**AI-Powered Virtual Garment Trial Room**

|  |  |
| --- | --- |
| **CONTENT** | **PAGE NO** |
| **Abstract** |  |
| **1.INTRODUCTION** |  |
| 1.1 Motivation |  |
| 1.2 Problem Statement |  |
| 1.3 Objective of the Project |  |
| 1.4 Scope |  |
| 1.5 Project Introduction |  |
| **2.LITERATURE SURVEY** |  |
| 2.1 Related Work |  |
| **3. SYSTEM ANALYSIS** |  |
| 3.1 Existing System |  |
| 3.2 Disadvantages |  |
| 3.3 Proposed System |  |
| 3.4 Advantages |  |
| 3.5 work Flow of Proposed system |  |
| **4. REQUIREMENT ANALYSIS** |  |
| 4.1Function and non-functional requirements |  |
| 4.2 Hardware Requirements |  |
| 4.3 Software Requirements |  |
| **5.METHODOLOGY** |  |
| 5.1 LSTM |  |
| **6. SYSTEM DESIGN** |  |
| 6.1 Introduction of Input design |  |
| 6.2 UML Diagram(class, use case, sequence, collaborative, deployment, activity, ER diagram and Component diagram) |  |
| 6.3 Data Flow Diagram |  |
| **7. IMPLEMENTATION AND RESULTS** |  |
| 7.1 Modules |  |
| **11. REFERENCES** |  |

**ABSTRACT**

The AI-Powered Virtual Garment Trial Room is an innovative solution designed to enhance the online shopping experience by enabling users to virtually try on apparel and accessories. The project addresses a major limitation of e-commerce: the inability to physically try products before purchase. Using augmented reality (AR) technology and advanced image processing, the system captures the user’s image via a webcam and superimposes selected garments and accessories onto their body in real-time.

The system leverages Haar cascade datasets for body and face detection and convolutional neural networks (CNNs) for accurate alignment of apparel. The Flask framework integrates the back-end Python scripts with an interactive HTML front-end, allowing seamless user interaction. Users can register, shop, and virtually try on items, while administrators can manage the product catalog through an intuitive interface.

This cost-effective solution eliminates the need for expensive hardware, relying instead on efficient software tools like OpenCV and Dlib. Future enhancements include the integration of advanced networks, such as Pose Alignment Network (PAN) and Texture Refinement Network (TRN), to improve accuracy and realism. By bridging the gap between physical trials and online shopping, this project promises to revolutionize the e-commerce industry and enhance customer satisfaction.

Keyword: OpenCV, Dlib, CNN.

**INTRODUCTION**

* 1. **Motivation:**

The AI-Powered Virtual Garment Trial Room is inspired by the challenges of online apparel shopping, where customers cannot physically try on garments before purchase, leading to dissatisfaction, returns, and cancellations. This project aims to address these issues by leveraging augmented reality and image processing techniques to provide a virtual trial experience. Initially conceptualized from a problem statement in the Smart Gujarat Hackathon, the project seeks to offer an affordable and accessible alternative to expensive hardware-based solutions, such as Kinect motion sensors. By combining innovative technologies with user-centric design, it enhances customer confidence and satisfaction in e-commerce platforms.

**1.2 Problem Statement**

The inability to try on apparel before purchase is a significant limitation of e-commerce platforms, leading to customer dissatisfaction, high return rates, and order cancellations. Traditional solutions, like Kinect motion sensors, are expensive and inaccessible for most users. This project aims to develop a cost-effective, AI-powered virtual garment trial room using augmented reality and image processing techniques. By allowing users to virtually try on garments and accessories through a webcam, the system enhances the online shopping experience, reduces return rates, and increases customer satisfaction, bridging the gap between physical and digital retail experiences.

* 1. **Objective of the Project:**

The main objective of this project is to provide an augmented reality-based solution for trying apparel and accessories online without the need for physical trials. This system reduces return rates and boosts customer satisfaction in e-commerce.

* 1. **Scope:**

The AI-Powered Virtual Garment Trial Room has broad applications and potential for enhancing the online shopping experience in the e-commerce industry. Its primary scope includes:

1. E-commerce Integration:
   * Seamlessly integrates with e-commerce platforms to provide users with a virtual dressing room.
   * Allows users to try on apparel and accessories virtually, reducing product return rates and cancellations.
2. Customer Experience:
   * Enhances the user experience by offering an interactive and personalized shopping journey.
   * Builds customer confidence in purchasing decisions through real-time virtual try-ons.
3. Cost-Effective Solution:
   * Eliminates the need for expensive hardware like Kinect motion sensors by utilizing image processing techniques that work efficiently on standard devices.
4. Technology Applications:
   * Leverages advanced technologies like Haar cascades, convolutional neural networks (CNNs), and augmented reality to ensure accurate body detection and realistic garment overlay.
   * Offers scalability to incorporate advanced networks, such as Pose Alignment Network (PAN) and Texture Refinement Network (TRN), for improved performance.
5. Future Expansion:
   * Adaptable for use in other industries, such as cosmetics, eyewear, and furniture visualization.
   * Potential for integration with virtual reality (VR) for a fully immersive shopping experience.

By addressing critical gaps in online retail, this project promises significant value to both customers and businesses.

* 1. **Introduction:**

The AI-Powered Virtual Garment Trial Room is an innovative solution aimed at revolutionizing the online shopping experience for apparel and accessories. It addresses a critical limitation of e-commerce platforms: the inability to physically try on garments before purchase. This augmented reality (AR)-based system provides users with a virtual dressing room, allowing them to visualize how clothes and accessories would look on their bodies in real-time. By using advanced body and face detection techniques, the system captures the user's image through a webcam or any camera device and superimposes selected products, offering an interactive and personalized experience.

The project uses Haar cascade datasets for body part detection and leverages convolutional neural networks (CNNs) for accurate positioning and alignment of virtual apparel. The garments, accessories, and other items are digitally masked and seamlessly overlaid on the user's image, ensuring a realistic and engaging experience. Users can virtually try on products such as shirts, dresses, goggles, earrings, and tiaras before making a purchase decision, reducing the chances of product returns or cancellations.

From a technical perspective, the project incorporates a Flask web framework to connect the Python-based back-end with an intuitive HTML front-end. Libraries like OpenCV, Dlib, and NumPy are utilized to handle image processing tasks, while the Tkinter library facilitates a user-friendly graphical interface for certain interactions. The system flow includes modules for both administrators and users, enabling admins to manage product catalogs and users to register, shop, and virtually try on items.

The Virtual Dressing Room stands out by being both cost-effective and practical, as it eliminates the need for expensive hardware like Kinect motion sensors. Instead, it focuses on implementing image processing algorithms that can run efficiently on standard hardware setups. Additionally, the project proposes future enhancements using advanced networks like Pose Alignment Network (PAN), Texture Refinement Network (TRN), and Fitting Network (FTN) to further improve the accuracy and realism of the virtual trial experience.

By combining state-of-the-art AR technology with user-centric design, this project not only enhances the e-commerce experience but also bridges the gap between online shopping and physical trials, making it a valuable innovation in the retail industry.

**2. LITERATURE SURVEY**

**1. Virtual Try-On Systems**

Several research studies have explored virtual try-on systems for enhancing online shopping experiences. For instance, M2E-Try-On Net introduced a model for realistic virtual try-ons, where users can overlay garments on their images. Similarly, VITON (Virtual Try-On Network) focused on generating photo-realistic apparel overlays using deep learning techniques, which served as a foundation for creating more accurate and interactive solutions.

**2. Body Detection and Pose Estimation**

Body and pose detection are critical for accurately aligning apparel on a user’s image. Haar cascades, a popular technique in image processing, have been widely used for object and face detection due to their computational efficiency. More advanced methods, such as Pose Alignment Networks (PAN), have demonstrated improved accuracy in detecting complex human poses, ensuring better garment alignment.

**3. Image-Based Augmented Reality**

Augmented reality (AR) has gained traction in virtual try-on applications. AR solutions rely on image processing algorithms and deep neural networks to overlay virtual objects on live or captured images. Studies show that AR-based try-ons significantly enhance user engagement and decision-making in e-commerce platforms.

**4. Challenges in Virtual Try-Ons**

Existing solutions often require expensive hardware, such as Kinect sensors, or fail to provide real-time accuracy. Research highlights the importance of optimizing algorithms to run on standard consumer devices without compromising performance. These challenges underline the need for cost-effective, software-based solutions.

**5. Future Directions in Virtual Dressing Rooms**

Emerging trends in virtual try-ons involve integrating advanced neural networks like Texture Refinement Networks (TRN) and Fitting Networks (FTN) to enhance realism. Additionally, the use of generative adversarial networks (GANs) for creating high-resolution apparel overlays is an area of active research. Studies emphasize the potential of combining AR and VR for immersive shopping experiences.

**3. SYSTEM ANALYSIS**

**Existing Method**

The existing virtual try-on systems primarily rely on hardware-intensive solutions like Kinect motion sensors or high-end augmented reality setups. These methods use motion-tracking sensors and cameras to align garments with body movements.

**Disadvantages**

1. High Cost: These systems require expensive hardware, making them inaccessible for smaller businesses and regular users.
2. Time-Consuming Setup: The setup and calibration of motion sensors are time-intensive, requiring skilled personnel.
3. Limited Accuracy: Despite the high costs, the alignment of garments often lacks precision, especially for varied body shapes and movements.

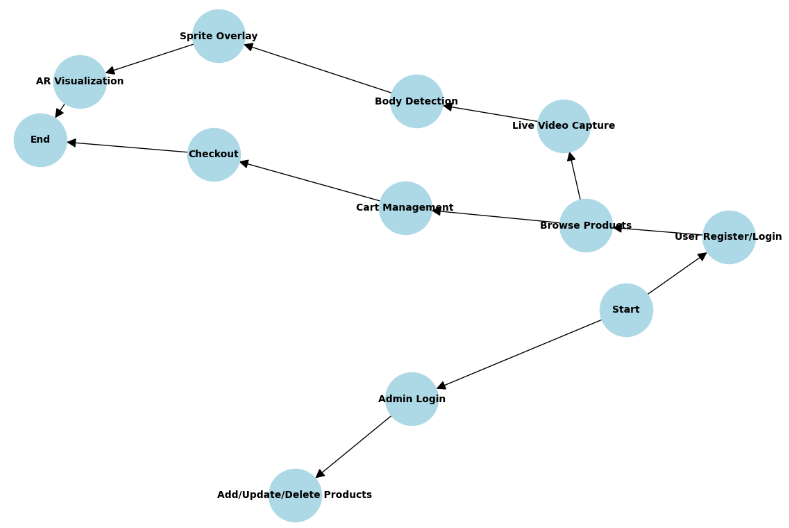
**Proposed Method**

The proposed system utilizes cost-effective software-based solutions using image processing techniques (Haar cascades) and convolutional neural networks (CNNs). It replaces hardware dependency with algorithms capable of running on standard devices.

Advantages

1. Cost-Effective: The reliance on readily available tools like webcams and open-source libraries significantly reduces implementation costs.
2. Time-Efficient: The software-based approach is faster to deploy and requires minimal setup, making it user-friendly and scalable.
3. Improved Accuracy: Advanced body detection techniques, such as Pose Alignment Networks (PAN), ensure precise garment overlay and alignment, enhancing the realism of virtual try-ons.

**Block Diagram**

****

**4.REQUIREMENT ANALYSIS**

**4.1 Function and non-functional requirements**

Requirement’s analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

**Functional Requirements**: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

1. Authentication of user whenever he/she logs into the system
2. System shutdown in case of a cyber-attack
3. A verification email is sent to user whenever he/she register for the first time on some software system.

**Non-functional requirements**: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.  
They basically deal with issues like:

* Portability
* Security
* Maintainability
* Reliability
* Scalability
* Performance
* Reusability
* Flexibility

Examples of non-functional requirements:

1. Emails should be sent with a latency of no greater than 12 hours from such an activity.
2. The processing of each request should be done within 10 seconds
3. The site should load in 3 seconds whenever of simultaneous users are > 10000
   1. **Hardware Requirements**

• Processor - I3/Intel Processor

• Hard Disk -160GB

• RAM - 8Gb

* 1. **Software Requirements:**

• Operating System : Windows 7/8/10 .

• IDE : Pycharm.

• Libraries Used : CMake (3.12.0), Dlib (19.15.0), OpenCV (3.4.2.17), SciPy (1.0.0), Tkinter, NumPy (1.18.1), Anaconda (4.8.2), Flask (1.1.1)

• Technology : Python 3.6+.

**5. METHODOLOGY**

### ****1. Haar Cascade Algorithm****

#### **Purpose**: Detect body parts such as face, eyes, and upper body in the video frames.

#### **Steps**:

1. **Feature Extraction**:
   * Sliding windows of 24x24 pixels are passed over the image.
   * Features like edges, lines, and rectangles are computed using integral images.
2. **AdaBoost Classifier**:
   * Relevant features are selected, and non-relevant ones are discarded.
   * The AdaBoost algorithm aggregates weak classifiers into a strong classifier for better accuracy.
3. **Cascade Structure**:
   * Multiple stages of classifiers are applied sequentially.
   * Each stage determines whether a region should proceed to the next stage for further analysis.
4. **Output**:
   * Bounding boxes indicating detected regions (e.g., face, upper body).

### ****2. Dlib Facial Landmark Detection****

#### **Purpose**: Detect specific facial landmarks for accessories like goggles, earrings, and tiaras.

#### **Steps**:

1. **Face Detection**:
   * Use Dlib’s pre-trained frontal face detector to identify the face region.
2. **Landmark Localization**:
   * Apply the shape predictor (shape\_predictor\_68\_face\_landmarks.dat) to extract 68 facial landmarks.
   * Landmarks correspond to features like eyes, nose, mouth, and jawline.
3. **Mapping Features**:
   * Specific landmarks are mapped to apply accessories. For example:
     + Goggles: Eyes (landmarks 36-47).
     + Earrings: Jawline (landmarks 1-5 and 12-16).
4. **Output**:
   * Precise landmark coordinates for aligning accessories.

### ****3. OpenCV Sprite Overlay****

#### **Purpose**: Overlay garments or accessories onto detected regions with transparency.

#### **Steps**:

1. **Read Sprite**:
   * Load the garment or accessory image with an alpha channel for transparency.
   * Use cv2.imread() with -1 to read the alpha channel.
2. **Resize Sprite**:
   * Adjust sprite dimensions using the width and height of the detected body part.
   * Maintain aspect ratio during resizing.
3. **Overlay Logic**:
   * Use alpha blending to combine the sprite with the video frame.
   * Formula: Output Pixel=(α×Sprite Pixel)+(1−α)×Frame Pixel\text{Output Pixel} = (\alpha \times \text{Sprite Pixel}) + (1 - \alpha) \times \text{Frame Pixel}Output Pixel=(α×Sprite Pixel)+(1−α)×Frame Pixel
   * Ensure proper alignment by adjusting offsets for position.
4. **Output**:
   * Frame with sprite seamlessly blended onto the detected region.

### ****4. Pose Alignment Network (Future Enhancement)****

#### **Purpose**: Enhance garment alignment by analyzing the user’s posture.

#### **Steps:**

1. **Pose Estimation**:
   * Use a neural network to identify keypoints of the user’s body (e.g., shoulders, hips, elbows).
2. **Alignment**:
   * Align garments based on the relative positions of keypoints.
   * Adjust the angle and scaling of the garment to fit the detected pose.
3. **Output**:
   * Garments dynamically adjusted to the user’s posture.

### ****5. Texture Refinement Network (Future Enhancement)****

#### **Purpose**: Improve the visual quality of the garment overlay.

#### **Steps**:

1. **Garment Segmentation**:
   * Segment the detected garment area from the frame using a segmentation network.
2. **Texture Transfer**:
   * Map the texture of the garment onto the segmented region.
   * Ensure smooth edges and realistic texture scaling.
3. **Output**:
   * High-quality garment overlay with natural texture transitions.

### ****6. Gated Recurrent Unit (GRU) for Future Real-Time Optimization****

#### **Purpose**: Handle temporal dependencies in frames for smoother transitions in real-time AR.

#### **Steps**:

1. **Frame Sequence Processing**:
   * Pass a sequence of consecutive frames through a GRU network.
   * Learn the temporal dependencies for stable sprite placement.
2. **Output Prediction**:
   * Predict adjustments for sprite positions in the next frame to ensure stability.
3. **Output**:
   * Reduced jitter in sprite alignment.

.

**6. SYSTEM DESIGN**

**6.1 Introduction of Input Design:**

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties −

* It should serve specific purpose effectively such as storing, recording, and retrieving the information.
* It ensures proper completion with accuracy.
* It should be easy to fill and straightforward.
* It should focus on user’s attention, consistency, and simplicity.
* All these objectives are obtained using the knowledge of basic design principles regarding
  + What are the inputs needed for the system?
  + How end users respond to different elements of forms and screens.

### Objectives for Input Design:

The objectives of input design are −

* To design data entry and input procedures
* To reduce input volume
* To design source documents for data capture or devise other data capture methods
* To design input data records, data entry screens, user interface screens, etc.
* To use validation checks and develop effective input controls.

**Output Design:**

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

### Objectives of Output Design:

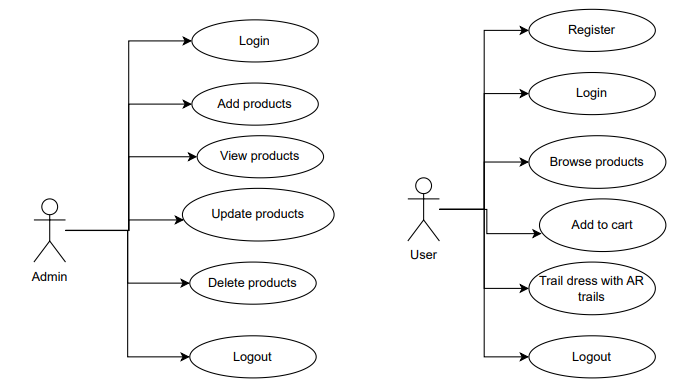
The objectives of input design are:

* To develop output design that serves the intended purpose and eliminates the production of unwanted output.
* To develop the output design that meets the end user’s requirements.
* To deliver the appropriate quantity of output.
* To form the output in appropriate format and direct it to the right person.
* To make the output available on time for making good decisions.

**7.2 UML Diagrams**

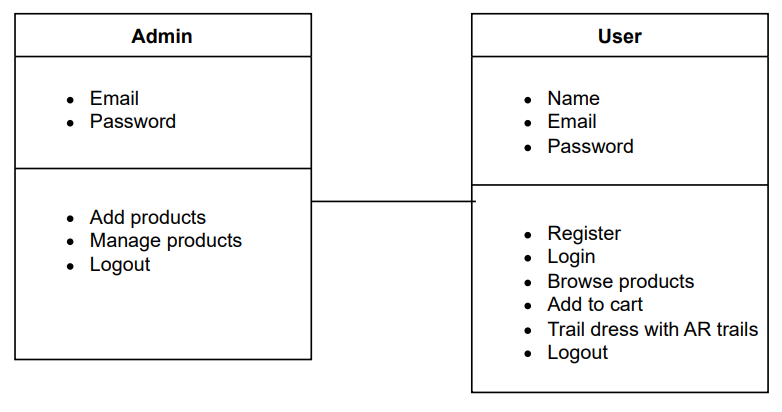
**7.2.1 USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



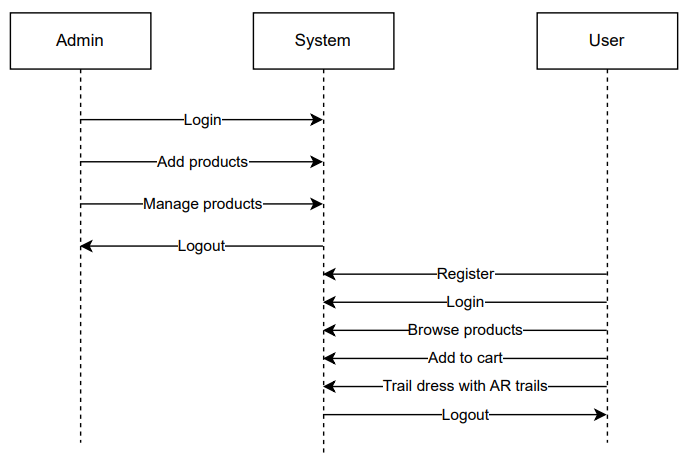
**6.2.2 CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



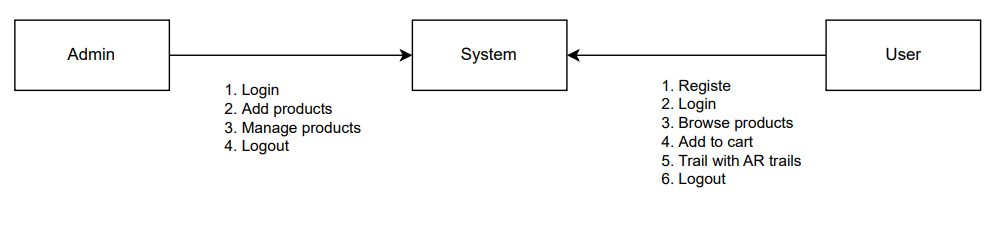
**6.2.3 SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



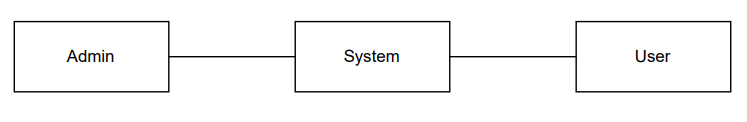
**6.2.4 COLLABORATION DIAGRAM:**

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.



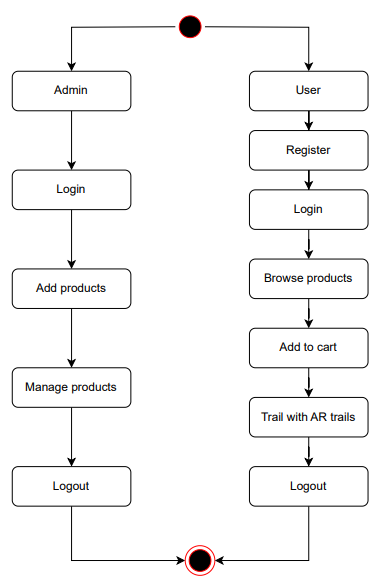
**6.2.5 DEPLOYMENT DIAGRAM**

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware’s used to deploy the application.



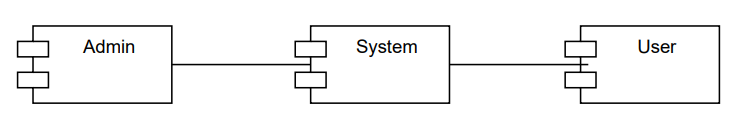
**6.2.6 ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**6.2.7 COMPONENT DIAGRAM**:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.



**6.2.8 ER DIAGRAM:**

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

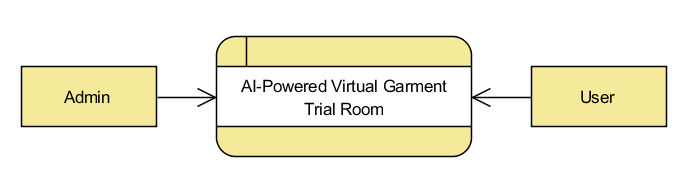
An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let’s have a look at a simple ER diagram to understand this concept.

# 

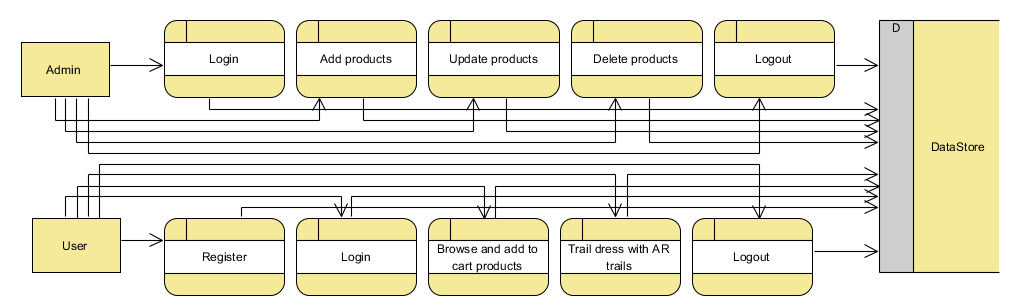
**6.2.9 DFD DIAGRAM:**

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

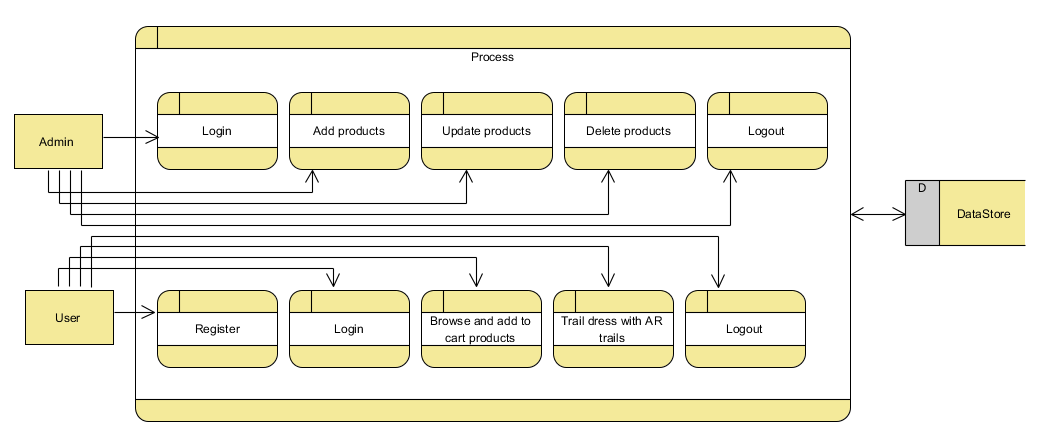
**Level 0 Diagram:**



**Level 1 Diagram:**

****

**Level 2 Diagram:**

****

**7. IMPLEMENTATION AND RESULTS**

**7.1 Modules**

### ****1. Admin Module****

#### **Purpose**: Manages the product catalog for the virtual trial room.

#### **Functionalities**:

* **Login**: Secure access for administrators to manage the system.
* **Add Product**:
  + Admin can upload new cloths:
    - Category
    - Name.
    - Cost.
    - Image of the product.
* **Update Product**:
  + Modify details such as the price or category of existing products.
* **Delete Product**:
  + Remove outdated or unavailable products from the catalog.

#### **Flow**:

1. Admin logs into the system.
2. Admin performs CRUD (Create, Read, Update, Delete) operations on the product catalog.
3. Updates are reflected in the user’s interface for real-time browsing.

### ****2. User Module****

#### **Purpose**: Enables users to interact with the system for virtual trials and purchases.

#### **Functionalities**:

* **Register**: Users create an account with basic details for a personalized experience.
* **Login**: Users securely log in to access the trial and cart functionalities.
* **Shop**: Browse products categorized by type.
* **Add to Cart**: Users can select and add items to their shopping cart for review or purchase.
* **Try On with AR**: The user can select garments and accessories to overlay on their live video feed or image for a virtual trial.

#### **Flow**:

1. Users log in and browse products.
2. Users select items to try on and visualize them in real-time.
3. Selected items can be added to the cart for checkout.

### ****3. AR Module****

#### **Purpose**: Handles live video feed and overlays garments/accessories on detected regions.

#### **Functionalities**:

* **Live Video Feed**: Captures the user’s video feed using a webcam or camera.
* **Body and Face Detection**:
  + Detects key regions using:
    - Haar cascades for initial detection.
    - Dlib for facial landmarks.
* **Overlay**:
  + Dynamically resizes and positions selected garments or accessories over detected body parts.
  + Ensures accurate alignment using offsets and transparency.

#### **Flow**:

1. The webcam captures frames.
2. Detection algorithms identify key regions for overlay.
3. Selected items are superimposed on the detected regions in real-time.

### ****5. Video Processing Module****

#### **Purpose**: Processes video frames for detecting and superimposing garments/accessories.

#### **Functionalities**:

* **Video Capture**:
  + Captures live feed using OpenCV.
* **Frame Processing**:
  + Reads frames from the video and applies detection algorithms.
* **Real-Time Rendering**:
  + Continuously updates the overlay for each new frame.

#### **Flow**:

1. Frames are captured from the webcam.
2. Detected regions are processed for sprite overlay.
3. Updated frames are rendered back to the user in real-time.

**11. REFERENCES**

1. **Han, X., Wu, Z., Wu, Z., Yu, R., & Davis, L. S. (2018),** "VITON: An Image-Based Virtual Try-On Network".
2. **Dong, H., Liang, X., Gong, K., Lai, H., Zhu, J., & Yin, J. (2019),** "FW-GAN: Flow-n-Warp GAN for Image-Based Virtual Try-On".
3. **Neuberger, G., Michalkó, A., & Märgner, V. (2021),** "Virtual try-on systems for retail: A systematic review of literature and implications for future research".
4. **Pumarola, A., Sanchez, J., Chien, J. T., & Moreno-Noguer, F. (2018),** "Dressing in Order: Recurrent Person Image Generation for Pose Guided Virtual Try-On".
5. **Huang, R., Zhang, S., Li, T., & He, B. (2021),** "AR Try-on: Real-Time Augmented Reality Try-on for Fashion E-Commerce".
6. **Zheng, W., Guo, H., & Wang, X. (2019),** "Deep Learning for Image-Based Virtual Try-On: A Survey".
7. **Ge, Y., Zhang, Z., Wang, W., Li, Z., Luo, P., & Tang, X. (2019),** "Parser-Free Virtual Try-On via Distilling Appearance Flows".
8. **Zhao, Y., Ma, S., Ning, L., & Gao, W. (2020), "**Pose-Guided Virtual Try-On for Garments in Images".
9. **Chen, Q., Liu, R., Xiao, C., & Chen, Y. (2021),** "AR Shopping: Implementation and Challenges in Augmented Reality Try-On Systems".
10. **Xu, R., Gong, K., Zhang, X., Chen, M., Han, L., & Bai, J. (2021),** "3D Virtual Try-On with Deep Learning-Based Pose Estimation".