of your account.

VIII. Try-On Page

The **Try-On** Page allows users to virtually try clothing items on their selected photo. Users can choose the source of their image, initiate the try-on process, and change the selected product if needed.

How to use:

- The selected **clothing item image** will be displayed at the top of the screen.
- Tap the **camera icon** to choose the image of yourself that the item will be tried on. You can select a photo from your **gallery** or take a new one using the **camera**.
- Once your image is selected, tap the **checkmark icon** to confirm and start the virtual try-on process.
- If you wish to try a different item instead, tap the **edit icon** to go back and select a new product.
- Tap the **back icon** to return to the previous screen if needed.



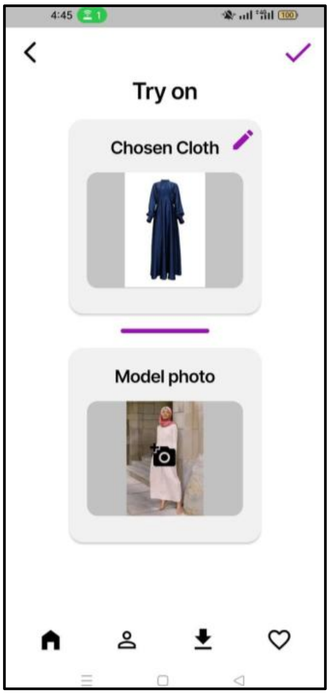


Figure 5.10 Try on screen

IX. Result Page

The **Result Page** displays the final virtual try-on output, combining the user's image with the selected clothing item. It allows users to download their results or try different clothing options.

How to use:

- After completing the try-on process, the **result image** will be displayed on this page.
- To save the result to your device, tap the **download icon**. The image will be stored in your downloads section.
- To try the same image with another clothing item, tap the **"Try Another Choices"** and select a different product.
- Tap the **back icon** to return to the previous screen if needed.

Figure 5.11 Try on result screen

X. Search Page

The **Search Page** allows users to find specific clothing items by filtering based on **color** and **type**. It helps users explore the catalog more efficiently based on their preferences.

How to use:

- Tap on the **color filter** to choose a preferred color (e.g., black, white, red, etc.).
- Tap on the **type filter** to select the type of clothing (e.g., **blouse**, **coat**, **shirt**, etc.).
- After selecting your filters, tap the **search icon** to display the results.
- Browse through the filtered items and tap on any product to view or try it on.
- Tap the **back icon** to return to the Home Screen at any time.

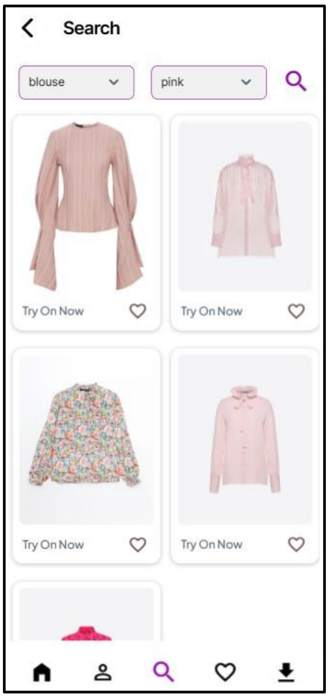


Figure 5.12 Search screen

XI. Wishlist (Favorites) Page

The **Wishlist Page** displays all clothing items that the user has marked as favorites. It allows users to quickly revisit and manage the items they are most interested in.

How to use:

- View all items you've added to your favorites by tapping the **Favorites icon** in the navigation bar.
- Each item appears with its image and relevant details.
- To remove an item from your favorites, tap the **heart icon** again — it will be unmarked and removed from the list.
- Tap on any product to view it in full or try it on virtually.
- Tap the **back icon** to return to the Home Screen at any time.

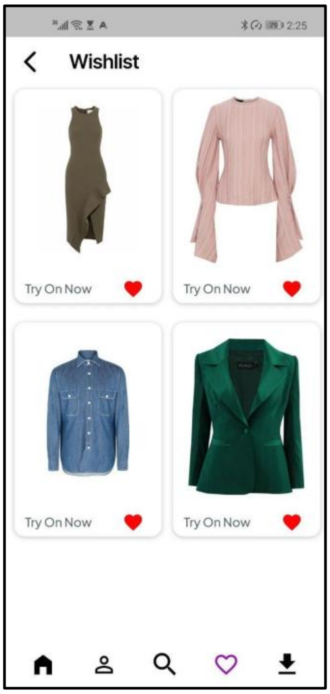


Figure 5.13 Wishlist screen

XI. Provider Profile Page

The **Provider Profile Page** allows authenticated providers to manage their profile and clothing items uploaded to the TryFit app. It includes all the functionalities available to normal users, with additional tools for product management.

How to use:

- Tap **"Edit Profile"** to update your name, password, or profile image.
- Tap **"Show Downloads"** to view all previously saved try-on results.
- Tap **"About Us"** to learn more about the app and its purpose.
- Tap **"Logout"** to securely sign out of your account.
- Tap **"Add Item"** to upload a new product (refer to Section XII).
- Tap **"Delete Item"** to manage and remove items you've uploaded.

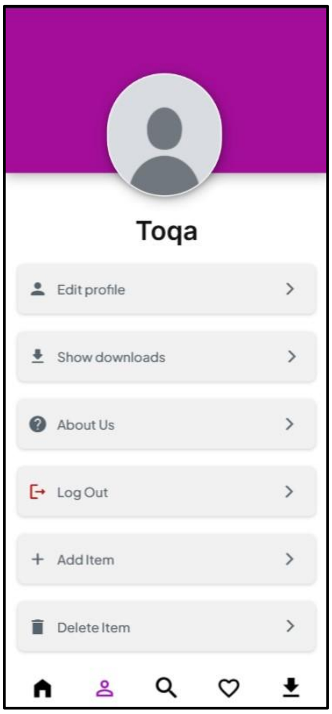


Figure 5.14 Provider profile screen

XII. Add Product Page (Provider Only)

The **Add Product Page** allows providers to upload new clothing items to the TryFit platform. Providers can add an image of the clothing piece, assign it to a **category**, and label it with its **type** and **color**.

How to Use:

- Tap the **camera icon** to upload the clothing photo from the gallery or take a new picture.
- Use the dropdown menus to:

  Select the main **category** of the item (e.g., Upper).

  Choose the **type** of clothing (e.g., blouse).

  Optionally tag with **color** (e.g., jeans).
- Tap the **checkmark icon** in the top-right corner to confirm and upload the item.

Figure 5.15 Add product screen

XIII. Delete Product Page (Provider Only)

The **Delete Product Page** enables providers to manage the clothing items available in the TryFit application by removing any item that is no longer needed or relevant. This feature is essential for maintaining an up-to-date and curated product catalog.

How to Use:

- Navigate to the **Delete Product** screen via the **Provider Profile**.
- Browse the list of products to locate the one you wish to remove.
- Tap the **Trash Icon** next to the item.
- The item will be **instantly deleted** from the app and removed from all categories and try-on availability for users.

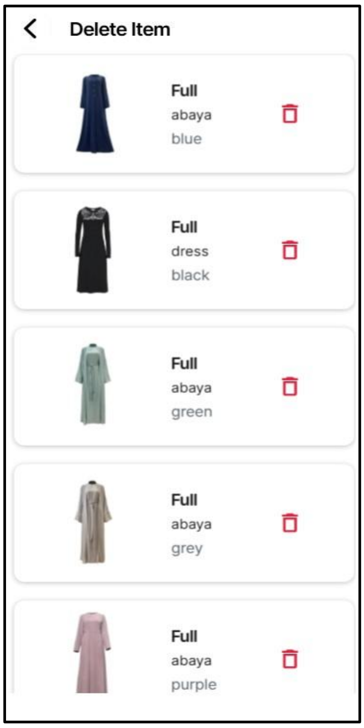


Figure 5.16 Delete product screen

6- Conclusion and Future Work

6.1 Conclusion

This project successfully developed **TryFit**, a culturally inclusive VTON system that addresses key limitations in existing solutions particularly the lack of support for veiled women and modest fashion. By integrating state-of-the-art deep learning models such as LaDI-VTON, CLIP, TPS, and EMASC with semantic preprocessing techniques, the system ensures privacy and realism in generated outputs. The architecture is built to be modular and scalable, utilizing Flask for backend processing, Firebase for cloud data management, and FlutterFlow for an intuitive mobile interface. Users can try on upper, lower, and full-body garments, including hijabs, by simply uploading their image and selecting a desired item. Through rigorous experimentation across multiple datasets both public and custom-scraped from local Egyptian brands—the system demonstrated strong performance in pose-aware garment alignment, texture preservation, and culturally respectful rendering. Notably, the system excelled in synthesizing veiled dresses with high realism and minimal visual artifacts. These results validate the effectiveness of TryFit in enhancing user trust, promoting inclusivity in fashion technology, and providing a powerful tool for modern e-commerce applications.

6.2 Future Work

I. Multi-Layer Try-On Support:

Introduce functionality for **multi-layered try-ons**, such as trying a shirt under a jacket or layering scarves and accessories, to simulate complete outfits.

II. Recommendation System:

Develop a **recommendation engine** that suggests clothing items based on the user's **wishlist, past try-ons, and preferences**, improving user engagement.

III. Size Recommendation Feature:

Incorporate a **size recommendation engine** using either AI-based body analysis or user-provided measurements to suggest appropriate garment sizes for better fit and satisfaction.

IV. Local Brands and Brand-Based Search:

Introduce a dedicated section for **local Egyptian brands** and provide a **brand-based search** feature that allows users to explore collections by specific designers or retailers.

V. Color Customization:

Allow users to change the **color of clothing items** dynamically within the app to visualize garments in different shades before making a decision.

VI. Expanded Hijab and Modest Fashion Data:

Although the current dataset includes hijab and modest fashion examples, future work should expand the dataset to incorporate **more diverse hijab styles, fabrics, and modest outfits** to better reflect cultural and individual differences among veiled users.

References

[1] S. Choi, S. Park, M. Lee, and J. Choo, "VITON-HD: High-Resolution Virtual Try-On via Misalignment-Aware Normalization." Accessed: Jun. 11, 2025. https://openaccess.thecvf.com/content/CVPR2021/papers/Choi_VITON-HD_High-Resolution_Virtual_Try-On_via_Misalignment-Aware_Normalization_CVPR_2021_paper.pdf.
[2] Z. Chong *et al.*, "CatVTON: Concatenation Is All You Need for Virtual Try-On with Diffusion Models," *arXiv.org*, 2024. https://arxiv.org/abs/2407.15886.
[3] D. Morelli, A. Baldrati, G. Cartella, M. Cornia, M. Bertini, and R. Cucchiara, "LaDI-VTON: Latent Diffusion Textual-Inversion Enhanced Virtual Try-On," Oct. 2023, doi: https://doi.org/10.1145/3581783.3612137.
[4] Y. Choi, S. Kwak, K. Lee, H. Choi, and J. Shin, "Improving Diffusion Models for Authentic Virtual Try-on in the Wild," *Lecture Notes in Computer Science*, pp. 206–235, Oct. 2024, doi: https://doi.org/10.1007/978-3-031-73016-0_13.
[5] Y. Xu, T. Gu, W. Chen, and A. Chen, "OOTDiffusion: Outfitting Fusion Based Latent Diffusion for Controllable Virtual Try-On," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 9, pp. 8996–9004, Apr. 2025, doi: https://doi.org/10.1609/aaai.v39i9.32973.
[6] J. Kim, G. Gu, M. Park, S. Park, and J. Choo, "StableVITON: Learning Semantic Correspondence with Latent Diffusion Model for Virtual Try-On," *Thecvf.com*, pp. 8176–8185, 2024, Accessed: Jun. 11, 2025. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2024/html/Kim_StableVITON_Learning_Semantic_Correspondence_with_Latent_Diffusion_Model_for_Virtual_CVPR_2024_paper.html.
[7] X. Gu *et al.*, "Recurrent Appearance Flow for Occlusion-Free Virtual Try-On," *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 20, no. 8, pp. 1–17, Apr. 2024, doi: https://doi.org/10.1145/3659581.
[8] https://huggingface.co/spaces/Kwai-Kolors/Kolors-Virtual-Try-On (accessed Jan. 17, 2025).
[9] https://vtry.io/ (accessed Feb. 10, 2025).
[10] https://flux1.ai/virtual-try-on (accessed Feb. 27, 2025).
[11] GoGoDuck912, "GitHub - GoGoDuck912/Self-Correction-Human-Parsing: An out-of-box human parsing representation extractor.," *GitHub*, 2019. https://github.com/GoGoDuck912/Self-Correction-Human-Parsing (accessed Feb. 15, 2025).
[12] Hzzone, "GitHub - Hzzone/pytorch-openpose: pytorch implementation of openpose including Hand and Body Pose Estimation.," *GitHub*, 2018. https://github.com/Hzzone/pytorch-openpose.git (accessed Feb. 21, 2025).
[13] https://pytorch.org/ (accessed March 21, 2025).
[14] "Dress Code Dataset," *GitHub*, Feb. 05, 2023. https://github.com/aimagelab/dress-code.

[15] "GitHub - albumentations-team/albumentations: Fast image augmentation library and an easy-to-use wrapper around other libraries. Documentation: https://albumentations.ai/docs/ Paper about the library: https://www.mdpi.com/2078-2489/11/2/125," *GitHub*. https://github.com/albumentations-team/albumentations.
[16] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," Apr. 2022. Available: https://arxiv.org/pdf/2112.10752.
[17] D. Kingma and M. Welling, "Auto-Encoding Variational Bayes," 2014. Available: https://arxiv.org/pdf/1312.6114.
[18] R. Gal *et al.*, "An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion." Available: https://arxiv.org/pdf/2208.01618.
[19] A. Radford *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," 2021. Available: https://arxiv.org/pdf/2103.00020.