(SRS): Software Requirement Specifications

Project Title: AI-Powered Virtual Try-On System with Real-Time AR and Cloud Integration

Submitted by:

Muhammad Shifan Saeed
Roll No: F22BINFT1M01149

Submitted to:

Ms. Kainat

Department of Information Technology
Faculty of Computing

The Islamia University of Bahawalpur

Summary:

This Software Requirements Specification (SRS) document describes the functional and non-functional requirements for an AI-Powered Virtual Try-On System with Real-Time Augmented Reality (AR) and Cloud Integration. The system addresses the significant challenge of high product return rates (30-40%) in e-commerce by enabling customers to virtually try on products before purchasing.

The proposed system utilizes advanced computer vision techniques, deep learning models, and augmented reality to provide realistic product visualization. Users can upload photos or use real-time camera feeds to try on products virtually, particularly eyewear and accessories, enabling confident purchase decisions while reducing business return costs.

Key features include real-time face detection and tracking with 68 facial landmark points, 3D product rendering with proper lighting and shadow simulation, size recommendations based on facial measurements, and a scalable cloud infrastructure supporting multiple concurrent users. The system will be accessible via responsive web and mobile interfaces.

Table of Contents:

1. <u>Introduction</u>:

The e-commerce industry faces significant challenges with high product return rates, particularly for products that customers cannot physically try before purchasing. Virtual try-on technology represents a transformative solution that bridges the gap between online and offline shopping experiences.

This project proposes the development of a cloud-based virtual try-on system that leverages computer vision, deep learning, and augmented reality technologies to provide realistic product visualization. The system enables users to see how products, specifically eyewear and accessories, look on them before making a purchase decision.

The virtual try-on system will support two primary modes of operation: real-time camera mode for live visualization and photo upload mode for trying products on saved images. Both modes will provide accurate product placement with proper scaling, lighting adjustment, and perspective transformation.

1.1. <u>Purpose</u>:

The primary goals and objectives of this project are:

- To develop a real-time virtual try-on system using computer vision and AR for eyewear and accessories
- To implement accurate face detection with 68 facial landmark identification
- To create realistic product overlay with proper scaling, lighting, shadows, and perspective adjustment
- To provide size recommendations based on facial measurements
- To build a responsive web and mobile interface with live camera and photo upload modes
- To integrate cloud infrastructure for scalability, storage, and multi-user support
- To implement real-time processing with minimal latency (under 200ms per frame)
- To create a product catalog management system with 3D model storage
- To generate analytics on user behavior, popular products, and try-on patterns
- To achieve realistic rendering indistinguishable from actual product photos

1.2. Scope:

The following items are within the scope of this project:

- Virtual try-on for eyewear (glasses, sunglasses) and accessories
- Real-time face detection and tracking using MediaPipe
- 3D product rendering using Three.js
- Web application using React.js with camera access
- Cloud deployment with user authentication
- Product catalog management with 3D models
- Size recommendation engine
- User analytics and behavior tracking

The following items are outside the scope of this project:

- Full-body clothing try-on
- Physical inventory management
- Payment processing and e-commerce transactions
- AR try-on in physical stores
- VR headset support
- Voice command interaction

1.3. Product Perspective:

The Virtual Try-On System is designed as a standalone cloud-based application that can be integrated with existing e-commerce platforms through REST APIs. The system follows a multi-tier architecture:

- Frontend Layer: Responsive web interface accessible from any device with camera support
- Processing Layer: Computer vision and 3D rendering services for real-time product visualization
- Backend Layer: REST APIs for user management, product catalog, and analytics
- Cloud Infrastructure: Scalable cloud storage, database, and compute resources

The system can function as an independent application or be embedded into existing e-commerce websites through white-label API integration.

1.4. User Characteristics:

The system will serve the following user types:

- End Users/Customers: Primary users who want to try on products virtually before purchasing. They may have varying levels of technical expertise and access the system through web browsers or mobile devices.
- Store Administrators: Business users who manage the product catalog, upload 3D models, and monitor analytics. They require access to the admin dashboard.
- System Administrators: Technical staff responsible for system maintenance, monitoring, and configuration.
- E-commerce Integration Developers: Third-party developers who integrate the virtual try-on functionality into their platforms using the provided APIs.

1.5. Similar Apps and Systems / Literature Review:

Several virtual try-on solutions exist in the market, each with their own strengths and limitations:

Warby Parker Virtual Try-On: Allows users to try on eyewear using their phone camera. Strengths include smooth user experience and accurate face tracking. Limitations include being restricted to their own product catalog and requiring a mobile app.

Ray-Ban Virtual Try-On: Web-based try-on for sunglasses. Offers good lighting adjustment but has limited product customization options and occasional tracking issues in low light.

Lenskart 3D Try-On: Popular in South Asian markets with extensive eyewear catalog. Provides good product variety but sometimes struggles with accurate sizing recommendations.

Our proposed system aims to improve upon existing solutions by providing: more accurate face tracking with 68 landmark points, better lighting and shadow simulation, ML-based size recommendations with 85%+ accuracy, and a scalable cloud architecture supporting white-label integration.

1.6. Proposed Technologies:

The following technologies will be used to develop this system:

Programming Languages:

- Python: Backend services, machine learning models, computer vision processing
- JavaScript/TypeScript: Frontend web application, 3D rendering

Computer Vision and AI:

- OpenCV: Image processing and manipulation
- Media Pipe: Face and hand detection with 68 landmark points
- TensorFlow/PyTorch: Deep learning models for size recommendation
- Dlib: Facial landmark detection backup

AR and 3D Rendering:

- Three.js: 3D rendering in web browser
- AR.js: Augmented reality features
- Blender: 3D model creation and editing

Frontend:

- React.js: Web application interface
- Tailwind CSS: UI styling
- WebRTC: Real-time video streaming

Backend:

- Fast API: Python REST API framework
- WebSocket: Real-time communication
- Redis: Caching for fast performance

Cloud Infrastructure:

- AWS/Google Cloud: Cloud platform
- AWS S3/Cloud Storage: Product models and images
- MongoDB/PostgreSQL: User data and product catalog
- Docker: Containerization
- JWT: Authentication

2. Requirements:

The Virtual Try-On System provides a comprehensive solution for online product visualization. The core functionality enables users to see how eyewear and accessories would look on them through either real-time camera feed or uploaded photos. The system processes video frames to detect faces, identify facial landmarks, and overlay 3D product models with accurate positioning, scaling, and lighting.

Users can browse a product catalog, select items to try on, and switch between different products in real-time. The system provides size recommendations based on facial measurements extracted from the camera feed. Users can save their favorite products, capture try-on images, and share them on social media. Business users can manage the product catalog through an admin dashboard, upload new 3D models, and view analytics on product popularity and user engagement.

2.1 Functional Requirements:

FR001 - User Registration

Name	FR001
Purpose	User registration allows new users to create an account to access personalized features including saved try-ons, favorites, and size profiles.
User(s)	Customer, Admin
Input	• Full Name: User's full name for profile display • Email: Valid email address for account verification and communication • Password: Minimum 8 characters with at least one number and special character • Profile Picture (Optional): User's photo for profile • Face Scan (Optional): Initial face scan for size profile creation
Output	• Account created successfully • Verification email sent to user • User redirected to onboarding flow

FR002 - User Login

Name	FR002
Purpose	Registered users can log into their accounts to access saved preferences and try-on history.
User(s)	Customer, Admin
Input	• Email: Registered email address • Password: Account password
Output	• User authenticated and session created • JWT token generated for API access • User redirected to dashboard

FR003 - Real-Time Camera Try-On

Name	FR003
Purpose	Users can try on products in real-time using their device camera with live face tracking and product overlay.
User(s)	Customer
Input	• Camera Access Permission: User grants camera access • Selected Product: Product chosen from catalog • Lighting Conditions: Ambient lighting detected automatically
Output	• Live video feed with product overlay • Real-time face tracking at 30fps • Proper lighting and shadow simulation • Size adjustment based on face measurements

FR004 - Photo Upload Try-On

Name	FR004
Purpose	Users can upload a photo to try on products without using live camera.
User(s)	Customer
Input	• Photo File: JPG, PNG, or WebP format, max 10MB • Selected Product: Product chosen from catalog
Output	• Photo processed with face detection • Product overlay applied to photo • Ability to download or share result image

FR005 - Product Catalog Browse

Name	FR005
Purpose	Users can browse available products with filtering and search capabilities.
User(s)	Customer, Admin
Input	• Search Query (Optional): Text search for products • Filters: Category, brand, price range, color • Sort Option: By popularity, price, newest
Output	• List of matching products displayed • Product thumbnails and basic info shown • Pagination for large result sets

FR006 - Size Recommendation

Name	FR006
Purpose	System provides size recommendations based on facial measurements extracted from camera or photo.
User(s)	Customer
Input	• Face Image: From camera or uploaded photo • Product Category: Type of product being tried
Output	• Recommended size (S, M, L or numeric) • Confidence score for recommendation • Alternative sizes if applicable

FR007 - Save to Favorites

Name	FR007
Purpose	Logged-in users can save products to their favorites list for later reference.
User(s)	Customer
Input	• Product ID: Identifier of product to save • User Authentication: Valid login session
Output	• Product added to user's favorites • Confirmation message displayed • Favorites count updated

FR008 - Capture and Share Try-On

Name	FR008
Purpose	Users can capture their try-on image and share it on social media or download it.
User(s)	Customer
Input	• Try-On Session: Active try-on with product overlay • Share Platform (Optional): Facebook, Instagram, Twitter, WhatsApp
Output	• High-resolution image captured • Image saved to device or shared to selected platform • Shareable link generated

FR009 - Admin Product Management

Name	FR009
Purpose	Administrators can add, edit, and remove products from the catalog.
User(s)	Admin
Input	• Product Details: Name, description, category, price • 3D Model File: GLB or GLTF format • Product Images: Multiple angles • Size Information: Available sizes and measurements
Output	• Product added/updated in catalog • 3D model processed and stored • Product available for try-on

FR010 - Analytics Dashboard

Name	FR010
Purpose	Administrators can view analytics on user behavior, popular products, and try-on patterns.
User(s)	Admin
Input	• Date Range: Period for analytics • Metrics Selection: Views, try-ons, favorites, shares
Output	• Charts and graphs displaying metrics • Top products by try-on count • User engagement statistics • Conversion tracking data

2.2 Non-Functional Requirements:

Performance Requirements:

- Real-time processing speed of under 200ms per frame
- Smooth video rendering at 30fps minimum
- Page load time under 3 seconds
- API response time under 500ms
- Support for 100+ concurrent users

Security Requirements:

- All data transmitted over HTTPS/TLS
- Passwords hashed using bcrypt with salt
- JWT tokens for API authentication with 24-hour expiry
- Input validation and sanitization on all user inputs
- GDPR-compliant data handling and privacy

Usability Requirements:

- Intuitive user interface requiring no training
- Mobile-responsive design for all screen sizes
- Works on Chrome, Firefox, Safari, and Edge browsers
- Graceful degradation for older browsers
- Clear error messages and user feedback

Reliability Requirements:

- 99% system uptime
- Face detection accuracy above 95%
- Size recommendation accuracy above 85%
- Automatic failover for cloud services
- Daily automated backups

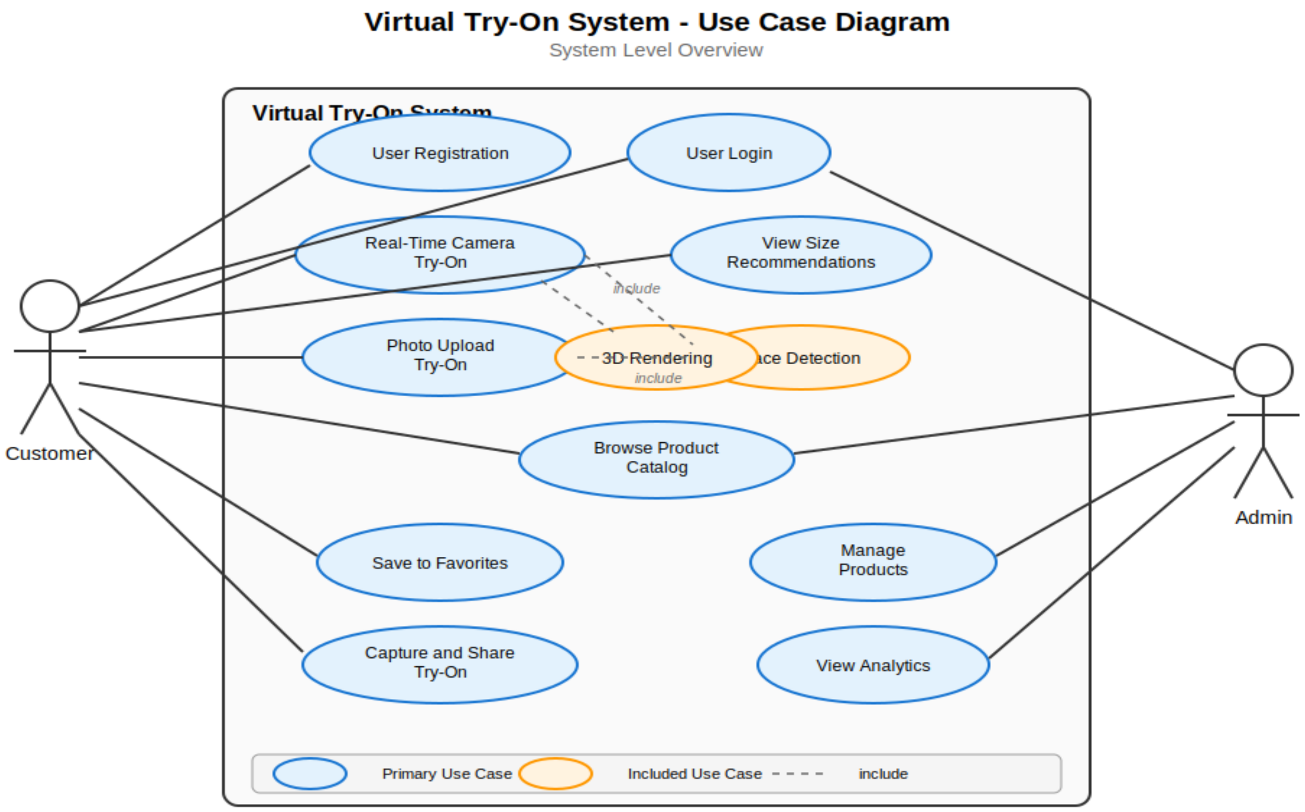
Scalability Requirements:

- Horizontal scaling for increased load
- Auto-scaling based on traffic patterns
- Support for 50+ products in catalog
- CDN for fast global content delivery

3. Use Cases and Flow of Processes:

Use cases provide formal representation of process flows defined by functional requirements. The following section presents the system-level use case diagram and detailed use case specifications for key system functions.

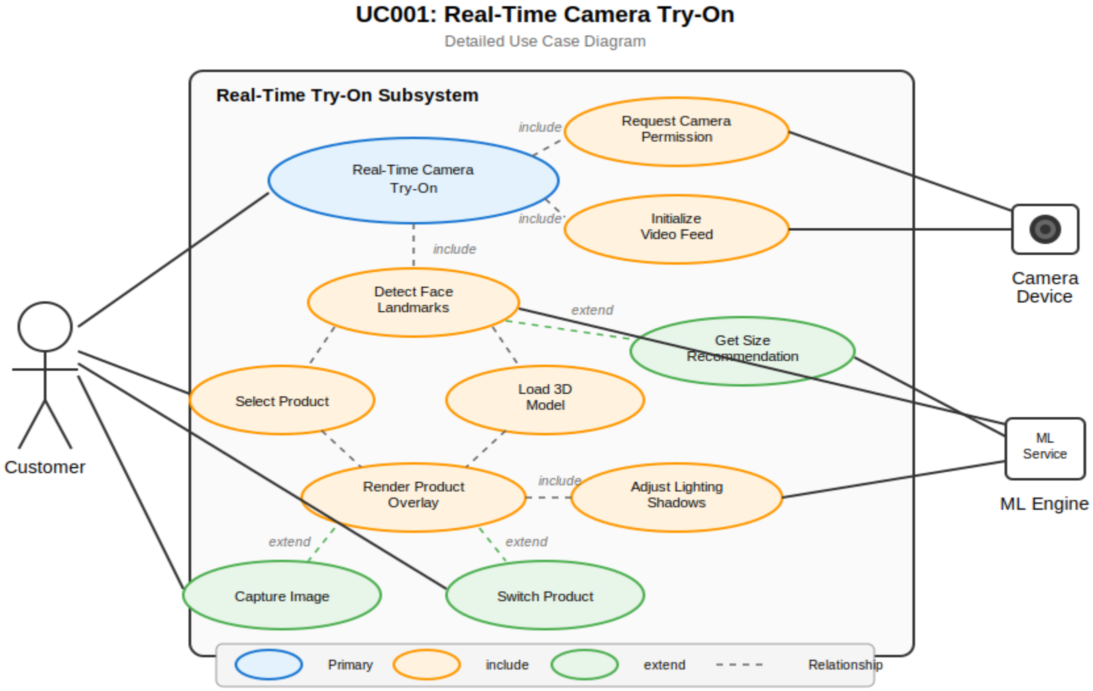
Figure 1: System Level Use Case Diagram



3.1 Use Case 1: Real-Time Camera Try-On

ID	UC001
Name	Real-Time Camera Try-On
Description	This case describes the process flow for a user trying on products using their device camera in real-time.
Requirement(s)	FR003, FR006
Actor(s)	Customer
Precondition	User has granted camera permission and has a supported browser
Postcondition	Users see live video feed with selected product overlaid on their face
Basic Flow	1. User navigates to Try-On page 2. System requests camera permission 3. User grants camera access 4. System initializes camera feed 5. System detects user's face and identifies 68 landmark points 6. User selects a product from the catalog 7. System loads 3D model of selected product 8. System calculates position, scale, and rotation based on facial landmarks 9. System renders product overlay on video feed 10. System adjusts lighting and shadows to match environment 11. User can switch products or adjust positioning 12. User can capture image of try-on result
Alternative Flow	1. User declines camera permission - System prompts to use Photo Upload mode instead 2. Face not detected - System displays message asking user to adjust position or lighting
Exceptions	1. Camera not available or not supported 2. Poor lighting conditions affecting face detection 3. Network interruption during 3D model loading 4. Browser not supporting WebRTC

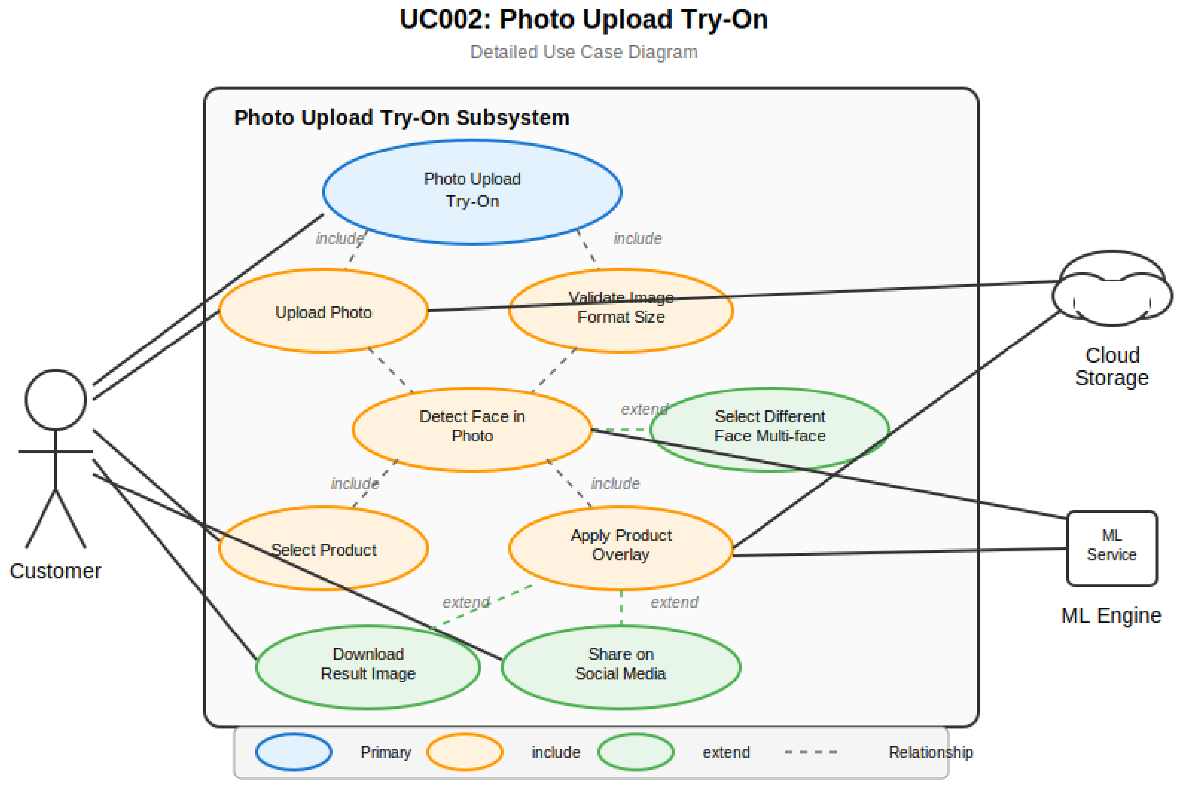
Figure 2: UC001 - Real-Time Camera Try-On Use Case Diagram



3.2 Use Case 2: Photo Upload Try-On

ID	UC002
Name	Photo Upload Try-On
Description	This use case describes the process flow for a user trying on products by uploading a photo.
Requirement(s)	FR004, FR006
Actor(s)	Customer
Precondition	User has a photo with a clearly visible face
Postcondition	User sees uploaded photo with selected product overlaid on their face
Basic Flow	1. User navigates to Photo Upload section 2. User clicks upload button or drags photo to upload area 3. User selects photo from device (JPG, PNG, WebP, max 10MB) 4. System uploads and processes the photo 5. System detects face in photo and identifies facial landmarks 6. System displays photo with face detection confirmation 7. User selects a product from the catalog 8. System loads 3D model and calculates positioning 9. System renders product overlay on photo 10. System adjusts lighting to match photo 11. User can download or share the result image
Alternative Flow	1. Multiple faces detected - System prompts user to select which face to use 2. Low quality image - System suggests uploading a higher resolution photo
Exceptions	1. No face detected in uploaded photo 2. File size exceeds 10MB limit 3. Unsupported file format 4. Photo orientation incorrect

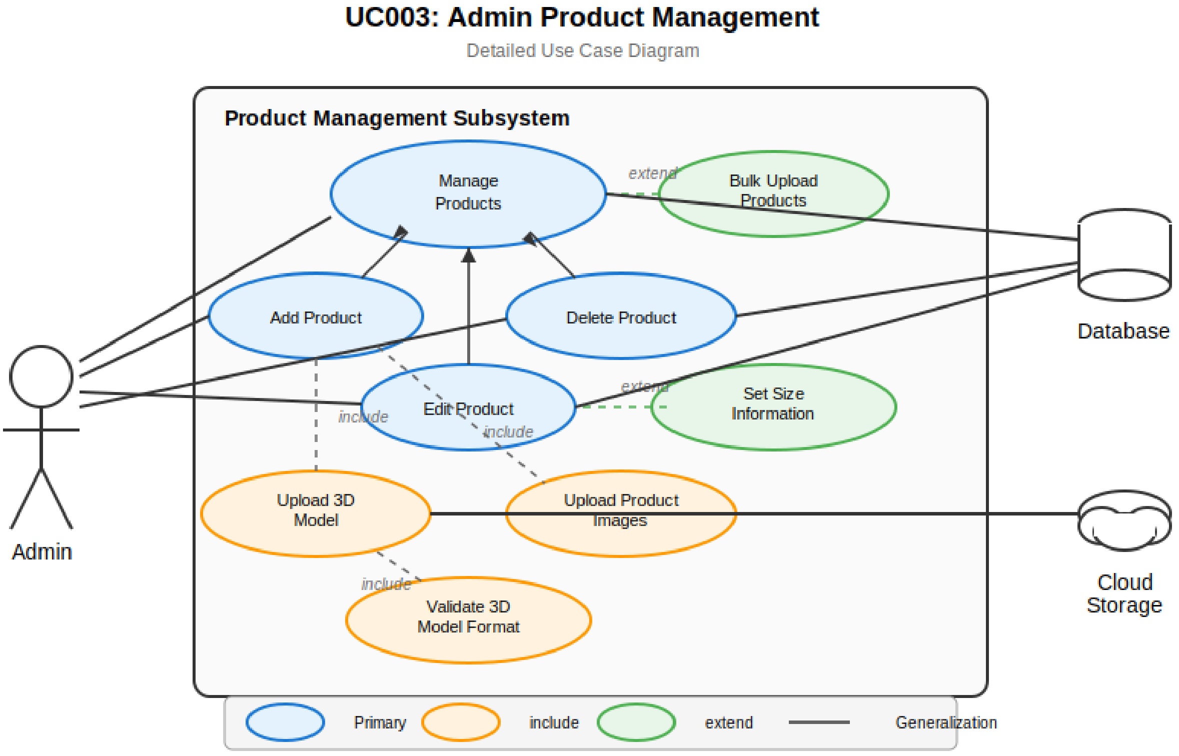
Figure 3: UC002 - Photo Upload Try-On Use Case Diagram



3.3 Use Case 3: Admin Product Management

ID	UC003
Name	Admin Product Management
Description	This use case describes the process flow for administrators adding new products to the catalog.
Requirement(s)	FR009
Actor(s)	Admin
Precondition	Admin is logged in with administrator privileges
Postcondition	New product is added to catalog and available for virtual try-on
Basic Flow	1. Admin navigates to Admin Dashboard 2. Admin clicks 'Add New Product' button 3. Admin fills in product details (name, description, category, price) 4. Admin uploads product images (multiple angles) 5. Admin uploads 3D model file (GLB/GLTF format) 6. System validates 3D model format and structure 7. Admin specifies size options and measurements 8. Admin sets product status (active/inactive) 9. Admin clicks 'Save Product' 10. System processes and stores 3D model in cloud storage 11. System creates product entry in database 12. System confirms product added successfully
Alternative Flow	1. Bulk upload - Admin uploads CSV with multiple products 2. Edit existing product - Admin modifies existing product details
Exceptions	1. Invalid 3D model format 2. Missing required product fields 3. 3D model file too large 4. Database connection error

Figure 4: UC003 - Admin Product Management Use Case Diagram



4. References:

[1] MediaPipe Face Detection Documentation, Google, https://mediapipe.dev/

[2] Three.js 3D Library Documentation, https://threejs.org/docs/

[3] WebRTC API Documentation, Mozilla MDN, https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API

[4] React.js Documentation, Meta, https://react.dev/

[5] FastAPI Framework Documentation, https://fastapi.tiangolo.com/

[6] AWS Cloud Services Documentation, Amazon, https://docs.aws.amazon.com/

[7] TensorFlow Machine Learning Documentation, Google, https://www.tensorflow.org/

[8] IEEE 830-1998 Standard for Software Requirements Specification

Appendix A: Glossary

AR (Augmented Reality): Technology that overlays digital content onto the real world view.

Computer Vision: Field of AI that enables computers to interpret visual information from images or video.

Face Landmark: Specific points on a face (eyes, nose, mouth corners) used for tracking and positioning.

GLB/GLTF: Standard file formats for 3D models used in web applications.

JWT (JSON Web Token): Secure method for transmitting authentication information.

MediaPipe: Google's framework for building multimodal machine learning pipelines.

Three.js: JavaScript library for creating 3D graphics in web browsers.

WebRTC: Web Real-Time Communication, enabling real-time video/audio in browsers.

WebSocket: Protocol for real-time bidirectional communication between client and server.