**AR Based Advertisement**



**Fall-23-28**

**BSCS Fall 2020 to 2024**

**Project Advisor: Ali Raza**

**Submitted By**

|  |  |
| --- | --- |
| Mubashra Ahmad | 70116234 |
| Muhammad Hamza | 70109065 |
|  |  |
|  |  |

Department of Computer Science & IT

The University of Lahore

Lahore, Pakistan

## *Declaration*

We have read the project guidelines and understand the meaning of academic dishonesty, particularly plagiarism and collusion. We hereby declare that the work we submitted for our final year project 1, entitled **AR Based Advertisement System** is original work and has not been printed, published, submitted before as a final year project, research work, publication or, any other documentation.

## 

**Mubashra Ahmad**

**SAP No: 70116234**

**Signature: …………………………**

**Muhammad Hamza**

**SAP No: 70109065**

**Signature: …………………………**

## *Statement of Submission*

This is to certify that **Mubashra Ahmad** Roll No. **70116234, and Muhammad Hamza** Roll No. 70109065 have successfully submitted the final project named as: **AR Based Advertisement System**, at the Computer Science & IT Department, The University of Lahore, Lahore Pakistan, to fulfill the partial requirement of the degree of **BS in Computer Science**.

**Supervisor Name: ………………………**

**Signature: …………………………**

**Date: ………………………**

## *Dedication*

This project pays tribute to our collective inspirations, fathers who instilled knowledge, and mothers who guided us, honoring their wisdom and encouragement on our collaborative journey.

## *Acknowledgement*

## 

We truly acknowledge the cooperation and support by **Mr. Ali Raza**, **Lecturer** from **The** **University of Lahore**. He has been a constant source of guidance throughout the course of this project. We are also thankful to our friends and families whose silent support led us to complete our project.

Date:

Dec 19, 2023

## *Abstract*

In recent times, almost every company has invested most of their capital in the advertisement of their product which resulted in the use of the advancement of the internet to generate personalized ads for the users. The personalization of advertisement is done through various Artificial Intelligence techniques and algorithms that collect data from the user’s searches and categorize them accordingly. Instead of just showing the user advertisements, we want to make them interactive in a way in which users can interact with the objects shown via camera. For example, virtually try clothes and glasses on themselves in 3-Dimensional form using augmented reality. This will make the online shopping experience for users more pleasant. The purpose of this project is to make an application which uses the user's device’s camera to detect the objects captured and show the users advertisements related to those detected objects. It would also allow users to try shirts, glasses, and hats virtually on themselves to see how they fit them and how they would look with the use of augmented reality. For this purpose, we would be using algorithms from machine learning, artificial intelligence, data science, computer vision and image processing. Researching this project and by studying multiple research papers related to this we found that in systems like these, they require extra hardware such as multiple cameras or Microsoft Kinect Sensors. Our goal is to use only a single camera for this purpose, preferably the camera of the user's device. We have reviewed multiple research papers to find the solution to our problem. In these research papers different artificial intelligent and digital image processing algorithms have been used such as image-based visual hull, SIFT, and Gradient-based motion estimation. Also, for motion tracking the Kinect sensor has also been used. Our system uses YOLO v3-tiny to detect objects. Then it labels these objects and on selection by user, generates ads. If the model of the product is available in our database, it gives the option to view these objects in AR using MediaPipe and OpenCV. Our current implementation covers an object detection module using YOLO v3-tiny. As for future implementation, our goal is to cover the generation of ads and mapping of 3D models in AR.

***Area of the Project***

Machine Leaning, Image Processing, Artificial Intelligence, Data Science, Augmented Reality and Web Application development.

***Technologies used***

React, Python, HTML, CSS, JavaScript.

## List of Figures

[Figure 1 Product Perspective Diagram 1](#_Toc153823181)

[Figure 2 Use Case Diagram for Signup 1](#_Toc153823182)

[Figure 3 Use Case Diagram for Signup (Email Conflict or Password Specifications) 1](#_Toc153823183)

[Figure 4 Use Case Diagram Login 1](#_Toc153823184)

[Figure 5 Use Case Diagram for Login (Invalid Email or Password) 1](#_Toc153823185)

[Figure 6 Use Case Diagram for Detected Object Pop-Up 1](#_Toc153823186)

[Figure 7 Use Case Diagram for Try-On Products 1](#_Toc153823187)

[Figure 8 Use Case Diagram for Try-On Products (Non Availability of 3D-Model) 1](#_Toc153823188)

[Figure 9 Use Case Diagram for Try-On Products (Device Incompatibility) 1](#_Toc153823189)

[Figure 10 Use Case Diagram for Try-On Products (User Exits) 1](#_Toc153823190)

[Figure 11Use Case Diagram for Logout 1](#_Toc153823191)

[Figure 12 Architecture Diagram 1](#_Toc153823192)

[Figure 13 Entity Relationship Diagram 1](#_Toc153823193)

[Figure 14 Level 0 Data Flow Diagram 1](#_Toc153823194)

[Figure 15 Level 1 Data Flow Diagram 1](#_Toc153823195)

[Figure 16 Class Diagram 1](#_Toc153823196)

[Figure 17Activity Diagram for Signup 1](#_Toc153823197)

[Figure 18 Activity Diagram for Login 1](#_Toc153823198)

[Figure 19 Activity Diagram for Detected Object 1](#_Toc153823199)

[Figure 20 Activity Diagram for User Interaction 1](#_Toc153823200)

[Figure 21 Activity Diagram for AR-Model for Selected Object Existence in Database 1](#_Toc153823201)

[Figure 22 Activity Diagram for Logout 1](#_Toc153823202)

[Figure 23 Aggregated Activity Diagram 1](#_Toc153823203)

[Figure 24 Sequence Diagram for User Signup 1](#_Toc153823204)

[Figure 25 Sequence Diagram for User Login 1](#_Toc153823205)

[Figure 26 Sequence Diagram for Detected Object Advertisement 1](#_Toc153823206)

[Figure 27 Sequence Diagram for User Logout 1](#_Toc153823207)

[Figure 28 Sequence Diagram for Admin Signup 1](#_Toc153823208)

[Figure 29 Sequence Diagram for Admin Login 1](#_Toc153823209)

[Figure 30 Sequence Diagram for Admin Logout 1](#_Toc153823210)

[Figure 31 Sequence Diagram for Admin (Delete User) 1](#_Toc153823211)

[Figure 32 Sequence Diagram for Try-On Product 1](#_Toc153823212)

[Figure 33 Collaboration Diagram 1](#_Toc153823213)

[Figure 34 State Transition Diagram 1](#_Toc153823214)

[Figure 35 Component Diagram 1](#_Toc153823215)

[Figure 36 Deployment Diagram 1](#_Toc153823216)

[Figure 37 Signup Screen 1](#_Toc153823217)

[Figure 38 Login Screen 1](#_Toc153823218)

[Figure 39 Home Page Screen 1](#_Toc153823219)

[Figure 40 Detected Object Screen 1](#_Toc153823220)

[Figure 41 AR- Advertisement Screen 1](#_Toc153823221)

[Figure 42 Try-On Product Screen 1](#_Toc153823222)

[Figure 43 Logout Screen (Admin & User) 1](#_Toc153823223)

## List of Tables

[Table 1 Functional Requirement for Signup 1](#_Toc153823225)

[Table 2 Functional Requirement for Login 1](#_Toc153823226)

[Table 3 Functional Requirement for Detected Object Pop-Up 1](#_Toc153823227)

[Table 4 Functional Requirement for Try-On Product 1](#_Toc153823228)

[Table 5 Functional Requirement for Logout 1](#_Toc153823229)

[Table 6 Signup Use Case Detail Table for Figure 2 & 3 1](#_Toc153823230)

[Table 7 Login Use Case Detail Table for Figure 4 & 5 1](#_Toc153823231)

[Table 8 Detected Object Pop-Up Use Case Detail Table for Figure 6 1](#_Toc153823232)

[Table 9 Try-On Product Use Case Detail Table for Figure 8, 9 & 10 1](#_Toc153823233)

[Table 10 Logout Use Case Detail Table for Figure 11 1](#_Toc153823234)

[Table 11 Admin Data Dictionary 1](#_Toc153823235)

[Table 12 User Data Dictionary 1](#_Toc153823236)

[Table 13 Model Data Dictionary. 1](#_Toc153823237)

**Table of Content**

[*Declaration* 1](#_Toc153828238)

[*Statement of Submission* 1](#_Toc153828239)

[*Dedication* 1](#_Toc153828240)

[*Acknowledgement* 1](#_Toc153828241)

[*Abstract* 1](#_Toc153828242)

[List of Figures 1](#_Toc153828243)

[List of Tables 1](#_Toc153828244)

[Chapter 1: Introduction to the Problem 1](#_Toc153828245)

[1.1 Introduction 1](#_Toc153828246)

[1.2 Purpose 1](#_Toc153828247)

[1.3 Objective 1](#_Toc153828248)

[1.4 Existing Solution 1](#_Toc153828249)

[1.5 Proposed Solution 1](#_Toc153828250)

[Chapter 2: Software Requirement Specification 1](#_Toc153828251)

[2.1 Introduction 1](#_Toc153828252)

[2.1.1 Purpose 1](#_Toc153828253)

[2.1.2 Scope 1](#_Toc153828254)

[2.1.3 Definitions, acronyms, and abbreviations 1](#_Toc153828255)

[2.2 Overall Description 1](#_Toc153828256)

[2.2.1 Product Perspective 1](#_Toc153828257)

[2.2.2 Product Functions 1](#_Toc153828258)

[2.2.3 User Characteristics 1](#_Toc153828259)

[2.2.4 Constraints 1](#_Toc153828260)

[2.2.5 Assumptions and Dependencies 1](#_Toc153828261)

[2.2.6 Apportioning of Requirements 1](#_Toc153828262)

[2.2.7 Specific Requirements 1](#_Toc153828263)

[2.2.8 Functional Requirements 1](#_Toc153828264)

[2.2.9 Non-Functional Requirements 1](#_Toc153828265)

[Chapter 3: Use Case Analysis 1](#_Toc153828266)

[3.1 Use Case Diagrams and Use Case Detail Tables 1](#_Toc153828267)

[3.1.1 Signup Use Case 1](#_Toc153828268)

[3.1.2 Signup Use Case Detail Table 1](#_Toc153828269)

[3.2.1 Login Use Case 1](#_Toc153828270)

[3.2.2 Login Use Case Detail Table 1](#_Toc153828271)

[3.3.1 Detected Object Pop-Up Use Case 1](#_Toc153828272)

[3.3.2 Detected Object Pop-Up Use Case Detail Table 1](#_Toc153828273)

[3.4.1 Try-On Products Use Case 1](#_Toc153828274)

[3.4.2 Try-On Product Use Case Detail Table 1](#_Toc153828275)

[3.5.1 Logout Use Case 1](#_Toc153828276)

[3.5.2 Logout Use Case Detail Table 1](#_Toc153828277)

[Chapter 4: Design 1](#_Toc153828278)

[4.1 Architecture Diagram 1](#_Toc153828279)

[4.2 ERD with Data Dictionary 1](#_Toc153828280)

[4.2.1 Data Dictionary 1](#_Toc153828281)

[4.2.1.1 Admin 1](#_Toc153828282)

[4.2.1.2 User 1](#_Toc153828283)

[4.2.1.3 Model 1](#_Toc153828284)

[4.3 Data Flow diagram 1](#_Toc153828285)

[4.3.1 The level 0 1](#_Toc153828286)

[4.3.2 The level 1 1](#_Toc153828287)

[4.4 Class Diagram 1](#_Toc153828288)

[4.5 Activity Diagram 1](#_Toc153828289)

[4.5.1 Sign Up 1](#_Toc153828290)

[4.5.2 Login 1](#_Toc153828291)

[4.5.3 Object Detection 1](#_Toc153828292)

[4.5.4 User Interaction 1](#_Toc153828293)

[4.5.5 AR Visualization 1](#_Toc153828294)

[4.5.6 Log Out 1](#_Toc153828295)

[4.5.7 Aggregated Activity Diagram 1](#_Toc153828296)

[4.6 Sequence Diagram 1](#_Toc153828297)

[4.6.1 User Signup 1](#_Toc153828298)

[4.6.2 User Login 1](#_Toc153828299)

[4.6.3 Advertisement of Detected Objects 1](#_Toc153828300)

[4.6.4 User Logout 1](#_Toc153828301)

[4.6.5 Admin Signup 1](#_Toc153828302)

[4.6.6 Admin Login 1](#_Toc153828303)

[4.6.7 Admin Logout 1](#_Toc153828304)

[4.6.8 Admin Deletes a User 1](#_Toc153828305)

[4.6.9 Virtual Try-On Product 1](#_Toc153828306)

[4.7 Collaboration Diagram 1](#_Toc153828307)

[4.8 State Transition Diagram 1](#_Toc153828308)

[4.9 Component Diagram 1](#_Toc153828309)

[4.10 Deployment Diagram 1](#_Toc153828310)

[Chapter 5: User Manual 1](#_Toc153828311)

[5.1 Sign Up Screen 1](#_Toc153828312)

[5.2 Login Page 1](#_Toc153828313)

[5.3 Home Page 1](#_Toc153828314)

[5.4 Detected Object 1](#_Toc153828315)

[5.5 Advertisement 1](#_Toc153828316)

[5.6 Try-on Product 1](#_Toc153828317)

[5.7 Logout Page 1](#_Toc153828318)

[References 1](#_Toc153828319)

### Chapter 1: Introduction to the Problem

##### Introduction

The evolution of digital advertisement market has experienced some great milestones over time, as it had to adapt, change and update to fit new mediums, technology and audiences around the world. Excellent work has been achieved in the field of object detection using artificial intelligence but object detection with fast speed is still a challenging task. In the proposed research paper [1] existing YOLO algorithm has been enhanced and optimized to detect objects in real time,

named as YOLO v3. YOLO algorithm uses Convolutional Neural Network (CNN) [2] for object

detection. The most important change it has experienced is to become much more personalized throughout history. The one medium that has the biggest impact on the history of advertising and advertising personalization is the internet, and its ability to collect billions of data points on users. The advertisements we see today are personalized according to these data points; you see advertisements on social media based on your browsing history or social media interests. Along with the evolution of advertisements, Artificial Intelligence (AI) has advanced so that one can detect objects using cameras, categories those and search for similar objects like google image search. If advertisements are designed so that users can interact with them the whole idea of advertisements can be revolutionized.

The purpose of this document is to develop an understanding of previously done work on detecting objects using cameras and other techniques and mapping 3D objects in AR. Also, the purpose is to find out limitations and drawbacks in previously done work so that they can help us improve our research. Moreover, to give an overview of algorithms and technologies that we will use in our project.

##### Purpose

Digital advertising has become a crucial component of online marketing in today's world. Traditional advertising methods primarily focus on promoting various brands, lacking the capability for customers to try on products before making a purchase. This limitation poses a significant drawback, particularly in the context of online shopping. The objective of the proposed project is to address this issue by introducing an augmented reality (AR)-based web application. This application would enable users to view diverse advertisements and experience a virtual try-on option, enhancing their shopping experience.

##### Objective

Our project’s main objectives are:

* Detecting objects present in the user's surroundings using a camera with python libraries which helps to classify images.
* Generating advertisements related to detected objects for the users to see.
* Implement AR techniques to provide a virtual try room using a single camera to give users an immersive experience.
* Develop a mobile/web application that is easy for users to interact with.

##### Existing Solution

There are fewer AR applications like Style.me, Snapchat AR lens, IKEA. These websites provide object detection capabilities, but they have specific limitations. For instance, some are limited to detecting shoes only, while others do not display advertisements. Additionally, certain websites allow users to virtually place items in their personal space to aid in making informed purchasing decisions, but they do not feature ads from external brands.

##### Proposed Solution

Our suggested solutions comprise using a camera-based object identification model that can recognize items on the user's body such hats, glasses, shoes, and simple shirts. Our model uses a large database to produce adverts that are directly connected to the objects that are being spotted. Additionally, we offer models for the advertisements so that users can virtually try on and evaluate the promoted goods before making a purchasing decision. By providing interactive virtual fitting alternatives and personalized adverts, this strategy seeks to improve the user experience.

### Chapter 2: Software Requirement Specification

##### Introduction

###### Purpose

1. **Purpose of The SRS**

The SRS has a dual goal, which is described in the following paragraphs. In the beginning, it seeks to give a thorough understanding of earlier research on the mapping of 3D objects in augmented reality (AR), the detection of objects using cameras and various techniques, etc. The knowledge gained from this research project will enhance and inform it. Second, the SRS will outline the functional and non-functional requirements for the system or software that will be created as part of this project, outlining its intended functionality as well as performance goals, limitations, and quality criteria. Additionally, the SRS acts as a fundamental tool for promoting effective communication among project stakeholders, including developers, researchers, project managers, and all people involved in the project's planning, development, or evaluation.

1. **Indented Audience of SRS**

The intended audience for this SRS document encompasses a diverse group of stakeholders. This includes researchers and developers actively engaged in the project, utilizing the SRS as a guiding resource to navigate the technical and functional intricacies of their work. Project managers and team leaders will benefit from the SRS by gaining a comprehensive understanding of the project's technical requirements, constraints, and overarching goals. Quality assurance and testing teams will rely on the document to ensure that the developed system aligns with the specified requirements and industry standards. Moreover, stakeholders and sponsors, who provide financial support or express a keen interest in the project, can utilize the SRS to grasp the project's scope and objectives. Future maintainers and system administrators will find valuable insights into the system's design and functionality within the SRS, aiding them in their post-development responsibilities. Finally, the document also caters to the interests of anyone seeking a high-level understanding of the project, its objectives, and the technologies at play.

###### Scope

The project covers fields such as machine learning, artificial intelligence, data science,

augmented reality and application development. The project is to create an augmented reality based advertisement product which consists of a mobile application that will use the cell phone’s camera to capture the surroundings of the user and the user itself in order to let the user try clothes virtually and show the user advertisements based on the objects detected in the user's camera. The application will use artificial intelligence techniques to detect the user via camera instead of sensors and Microsoft Kinect system. Our goal is to develop an application which will be capable of letting the user try T-shirts/shirts, hats and glasses virtually which will already be stored as 3D models in our application.

Moreover, the application will also use data science algorithms to classify the objects detected by the cell phone camera which will then be used to show the user advertisements related to those objects. For example, if the camera detects a chair or a wall frame, then the application will show the user advertisements related to the mentioned objects..

###### Definitions, acronyms, and abbreviations

**Definitions**: a statement of the exact meaning of some words which are used in the document.

**Acronyms**: The acronyms of some abbreviations that are used in this document are as follow:

1. **YOLO-V3**
2. **OpenCV**
3. **SIFT**

**Abbreviations**: The abbreviation of some acronyms that are used in our document

are as follow:

1. YOLO-V3 is ***(You Only Look Once).***
2. OpenCV is ***(Open Source Computer Vision Library)***
3. SIFT ***(Scale-Invariant Feature Transform)***

##### Overall Description

###### Product Perspective

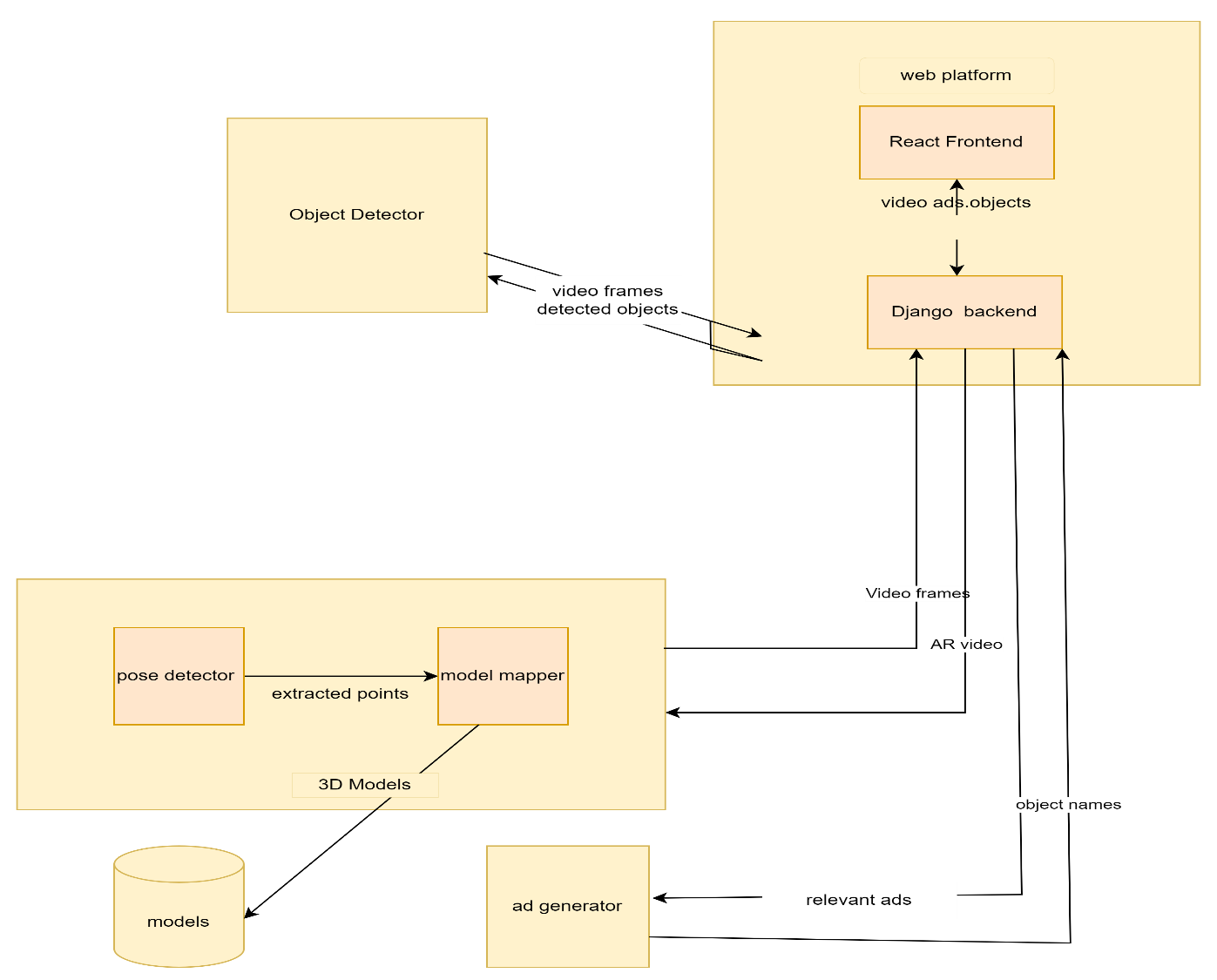
****

Figure 1 Product Perspective Diagram

This subsection is sub-divided into the following:

* **System Interfaces**

1. We will ask the user for their consent before accessing the camera of their device in order to use our system.

* **User Interfaces**

1. The user interface would be in such a way that there is no complexity faced by the user.
2. The system will be easy to use with proper labeling of features and steps included.

* **Hardware Interfaces**

1. The hardware interfaces for the system are: the device should have a working camera and the device’s camera must support AR.

* **Software Interfaces**

1. The Software interfaces of the system are Android devices should support ARCore [3]. Apple devices must support ARKit [4]. The user must have a browser like Google Chrome, Opera, Mozilla Firefox etc. to use our system’s website.

* **Communications Interfaces**

1. It involves the program, user devices, and outside components interacting seamlessly with one another. Providing a dynamic and customized user experience is dependent on successful communication as the project combines artificial intelligence (AI) and augmented reality (AR) technology.
2. **Device Camera Interface:**

**Description:** Real-time picture capture for object recognition is accomplished by the program through communication with the user's device camera.

**Protocol:** For smooth device integration, makes use of industry-standard camera APIs.

1. **Integration of Algorithms:**

**Description:** Accurate object recognition and labelling depend on communication with machine learning, artificial intelligence, and image processing algorithms (such as YOLO v3-tiny).

**Protocol:** Combines MediaPipe [5] with OpenCV [6] for effective algorithm interface.

1. **User Interaction Interface:**

**Description:** Allows interaction with virtual try-on features and communication with user inputs for object selection.

**Protocol:** Makes advantage of the application's user-friendly interfaces to enable simple interactions.

1. **Advertisement Generation Interface:**

**Description:** Communicates with the advertisement generation module to dynamically create ads based on detected objects.

**Protocol:** Utilizes data from object detection and user preferences to generate personalized advertisements.

1. **Database Connectivity:**

**Description:** Establishes communication with the product database to check for available 3D models and relevant information.

**Protocol:** Integrates with the database using secure and efficient data retrieval protocols.

1. **AR Visualization Interface:**

**Description:** Facilitates communication with AR visualization components (MediaPipe, OpenCV) for rendering 3D models.

**Protocol:** Utilizes AR frameworks and libraries for seamless integration and real-time visualization.

1. **Future Integration for Ad Generation:**

**Description**: Prepares for communication with future modules dedicated to ad generation, enhancing the interactive and dynamic advertising experience.

**Protocol:** A flexible architecture to integrate additional algorithms seamlessly.

* **Memory**

1. For the database management storage, we will be using firebase. The reason behind using firebase is that it can store 3D Models and pictures of the products to be tried on.

* **Operations**

1. The operations for this project encompass both normal functionalities and special operations, ensuring the seamless functioning of the application and addressing potential challenges. These operations are designed to optimize user experience, support system reliability, and provide a robust framework for ongoing development.

* **Normal Operations:**

1. **Object Detection:**

* **Description:** The application performs real-time object detection using YOLO v3-tiny, identifying objects captured by the user's device camera.
* **Procedure:** Utilizes the YOLO algorithm to detect and label objects within the camera feed.

1. **User Interaction:**

* **Description:** Allows users to interact with detected objects, selecting items for virtual try-on and interaction.
* **Procedure:** Integrates intuitive interfaces for users to choose and engage with virtual try-on features.

1. **Ad Generation:**

* **Description:** Dynamically generates advertisements based on detected objects and user preferences.
* **Procedure:** Utilizes data from object detection and user profiles to create personalized and engaging ads.

1. **Database Interaction:**

* **Description:** Communicates with the product database to check for the availability of 3D models and product information.
* **Procedure:** Retrieves relevant data from the database to enhance the virtual try-on experience.

1. **AR Visualization:**

* **Description:** Integrates with AR frameworks (MediaPipe, OpenCV) to render 3D models for virtual try-on.
* **Procedure:** Utilizes augmented reality technologies to overlay virtual objects onto the user's real-world environment.
* **Special Operations:**

1. **Backup and Recovery:**

* **Description:** Implements periodic backups of user preferences, profiles, and important data to ensure data integrity and recovery capabilities.
* **Procedure:** Utilizes secure backup protocols and establishes recovery procedures in case of data loss or system failure.

1. **Security Measures:**

* **Description:** Implements security protocols to safeguard user data, ensuring privacy and compliance with data protection regulations.
* **Procedure:** Regularly updates security measures, encrypts sensitive information, and monitors for potential vulnerabilities.

1. **Error Handling:**

* **Description:** Incorporates robust error-handling mechanisms to identify and resolve issues during runtime.
* **Procedure:** Logs errors, provides user-friendly error messages, and includes mechanisms for reporting and resolving issues.

1. **Performance Optimization:**

* **Description:** Conducts periodic performance assessments to optimize application speed and responsiveness.
* **Procedure:** Identifies and addresses bottlenecks, streamlines algorithms, and ensures efficient resource utilization.

1. **Future Development:**

* **Description:** Prepares for future development phases by maintaining a modular and extensible architecture.
* **Procedure:** Allows for seamless integration of additional features, algorithms, and improvements to enhance the application over time.
* **Site Adaptation Requirements**

The requirements for site adaptation are mostly focused on the following areas:

1. **Calibration of Cameras:**

Requirement: The camera must be calibrated by the system based on the particulars of the user's device.

1. **User Preferences and Profile Data:**

Requirement: User preferences and profile information must be supported by the system.

1. **Product Database Integration:**

Requirement: Integration of a product database specific to the website or business is a need for the system.

1. **Localization and Language Preferences:**

Requirement: The program must adjust to user preferences for language and region localization.

1. **Network Configuration:**

Requirement: The system must adapt to variations in network configurations and speeds.

1. **Privacy and Data Regulations Compliance:**

Requirement: In order to comply with national and international privacy and data protection standards, the system must be modified.

1. **Device Compatibility:**

Requirement: The program must adjust to the capabilities of various devices.

1. **Dynamic Ad Generation:**

Requirement: The system must modify its ad creation algorithms in response to feedback and real-time user interactions.

###### Product Functions

Product functions of the system are as follows:

**2.2.2.1 Product Function for Signup**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_01 |  |  |  |
| Name: | Sign Up |  |  |  |
| Description | Input | Output | Requirements | Basic Work Flow |
| Enter credentials for  sign up account. | Name, Email, Password is required. | Account created and saved in database, redirected to login page. | Proper Internet Connectivity is required. | The user opens the signup page, signup page is displayed asking for email and password , enters email and password and stores email and password to database and redirect user to login page. |

Table 1 Functional Requirement for Signup

**2.2.2.2 Product Function for Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_02 |  |  |  |
| Name: | Login |  |  |  |
| Description | Input | Output | Requirements | Basic Work Flow |
| The user will provide their email and password for login. After login is  successful, user shall be displayed home page | Email and Password required. | The user is logged in and home page is displayed | Proper Internet Connectivity is required. | The user opens the login page; login page is displayed asking for  email and password, enters valid email and  password and system verifies the email and  password, and redirects the user to the  home page. |

Table 2 Functional Requirement for Login

**2.2.2.3 Product Function for Detected Object Pop-up**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_03 |  |  |  |
| Name: | Detected object pop-up |  |  |  |
| Description | Input | Output | Requirements | Basic Work Flow |
| The user clicks on pop-up displayed after an object is detected in user  environment. After clicking the pop-up, related advertisement is showed to  user. | The detected algorithm is trained on the object and is able to properly label the  object. | Related advertisement of detected object is shown to user. | Proper Internet Connectivity is required and a compatible AR device camera is required. | The user clicks on pop-up of detected  Object and user tried on the selected object. |

Table 3 Functional Requirement for Detected Object Pop-Up

**2.2.2.4 Product Function for Try-on Products**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_04 |  |  |  |
| Name: | Try-on products |  |  |  |
| Description | Input | Output | Requirements | Basic Work Flow |
| The user clicks on try-on button in section of showed products. The user is  virtually tried on product in real time | The user clicks on try on button. | The user is tried on the selected object. | The user camera must contain AR core or AR kit. The user must allow the  application to use the camera. | The user clicks on try on button and tried on the selected object. |

Table 4 Functional Requirement for Try-On Product

**2.2.2.5 Product Function for Logout**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_05 |  |  |  |
| Name: | Logout |  |  |  |
| Description | Input | Output | Requirements | Basic Work Flow |
| The user will click on logout from the menu. | The user should must be logged in beforehand. | The user is logged out and user is directed to login page. | Proper Internet Connectivity is required. | The user clicks on logout button and user is logged out and redirected to  login page. |

Table 5 Functional Requirement for Logout

###### User Characteristics

The user characteristics intended to using this system are

* + - 1. The user must have compatible device for using this system.
      2. The user should now basic knowledge for using a web app like Creating account, logging in, logging out.
      3. The intended user must have a device camera that supports the ARcore\ARkit.
      4. The intended user must understand the way of using like posing appropriately for the virtual try on and understand the basic technique of object detection for better experience.

###### Constraints

Following constraints are there on the system

1. We will use both Google Colab and Jupyter to train the YOLO-V3 Tiny as training models require high GPU power.
2. We will use Netlify [7] and Heroku [8] to deploy our website as they are free to use.
3. The user with valid email and password will be able to login the system.

###### Assumptions and Dependencies

Some of the assumptions and dependencies for the system are:

1. The user’s device should have a working camera. The camera must be ARkit [4] /ARcore [3] compatible.
2. An Internet browser must be installed on the user’s device.
3. There must be a stable internet connection connected to the device.
4. The products available for try-on would be only those products whose 3D models are stored.
5. The objects detected by the system will depend on the data used to train the algorithm.
6. The user must allow access of camera in order to use the system.
7. The lighting of the user’s surrounding should be better.
8. The users who signed up will be able to login the system

###### Apportioning of Requirements

There is no as such future requirements accept some more accessories and clothing may be added to the system for virtual try on according to the increasing demand of the products and user requirement like furniture, jewels, rings, variety of caps and ties etc.

###### Specific Requirements

The requirement and design consists of the system requirements which the system needs as well as the design which the system will follow. These includes architecture and database designs, functional and non-functional requirements and much more. The requirement and design of the system are described in detail below:

###### Functional Requirements

The functional requirements are those requirements that the system is bound to do. The functional requirements of our system can be divided amongst of the user, admin and the augmented reality system.

* **Functional Requirements for the Users**

1. System will let the user to create account.
2. System will authorize the unique email, password from the user and allow login.
3. System will allow the user to add his body measurements.
4. System will allow user to open camera.
5. System will allow the user to view the advertisement related to the objects

detected by the camera.

1. System will allow the user to try eyewear virtually.
2. System will allow the user to try caps and hats virtually.
3. System will allow the user to try shirts virtually.
4. System will allow the user to edit his profile.
5. System will allow the user to delete his account.

* **Functional Requirements for Admin**

1. System will allow the admin to add 3-D models of objects.
2. System will allow the admin to remove 3-D models of objects.
3. System will allow the admin to able/disable any action.
4. System will allow the user to manually remove existing user.

* **Functional Requirements for The Augmented Reality System**

1. The system will use the camera to detect objects.
2. The system will run the YOLO-v3 Tiny [1] to detect the objects through the camera.
3. The system will label the objects.
4. The system will show the products related to the objects detected by the camera.
5. The system will show the related products at the side of the screen.
6. The system will show objects which are stored in the database.
7. The system will use the MediaPipe [5] for pose detection of the person.
8. The system will use OpenCV [6] to map 3D models in real time.

###### Non-Functional Requirements

The Non-functional requirements are those which define the constraints which affect the working and performance of the system. The Non-functional requirements of our system are the following:

* **Usability**

1. The system will have easy to use interface which will be attractive.
2. The system should be able to respond to the user’s actions in a maximum of 5 seconds.

* **Reliability**

1. The application can be used from any laptop/mobile which has augmented reality supported camera.

* **Performance**

1. The application would load within 10 seconds due to the optimization techniques available in React.

* **Design Constraints**

1. We will use both Google Colab and Jupyter to train the YOLO-V3 Tiny as training models require high GPU power.
2. We will use Netlify [3] and Heroku [4] to deploy our website as they are free to use.
3. The user with valid email and password will be able to login the system.

* **Portability**

1. The application can be used from any laptop/mobile which has augmented reality
2. supported camera.

* **Maintainability**

1. The algorithm for object detection will be kept updated by training the algorithm on more objects.

* **License Agreement**

The software and its associated documentation files are granted to anyone who obtains a copy and subscription, with the right to use, merge, distribute and sublicense the software. However, the software is provided "as is" without warranty, including fitness for a particular purpose, and non-infringement.

### Chapter 3: Use Case Analysis

##### 3.1 Use Case Diagrams and Use Case Detail Tables

###### 3.1.1 Signup Use Case

* User provides required registration information (email, password).
* System validates the provided information.
* If valid, the system creates a new user account.
* User is directed to the dashboard.

****

Figure 2 Use Case Diagram for Signup

###### Signup Use Case for Email Conflict or Password Specifications

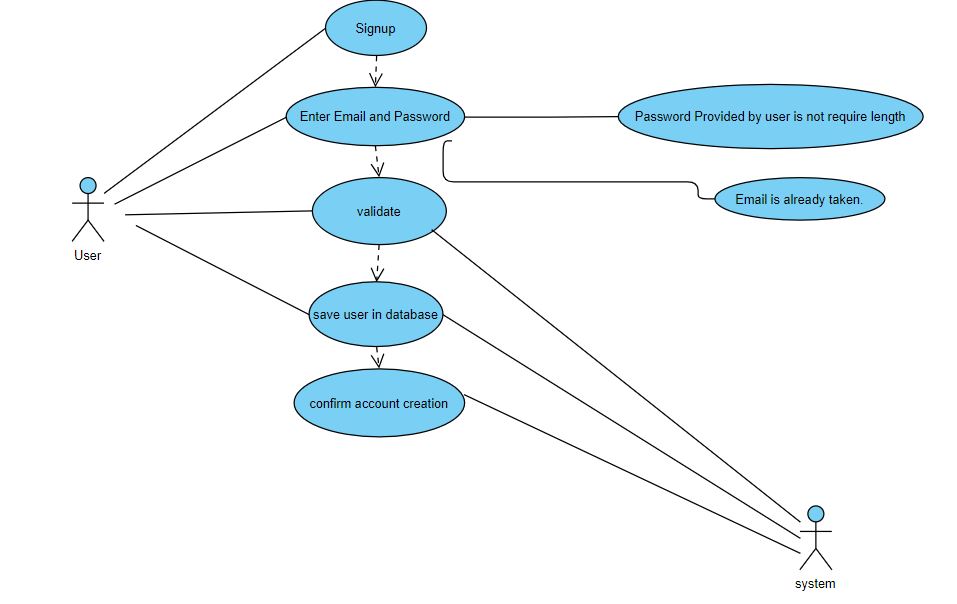


Figure 3 Use Case Diagram for Signup (Email Conflict or Password Specifications)

###### 3.1.2 Signup Use Case Detail Table

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC\_01 () | |
| Use Case Name | Signup | |
| Description | The user will provide information of email and password in displayed form. The information is then further stored in database for further login verification purpose. | |
| Primary Actor | User | |
| Secondary Actor | None | |
| Pre-Condition | The user has an existing account. | |
| Post-Condition | User information is saved in database and user is directed to login page. | |
| **Basic Flow** | | |
| **Actor Action** | | **System Response** |
| 1The user opens the signup page. | | 2The signup page is displayed asking for  Email and password. |
| 3 The user enters email and password. | | 4 the system stores email and password to  Database and redirect user t login page. |
| **Alternative Flow** | | |
| **Actor Action** | | **System Response** |
| 3The password provided by user is not  required length. | | 4-AThe system responds with an error  message. |

Table 6 Signup Use Case Detail Table for Figure 2 & 3

###### 3.2.1 Login Use Case

* User provides valid login credentials (username and password).
* System validates the credentials.
* If valid, the system grants access to the user.
* User is directed to the dashboard.

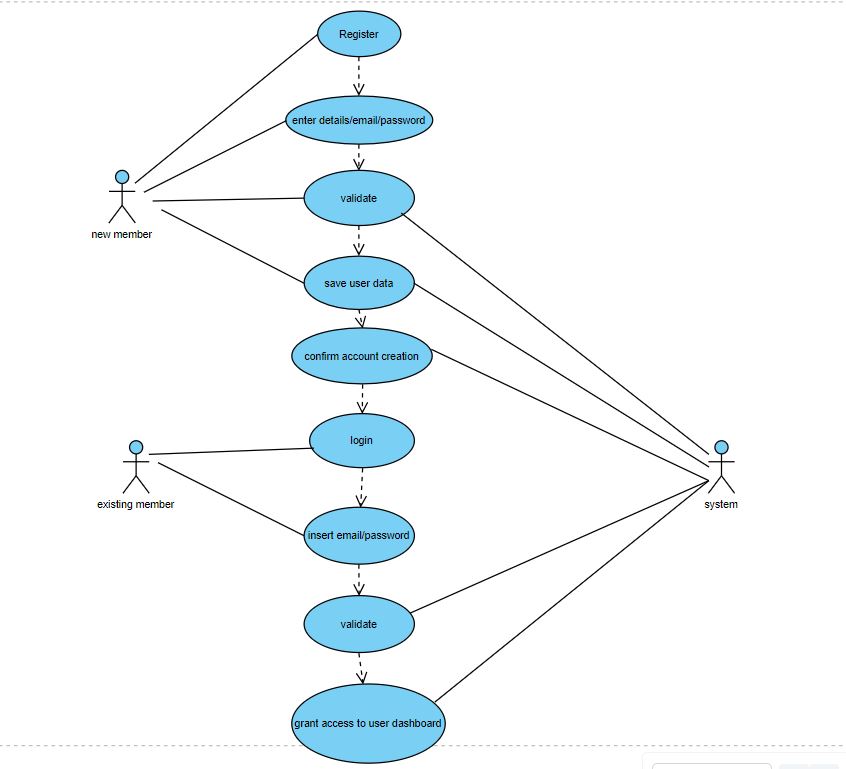


Figure 4 Use Case Diagram Login

**Login Use Case for invalid Email and Password**

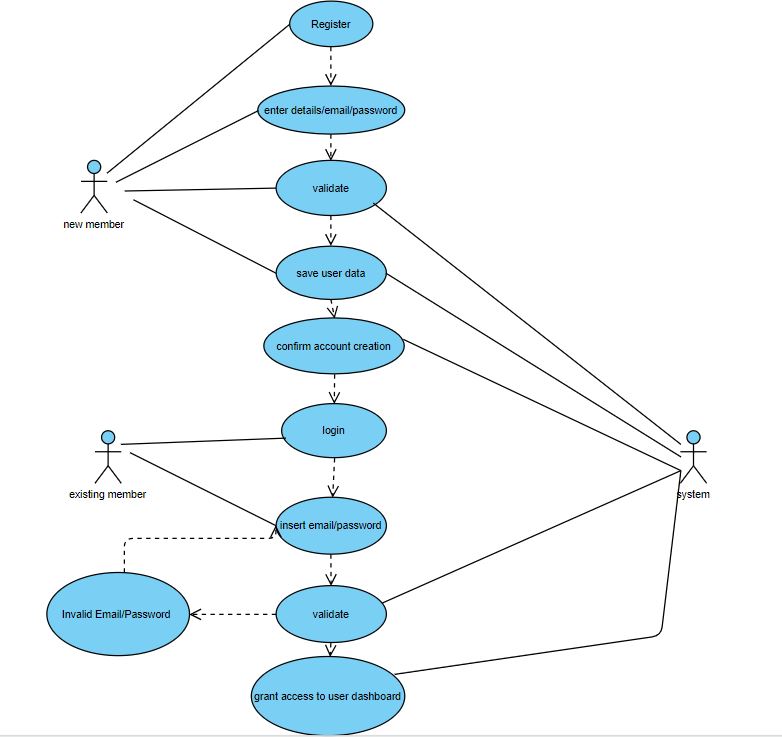
****

Figure 5 Use Case Diagram for Login (Invalid Email or Password)

###### 3.2.2 Login Use Case Detail Table

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC\_02 () | |
| Use Case Name | Login | |
| Description | The user will provide their email and password for login. After login is successful, user shall be displayed home page. | |
| Primary Actor | User | |
| Secondary Actor | None | |
| Pre-Condition | The email and password must exist in database for verification purpose. The user must not be signed in already. | |
| Post-Condition | The user is logged in and home page is displayed to user. | |
| **Basic Flow** | | |
| **Actor Action** | | **System Response** |
| 1The user opens the login page. | | 2The login page is displayed asking for  email and password. |
| 3 The user enters valid email and password. | | 4 The system verifies email and password,  and redirects the user to the home page. |
| **Alternative Flow** | | |
| **Actor Action** | | **System Response** |
| 3The user enters valid email or password. | | 4The system responds with an error message. |

Table 7 Login Use Case Detail Table for Figure 4 & 5

###### 3.3.1 Detected Object Pop-Up Use Case

* User interacts with the user interface and selects a labeled object.
* System displays a pop-up interface with additional information about the selected object.
* User explores the details presented in the pop-up.
* System provides options to interact further with the object, such as viewing in augmented reality or exploring related products.

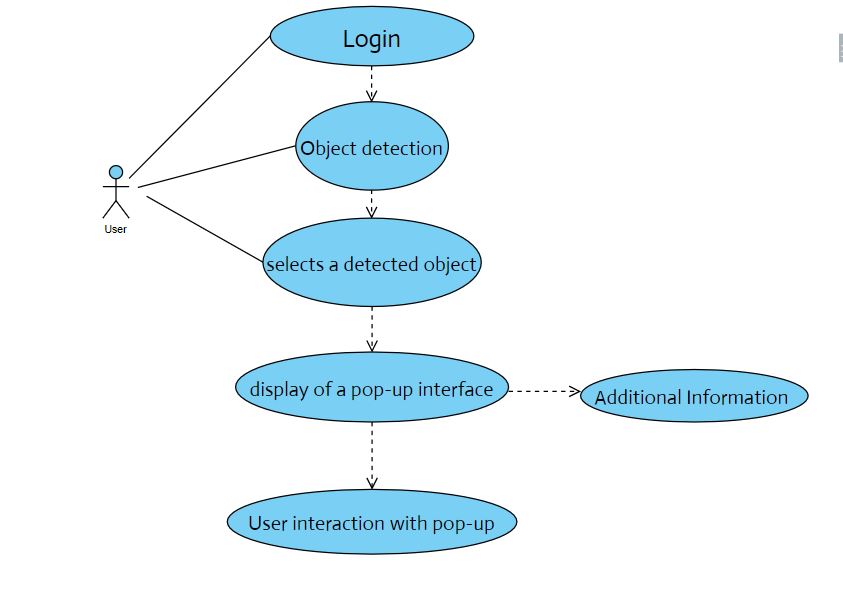


Figure 6 Use Case Diagram for Detected Object Pop-Up

###### 3.3.2 Detected Object Pop-Up Use Case Detail Table

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC\_03 () | |
| Use Case Name | Detected object pop-up | |
| Description | The user clicks on pop-up displayed after an object is detected in user  environment. After clicking the pop-up, related advertisement is showed to user. | |
| Primary Actor | User | |
| Secondary Actor | None | |
| Pre-Condition | The detected algorithm is trained on the object and is able to properly label the object. | |
| Post-Condition | Related advertisement of detected object is shown to user. | |
| **Basic Flow** | | |
| **Actor Action** | | **System Response** |
| 1 The user clicks on pop-up of detected object | | 2Similar related advertisements are shown to  user. |
| **Alternative Flow** | | |
| **Actor Action** | | **System Response** |
| 3None | | 4-ANone |

Table 8 Detected Object Pop-Up Use Case Detail Table for Figure 6

###### 3.4.1 Try-On Products Use Case

* User interacts with the user interface and selects a product labeled for try-on.
* System checks the product database for 3D models related to the selected product.
* If 3D models are available, the system provides an option for users to try on the selected product virtually using augmented reality.

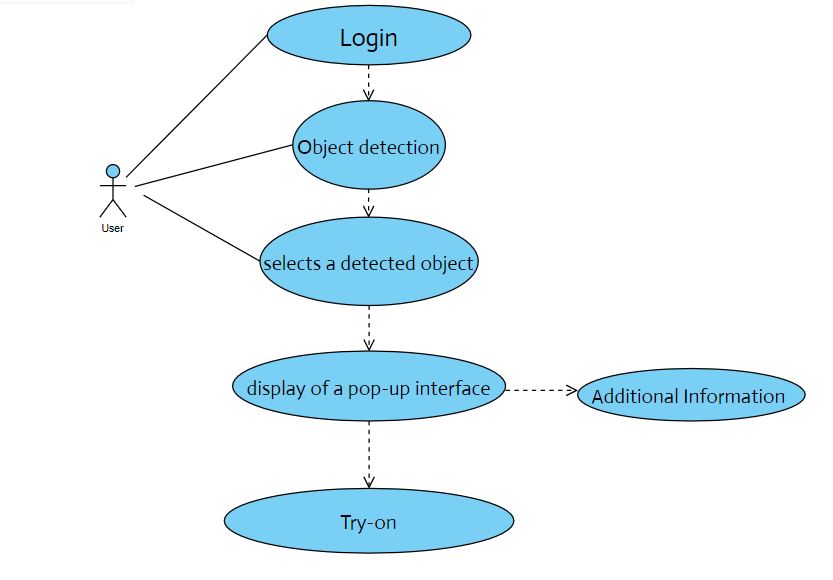


Figure 7 Use Case Diagram for Try-On Products

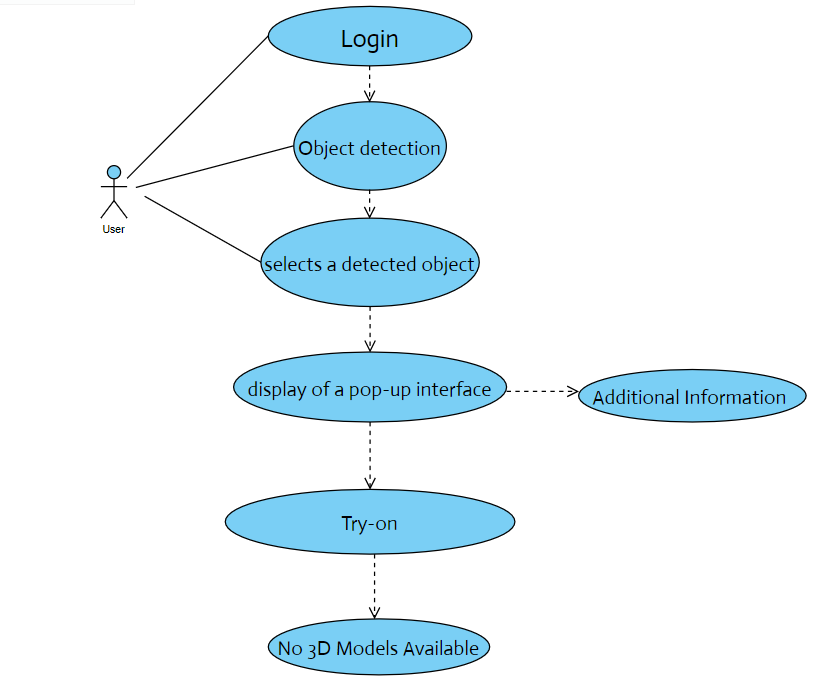


Figure 8 Use Case Diagram for Try-On Products (Non Availability of 3D-Model)

**Try-On Products Use Case for Device Incompatibility**

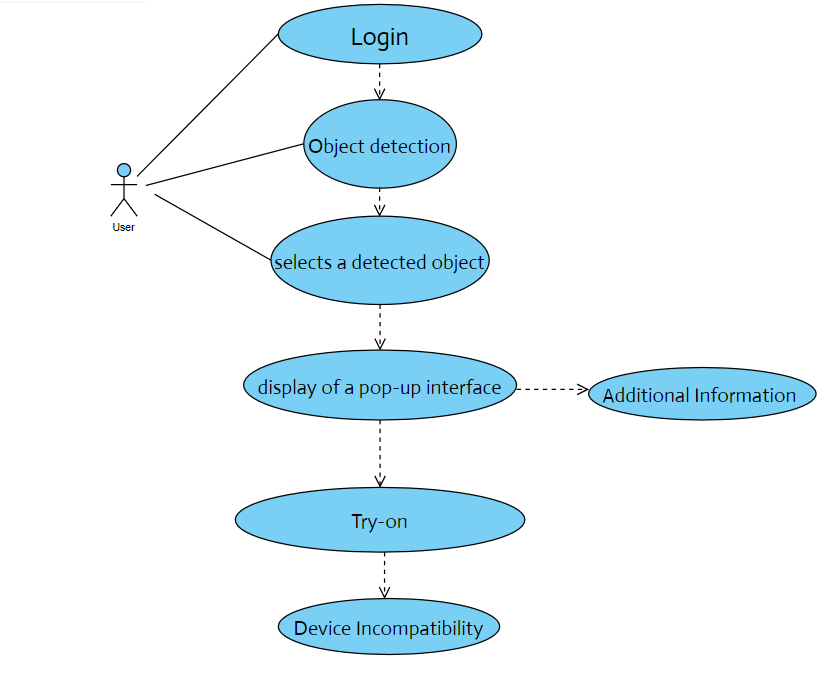


Figure 9 Use Case Diagram for Try-On Products (Device Incompatibility)

**Try-On Products Use Case for User Exits**

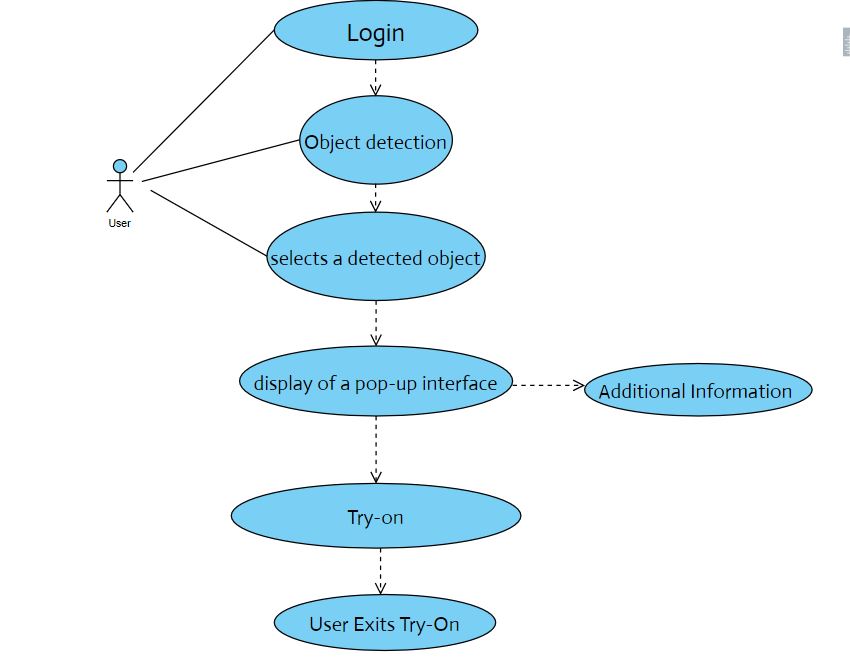
**

Figure 10 Use Case Diagram for Try-On Products (User Exits)

###### 3.4.2 Try-On Product Use Case Detail Table

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC\_04 () | |
| Use Case Name | Try-on products | |
| Description | The user clicks on try-on button in section of showed products. The user is virtually tried on product in real time. | |
| Primary Actor | User | |
| Secondary Actor | None | |
| Pre-Condition | The user clicks on try on button. | |
| Post-Condition | The user is tried on the selected object. | |
| Special Requirement | The user camera must contain AR core or AR kit. The user must allow the application to use the camera. | |
| **Basic Flow** | | |
| **Actor Action** | | **System Response** |
| **1** The user clicks on try-on button | | **2** The user is tried on the selected object. |
| **Alternative Flow** | | |
| **Actor Action** | | **System Response** |
| **3** The user clicks on try-on button. | | **4** No 3-D model is available for selected object.  The system redirected to object selection  page. |
| **5** The user clicks on try-on button. | | **6** The device is incompatible for Augmented  reality feautures. The system redirected to  exit page. |
| **7** The user clicks on try-on button. | | **8** The user click the exit button to leave try-on  Page and redireced to object selection page. |

Table 9 Try-On Product Use Case Detail Table for Figure 8, 9 & 10

###### 3.5.1 Logout Use Case

* User interacts with the user interface and selects the "Logout" option.
* System initiates the logout process.
* If there are unsaved changes, the system may prompt the user to confirm the logout.
* User is redirected to the login screen.

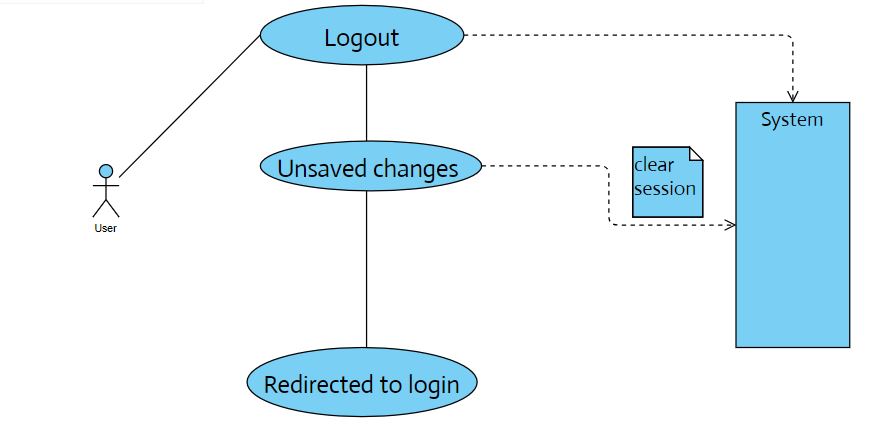
****

Figure 11Use Case Diagram for Logout

###### 3.5.2 Logout Use Case Detail Table

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC\_05 () | |
| Use Case Name | Logout | |
| Description | The user will click on logout from the menu. | |
| Primary Actor | User | |
| Secondary Actor | None | |
| Pre-Condition | The user should must be logged in beforehand. | |
| Post-Condition | The user is logged out and user is directed to login page. | |
| **Basic Flow** | | |
| **Actor Action** | | **System Response** |
| 1 The user clicks on logout button. | | 2The user is logged out and redirected to  login page. |
| **Alternative Flow** | | |
| **Actor Action** | | **System Response** |
| 3None | | 4-ANone |

Table 10 Logout Use Case Detail Table for Figure 11

# Chapter 4: Design

In this chapter, we provide the design analysis of our modules including the following designs

1. Architecture Diagram
2. ERD with data dictionary
3. Data Flow diagram
4. Class Diagram
5. Activity Diagram
6. Sequence Diagram
7. Collaboration Diagram
8. State Transition Diagram
9. Component Diagram
10. Deployment Diagram

#### Architecture Diagram

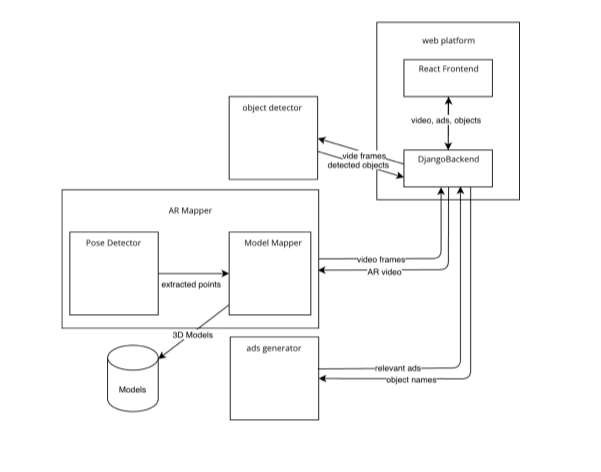


Figure 12 Architecture Diagram

#### ERD with Data Dictionary

The below figure shows the entity relationship Diagram of the database of the system:

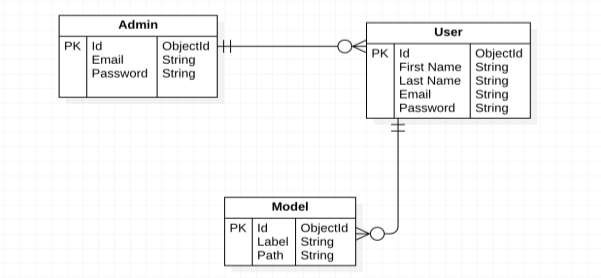


Figure 13 Entity Relationship Diagram

###### Data Dictionary

The data dictionary of the system is as follows:

###### Admin

The data dictionary of admin is as follows:

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Type** | **Example** |
| Id(PK) | ObjectId | ObjectId(“578f6fa2df35c7fbdbaed8d3”) |
| Email | String | Ali1234@gmail.com |
| Password | String | UOL@lhr |

Table 11 Admin Data Dictionary

###### User

The data dictionary for user is as follows:

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Type** | **Example** |
| Id(PK) | ObjectId | ObjectId(“578f6fa2df35c7fbdbaed8d3”) |
| First Name | String | Taha |
| Last Name | String | Munir |
| Email | String | Taha1234@gmail.com |
| Password | String | Fast@lhr |

Table 12 User Data Dictionary

###### Model

The data dictionary for model is as follows:

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Type** | **Example** |
| Id(PK) | ObjectId | ObjectId(“578f6fa2df35c7fbdbaed8d3”) |
| Label | String | Chair |
| Path | String | ../models/chair.obj |

Table 13 Model Data Dictionary.

#### Data Flow diagram

Data flow diagram includes two levels

###### The level 0

The flow of information inside the system is defined in level 0:

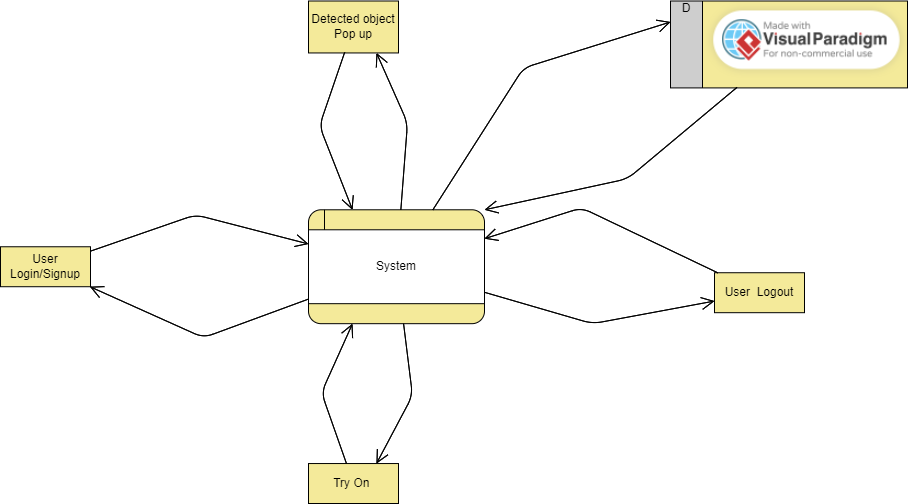


Figure 14 Level 0 Data Flow Diagram

###### The level 1

The flow of information outside the system is defined in level 1:

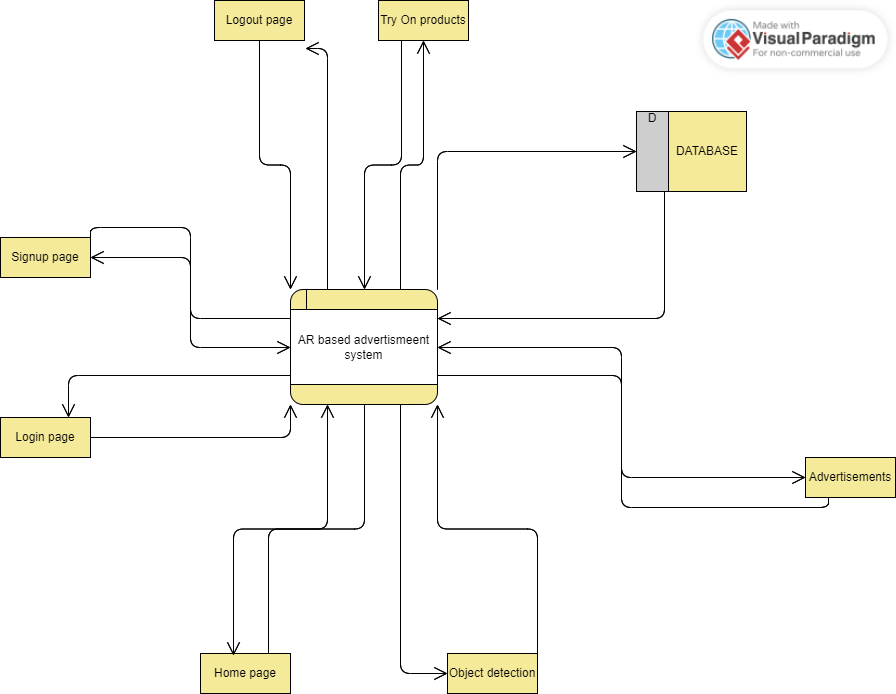


Figure 15 Level 1 Data Flow Diagram

#### Class Diagram

#### 

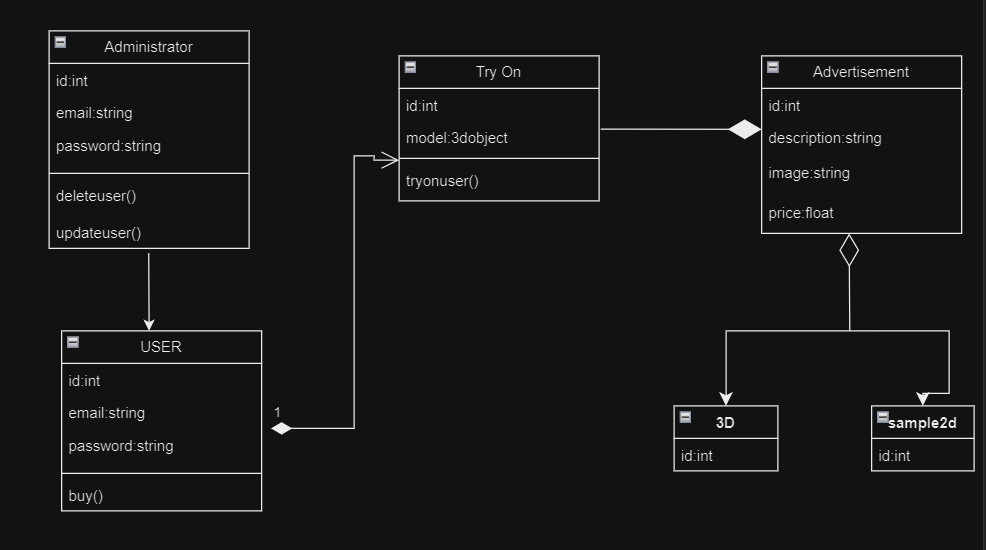


Figure 16 Class Diagram

#### Activity Diagram

###### Sign Up

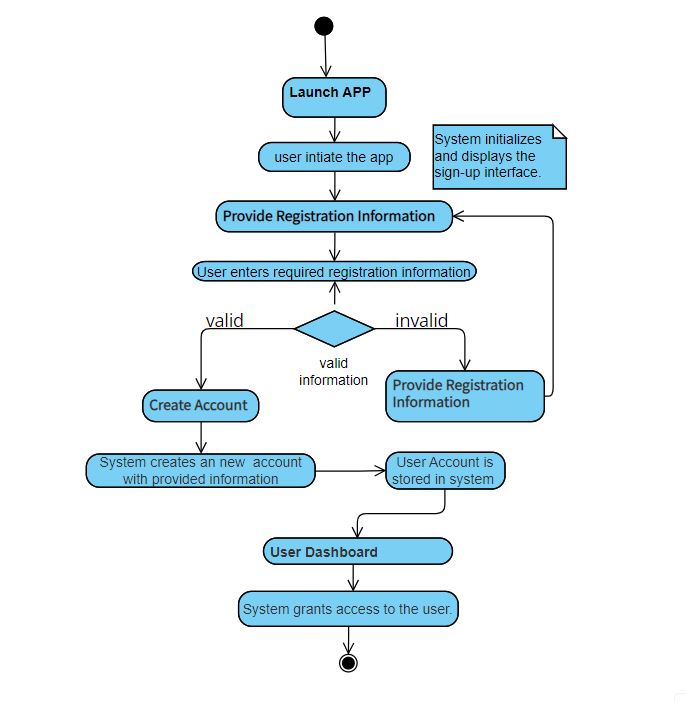
****

Figure 17Activity Diagram for Signup

###### Login

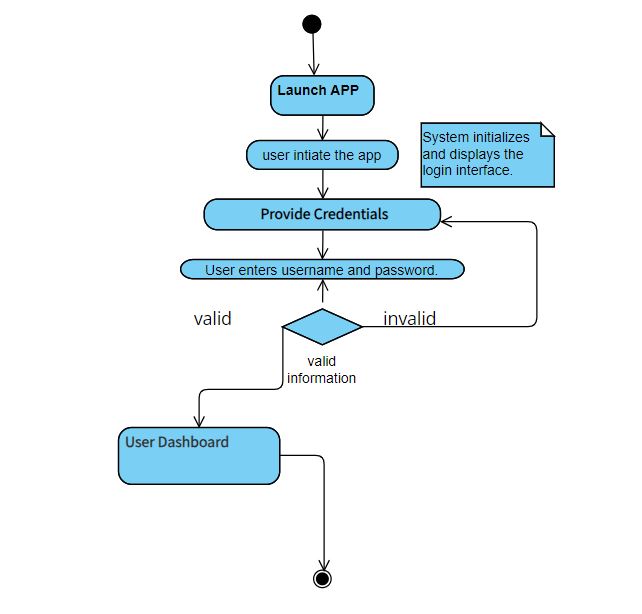
****

Figure 18 Activity Diagram for Login

###### Object Detection

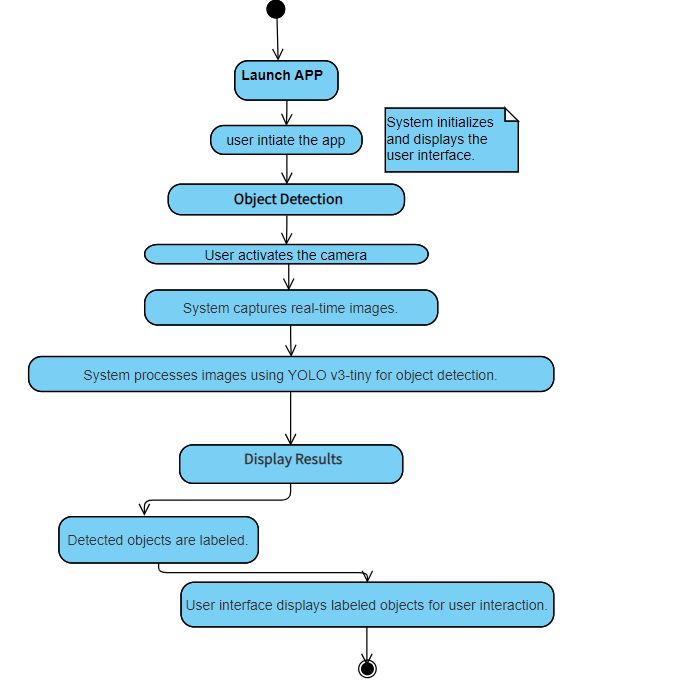
****

Figure 19 Activity Diagram for Detected Object

###### User Interaction

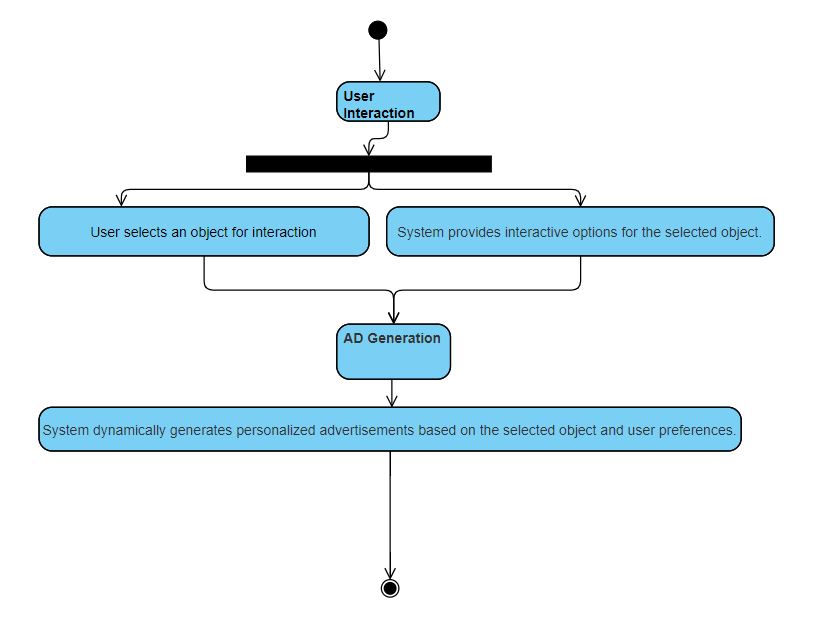
****

Figure 20 Activity Diagram for User Interaction

###### AR Visualization

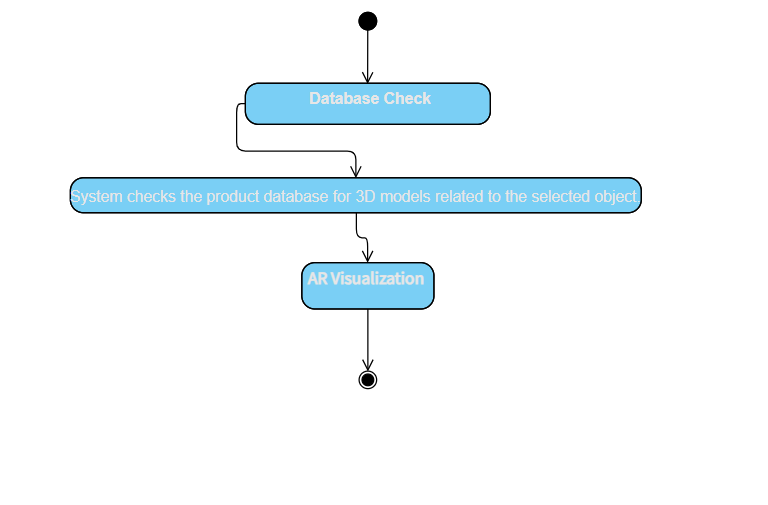


Figure 21 Activity Diagram for AR-Model for Selected Object Existence in Database

###### Log Out

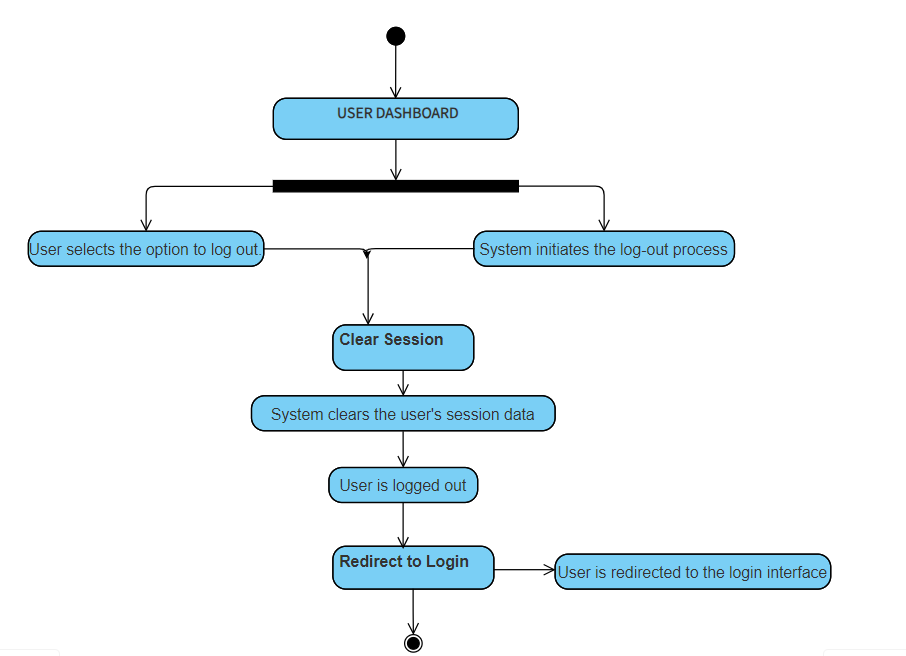


Figure 22 Activity Diagram for Logout

###### Aggregated Activity Diagram

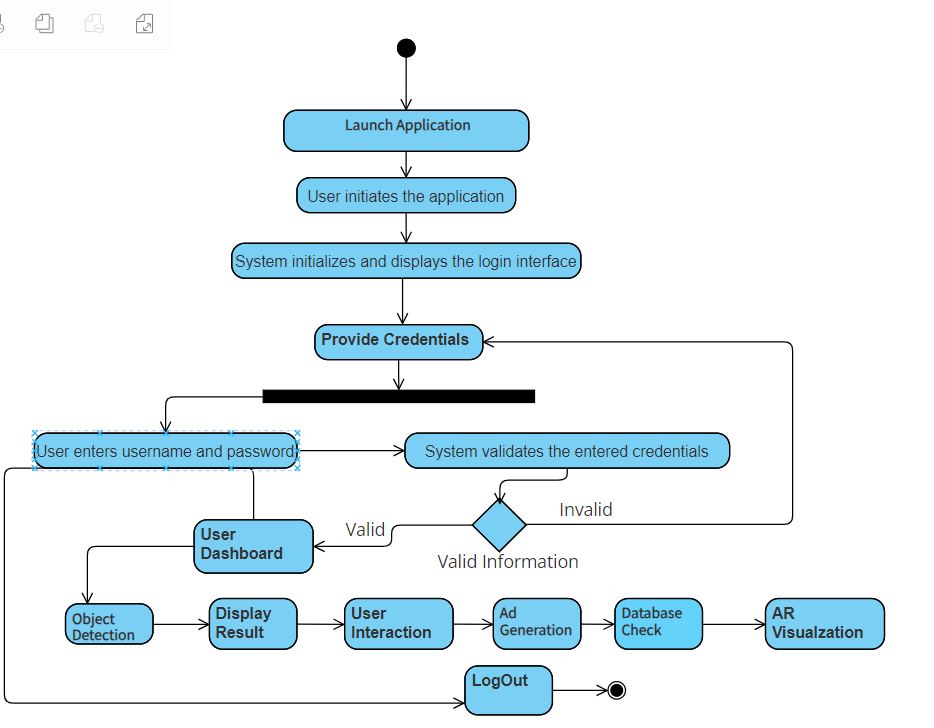
****

Figure 23 Aggregated Activity Diagram

#### Sequence Diagram

The sequence diagrams of the system are as follows:

###### User Signup

When the user wants to signup, the signup function is called:

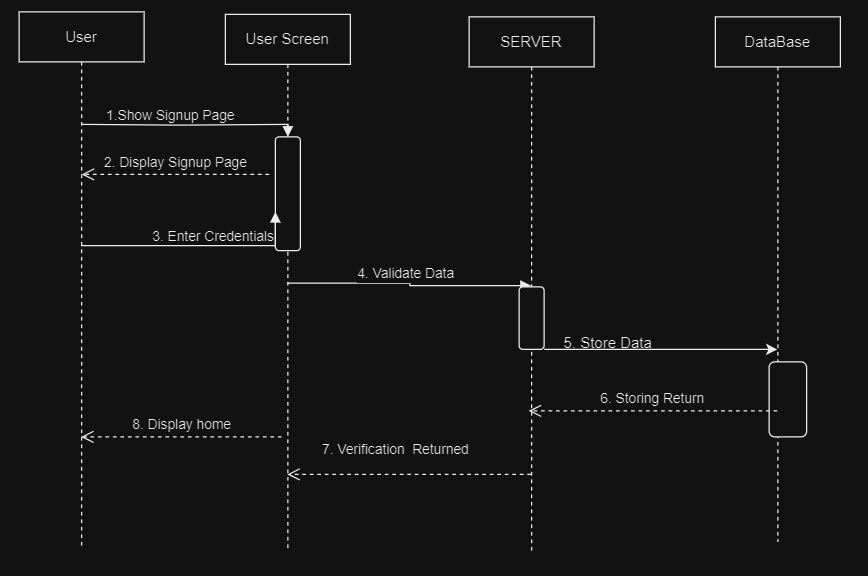
****

Figure 24 Sequence Diagram for User Signup

###### User Login

When the user logs in, the login function is called:

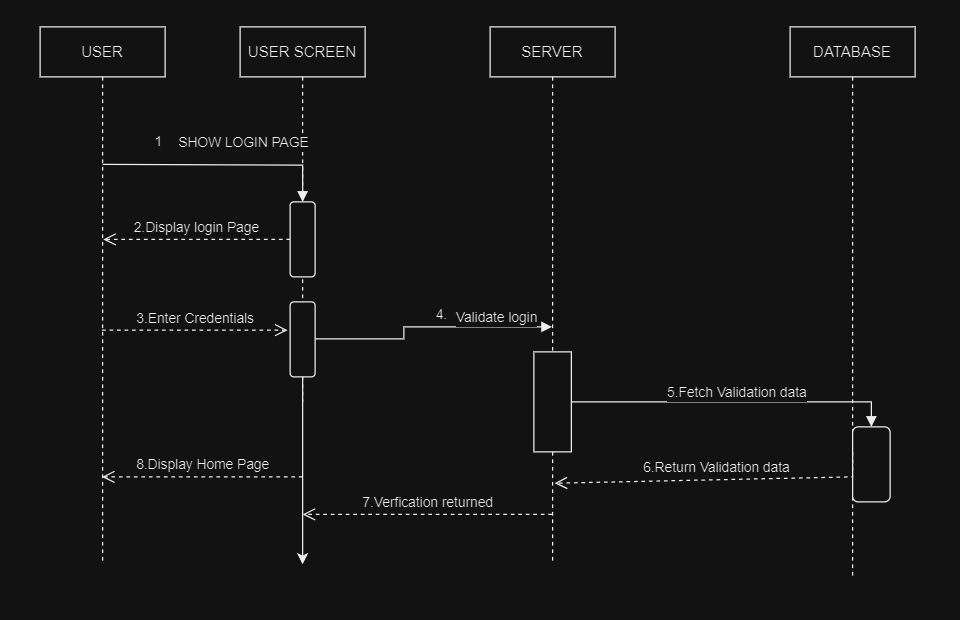


Figure 25 Sequence Diagram for User Login

###### Advertisement of Detected Objects

When the objects are detected, the related advertisements are shown:

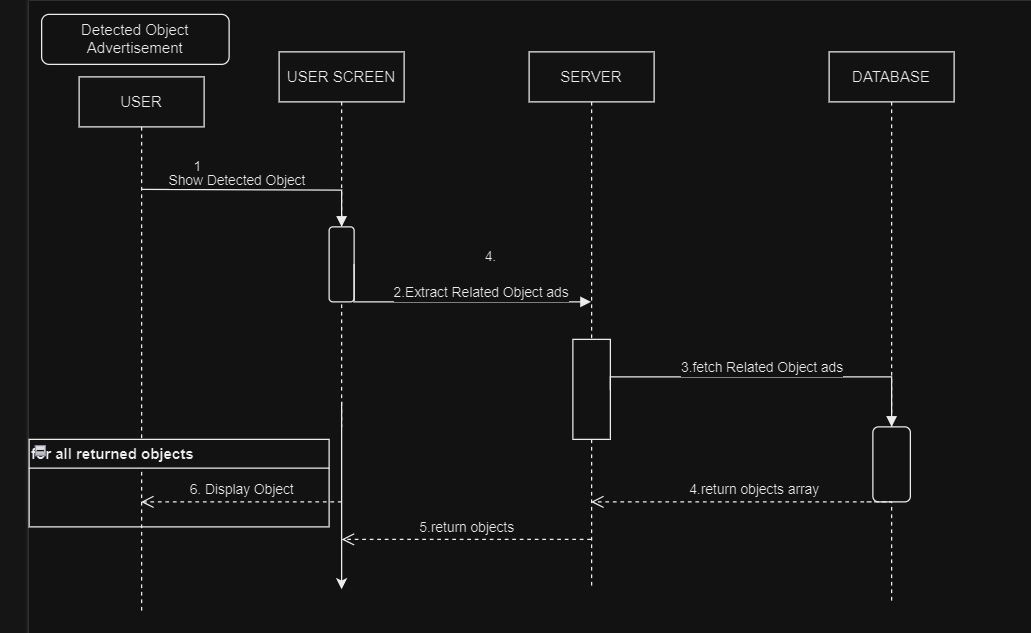


Figure 26 Sequence Diagram for Detected Object Advertisement

###### User Logout

When the user logs out, the logout function is called:

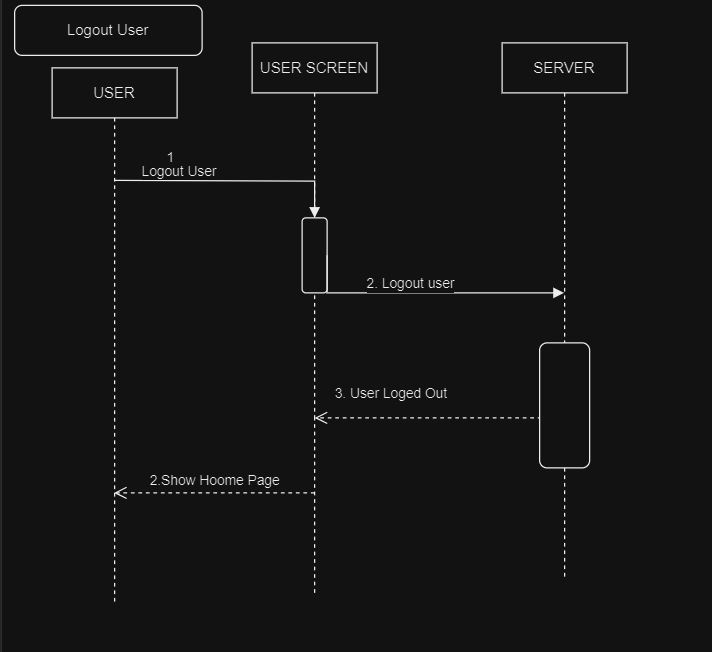


Figure 27 Sequence Diagram for User Logout

###### Admin Signup

When the admin wants to signup, the signup function is called:

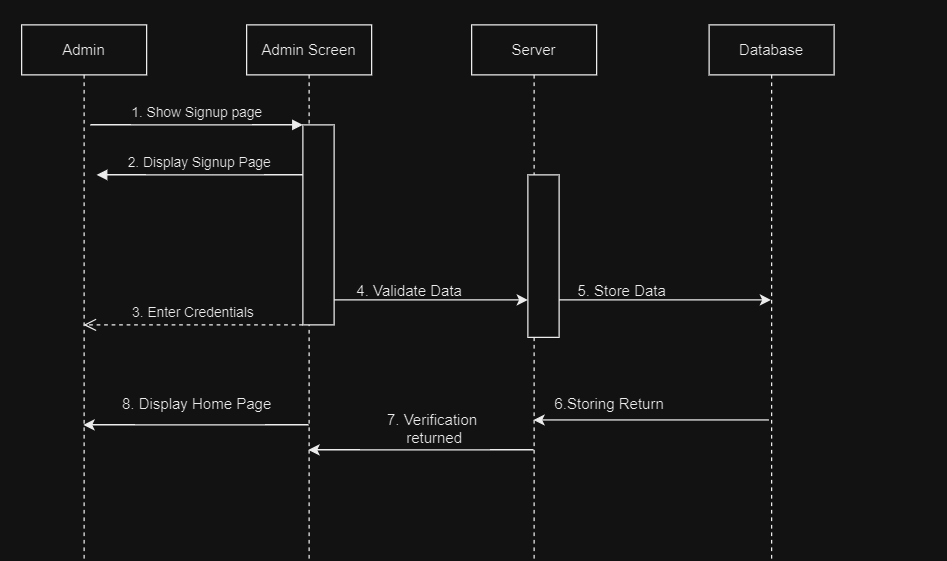


Figure 28 Sequence Diagram for Admin Signup

###### Admin Login

When the admin logs in, the login function is called:

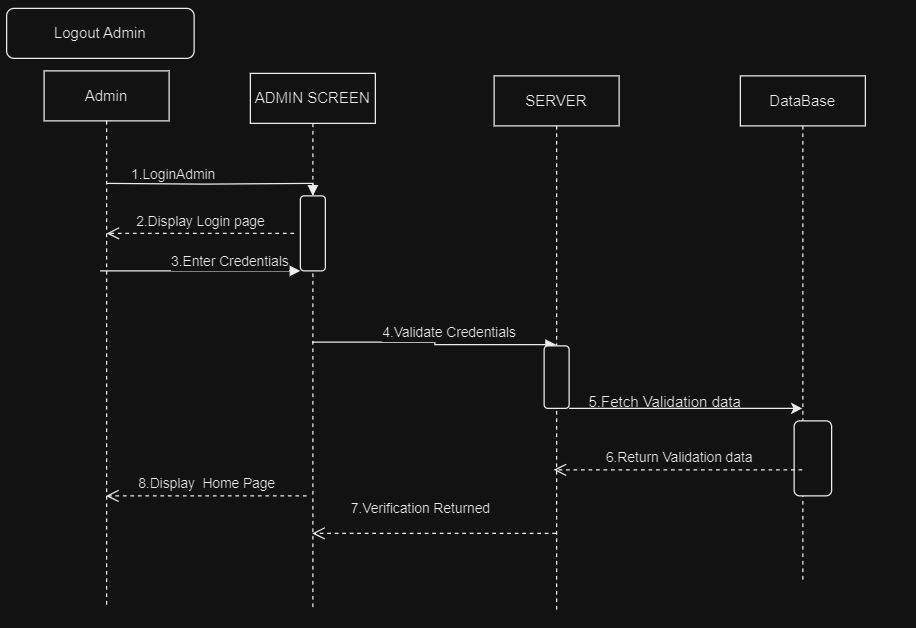
****

Figure 29 Sequence Diagram for Admin Login

###### Admin Logout

When the admin logs out, the logout function is called:

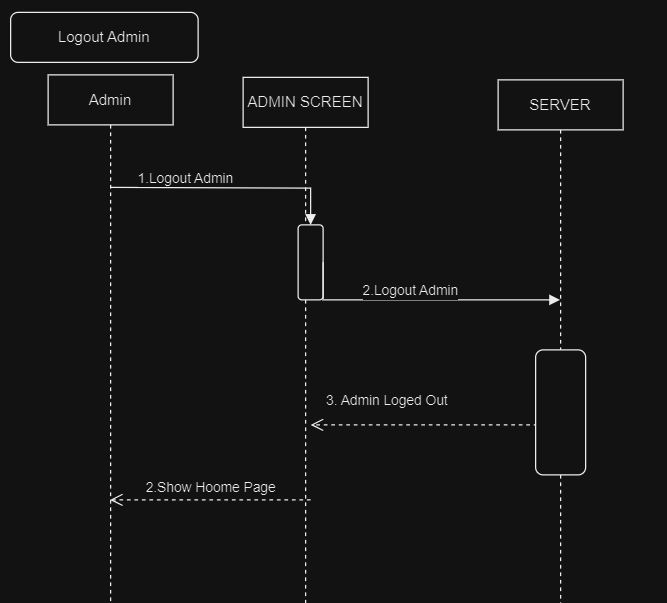


Figure 30 Sequence Diagram for Admin Logout

###### Admin Deletes a User

When the admin deletes a user, the delete user function is called:

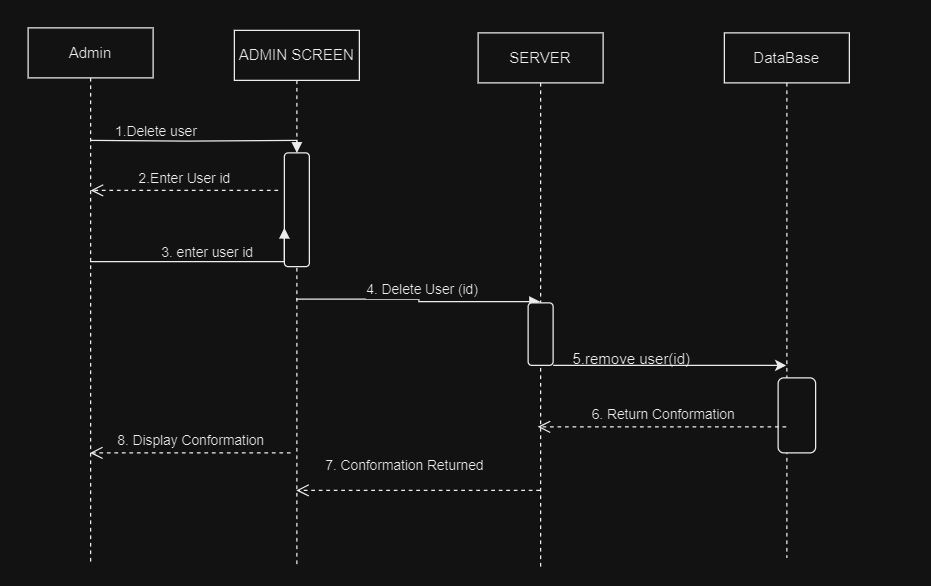


Figure 31 Sequence Diagram for Admin (Delete User)

###### Virtual Try-On Product

When the user wants to try-on a product, the virtual try on function will be called.

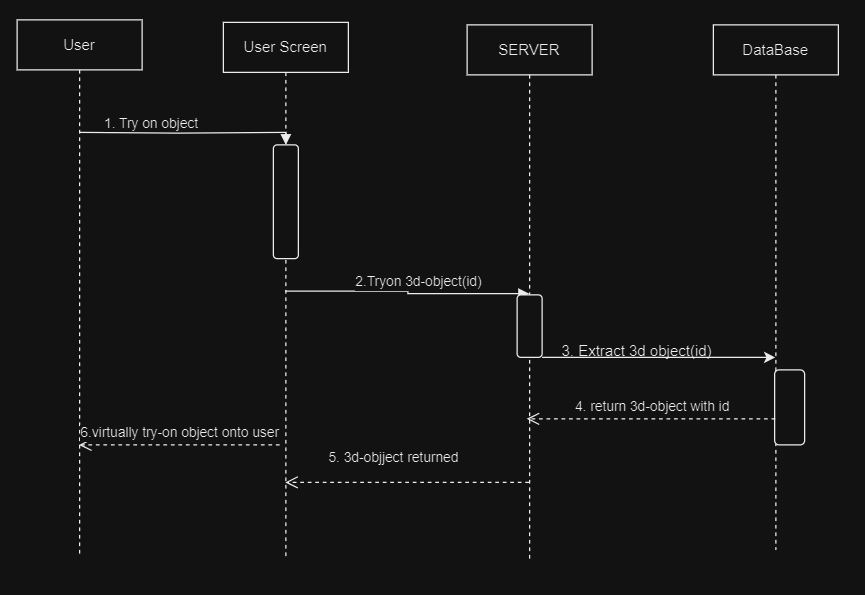


Figure 32 Sequence Diagram for Try-On Product

#### Collaboration Diagram

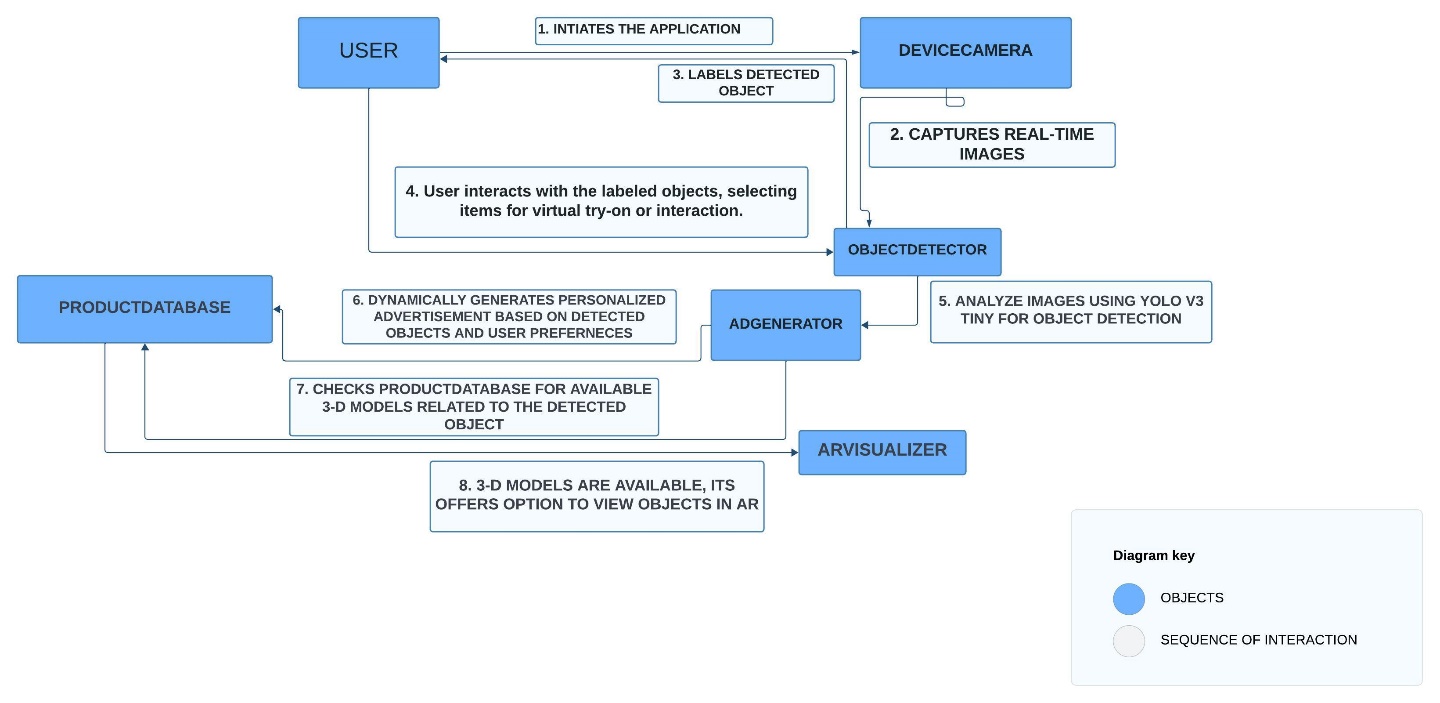


Figure 33 Collaboration Diagram

#### State Transition Diagram

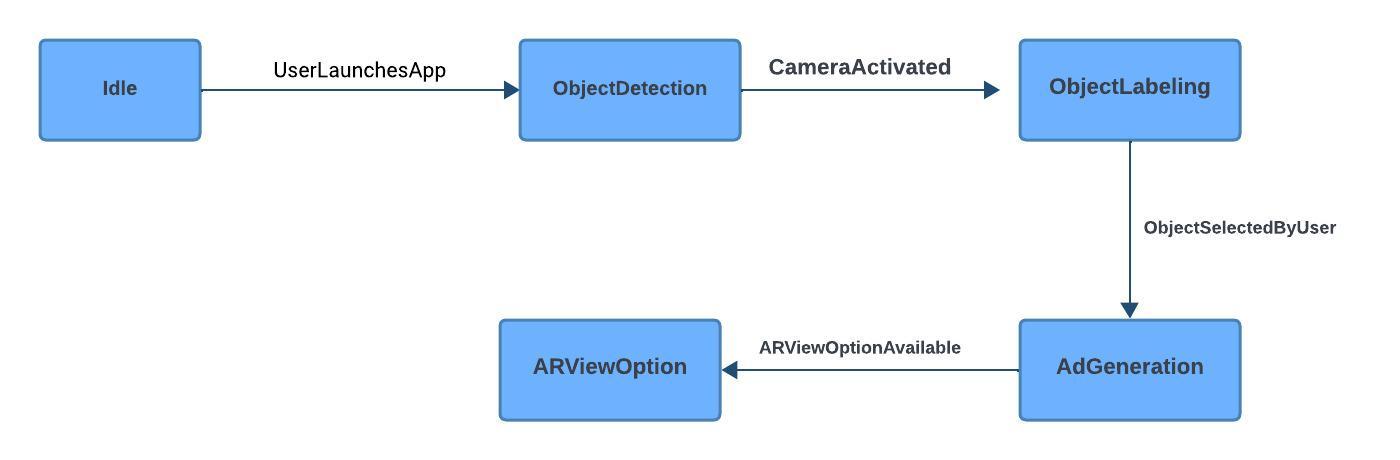


Figure 34 State Transition Diagram

#### Component Diagram

#### 

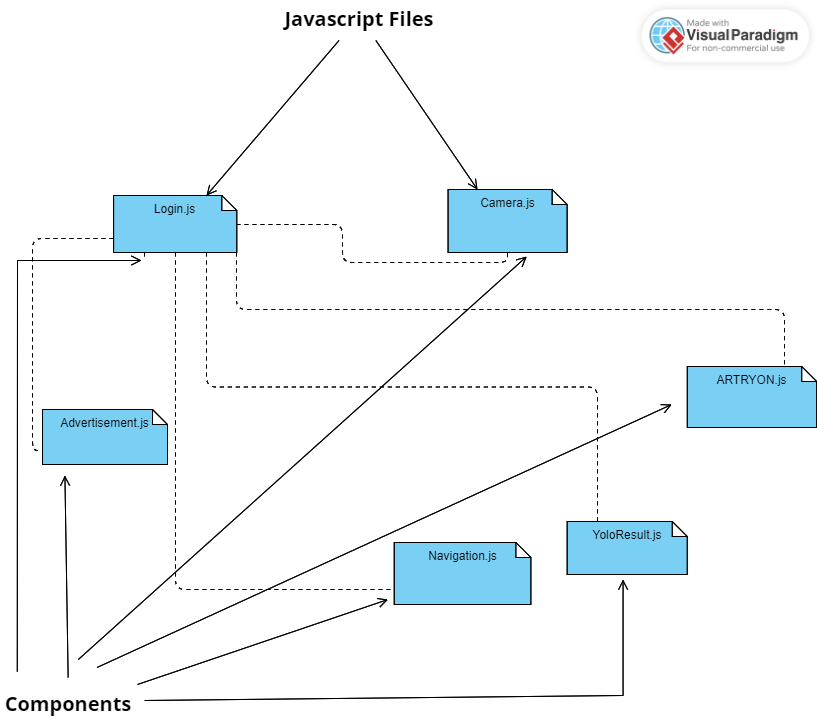


Figure 35 Component Diagram

#### Deployment Diagram

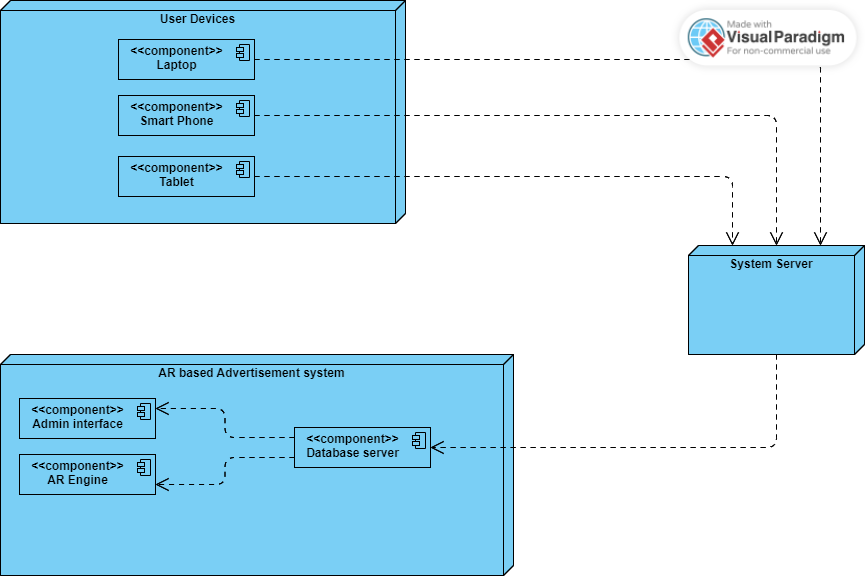


Figure 36 Deployment Diagram

# 

# Chapter 5: User Manual

Following are the **USER MANUAL** of the system to be developed:

#### Sign Up Screen

The Sign-Up screen allows users to create a new account by providing essential information.

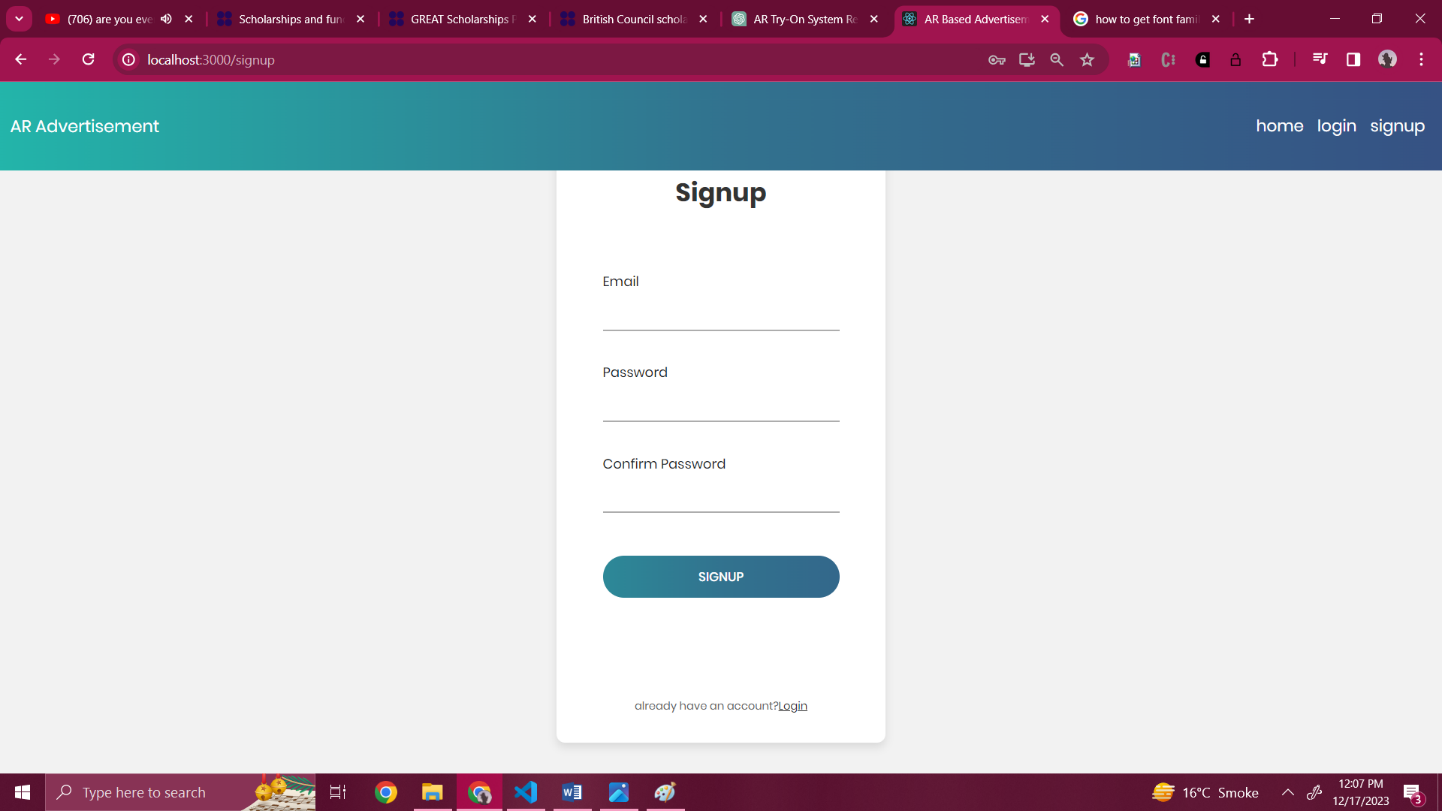
****

Figure 37 Signup Screen

#### Login Page

Users can access their accounts through the Login screen by entering their credentials.

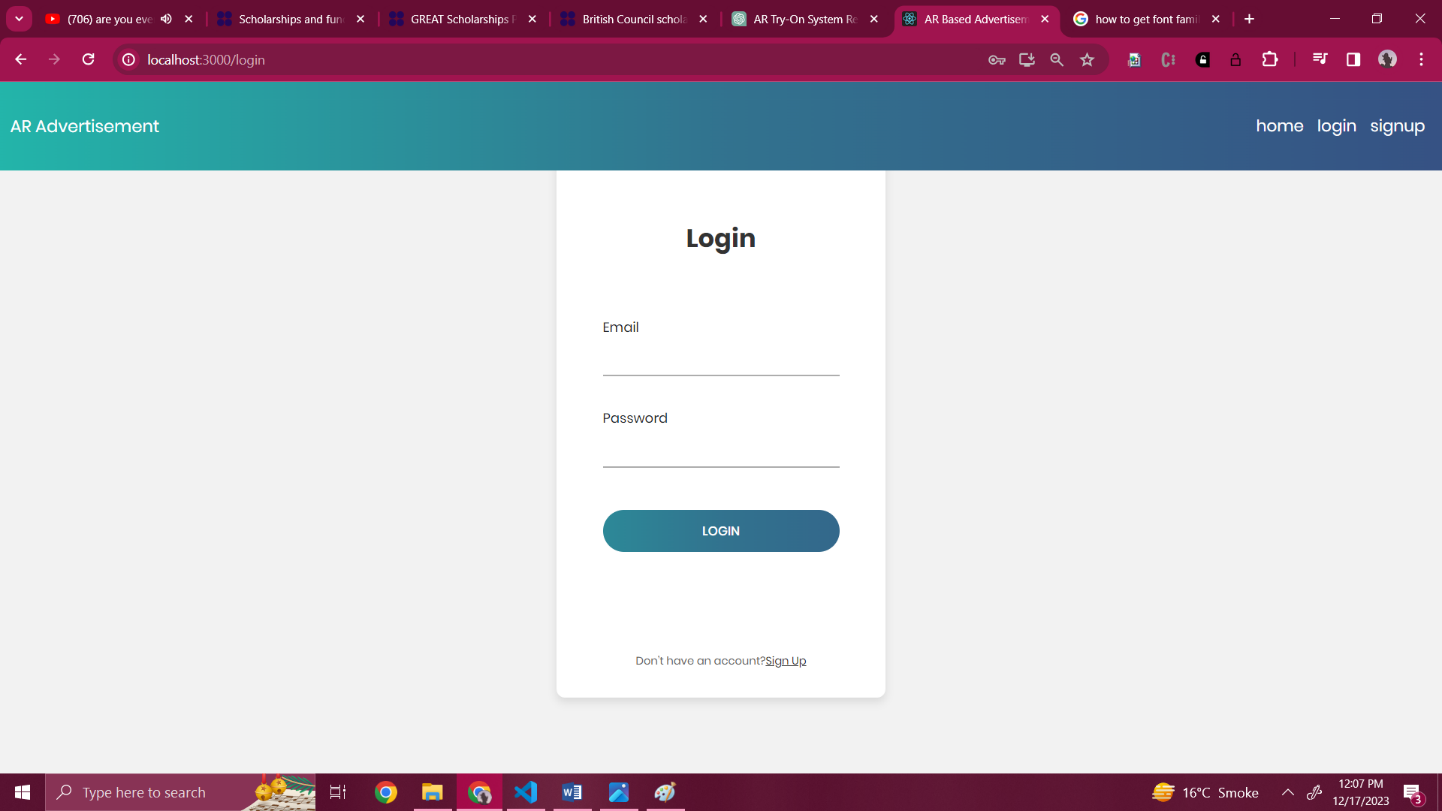
****

Figure 38 Login Screen

#### Home Page

The Home Page is the central hub where users can navigate through various features and functionalities.

****

Figure 39 Home Page Screen

#### Detected Object

The Object Detection screen showcases the system identifying and labeling objects through the device's camera.

****

Figure 40 Detected Object Screen

#### Advertisement

The Advertisement screen displays personalized ads related to detected objects, enhancing the user experience.

****

Figure 41 AR- Advertisement Screen

#### Try-on Product

Users can virtually try on different products, such as clothing and accessories, using augmented reality.

****

Figure 42 Try-On Product Screen

#### Logout Page

Users can securely log out from the application through the Logout screen, ending their current session.



Figure 43 Logout Screen (Admin & User)

# References

|  |  |
| --- | --- |
| [1] | P. Adarsh, P. Rathi and M.Kumar, "YOLO v3-Tiny: Object Detection and Recognition using one stage improved model," in *6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2020. |
| [2] | T. Ahmad, Y. Ma, M. Yahya, B. Ahmad, S.Nazir and A. .. Haq, "Object detection through modified YOLO neural network," *Scientific Programming,* vol. 2020, p. 10, 2020. |
| [3] | G. Developers, "ARCore API reference," Google Play Services for Augmented Reality, 1 March 2018. [Online]. Available: https://developers.google.com/ar/reference. [Accessed 10 October 2023]. |
| [4] | A. Developers, "Augmented Reality Kit," Apple , 01 June 2017. [Online]. Available: https://developer.apple.com/augmented-reality/arkit/. [Accessed 20 October 2023]. |
| [5] | G. Developers, "MediaPipe Holistic," Google, 13 February 2020. [Online]. Available: https://github.com/google/mediapipe. [Accessed 25 September 2023]. |
| [6] | A. Rosebrock, "OpenCV Augmented Reality (AR)," *pyimagesearch,* 04 January 2021. |
| [7] | C. Bach and M. Biilmann, "Netlify," Netlify, 07 April 2015. [Online]. Available: https://www.netlify.com. [Accessed 15 November 2023]. |
| [8] | J. Lindenbaum, A. Wiggins and O. Henry, "heroku," heroku, June 2007. [Online]. Available: https://www.heroku.com. [Accessed 20 November 2023]. |
| [9] | S. Ding, F. Long, H. Fan, L. Liu and Y. Wang, "A novel YOLOv3-tiny network for unmanned airship obstacle detection," in *IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS)*, 2019. |