SkinSage:An AI-Based Skincare Recommendation Mobile Application

ALEEHA AMIR
01-135221-005
LYBA IMTIAZ
01-135221-024

Supervisor: MS Mahwish Pervaiz

Bachelor of Science in Information Technology

Department of Computer Science
Bahria University, Islamabad

October 2025

Certificate

We accept the work contained in the report titled "SkinSage: An AI-Based Skincare Recommendation Mobile Application", written by Ms. Aleeha Amir (01-135221-005) and Ms. Lyba Imtiaz (01-135221-024) as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Information Technology.

Approved by . . . :

Supervisor:

_____

Internal Examiner:

_____

External Examiner:

_____

Project Coordinator:

_____

Head of the Department:

_____

Abstract

SkinSage addresses a critical challenge in the beauty and skincare industry, helping users find products that are personalized, effective, and suitable for their unique skin tones and types. With the rise of online shopping, users often purchase skincare items without truly understanding their skin, resulting in dissatisfaction and wasted resources. SkinSage aims to solve this by combining artificial intelligence with mobile technology to create a smart skincare recommendation platform.

SkinSage is a mobile application built using the Flutter framework, with machine learning integration for image analysis and a user-friendly questionnaire system to assess skin conditions. The application detects the user's skin tone using AI-powered image processing and provides personalized skincare routines and product suggestions based on the user's input. Features include manual shade verification, product reviews, and an admin dashboard for managing product data and user inputs. This project showcases how technology can revolutionize the skincare industry by empowering users with intelligent, reliable, and personalized beauty guidance.

Acknowledgments

We express our heartfelt gratitude and appreciation to all those who have contributed to the successful completion of the *SkinSage: An AI-Based Skincare Recommendation Mobile Application*. This project reflects collective efforts, determination, and guidance from numerous individuals without whom this journey would not have been possible.

We extend our sincere thanks to our respected supervisor, Ms. Mahwish Pervaiz, for her continuous support, encouragement, and expert advice throughout the project. We are also grateful to the participants who assisted in requirement gathering, testing, and evaluation of the application. Finally, we thank our families for their patience and unwavering support.

ALEEHA AND LYBA
Bahria University
E-8, Islamabad, Pakistan

October 2025

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

DSA	Data Structure and Algorithms
OOP	Object Oriented Programming
PF	Programming Fundamentals
SE	Software Engineering
SQL	Structured Query Language
UNESCO	United Nations Educational, Scientific and Cultural Organization
UNICODE	Unique, Universal, and Uniform Character enCoding
XML	Extensible Markup Language

Chapter 1

Introduction

The modern digital and beauty conscious society is turning to high levels of individuality in their skincare requirements[1]. Nevertheless, as e-commerce expands at an impressive rate and numerous beauty products are available on the internet, many users do not find a product that would fit their skin tone and type. This usually causes dissatisfaction, loss of money and even skin damage. Also, the classic trial and error approach or the use of customer reviews on the product in general does not consider the specifics of the skin and this makes the process of selection even more difficult.

In order to solve these problems, we suggest SkinSage a smart, AI-driven mobile app that offers the user personalized recommendations of skincare products. Through an advanced image analysis and visually guided questionnaire, the application identifies the skin tone and evaluates the skin type to recommend appropriate product and routine to the individual user. The app should be affordable, user-friendly, and inclusive and serve individuals with limited access to dermatological appointments or professional advice on using skin care.

SkinSage is set to serve as a mediator between the users and informed skin care choices through the use of modern technology and artificial intelligence. It has made the experience of skincare selection-distillation, put users in charge to learn more about their skin and make related decisions across thousands of products in the beauty industry, which is a confusing and overwhelming market. The vision is to transform the way individuals adopt skincare in the digital era by providing real-time suggestions using a smooth mobile experience with SkinSage.

1.1 Background

Today, consumers require more personalized skincare and beauty products that are being sold regrettably and it has resulted in personalization becoming the heart of the industry.

The most challenging task is to choose the most favorable skincare product on the internet because the customers have no chance to test and analyze it directly. Many are unaware of the type or the actual color of their skin that is important in the choice of beauty products. Lack of proper knowledge about skin products normally translates to the wrong choice that attracts skin issues and saves less. In addition, the cost of seeing dermatologists is typically expensive as well as time-consuming and laborious and hence not all users are able to use them. Due to such a gap, SkinSage has introduced an AI mobile application that is able to recognize your skin color and scan to identify which type of skin you have based on pictures and questions. Due to this measure, the user gets the right suggestions that will be perfect to him or her, and will strengthen their routine and make him or her confident to purchase the right products.

1.2 Problem Description

Individuals whose skin is unique cannot find online skincare products that suit their skin color and type. The absence of guidance normally causes erroneous purchases, scaled back skin, discarded products and depressed consumers. Most rely on trial and error, unreliable reviews of beauty products or dermatologist interviews that they would barely afford. The traditional skin care provides the same advice to all people that is not applicable to the skin needs of individuals. Many of the users also do not know how to find out the type and tone of their skin and this may be challenging. SkinSage is addressing these problems by combining the image viewing aspect of AI with a basic skincare quiz to recommend the most suitable products to you. Nevertheless, the implementation of this practice may have challenges on the way of gaining the trust of users in the assistance of AI, the discovery of enough and quality images and the availability of all devices to cope with the modification.

1.3 Objective

SkinSage is a mobile application that plans to utilize AI in assisting you with selecting the right skincare products in real time. It is the hope that it can identify the skin tone of a person by a photo and examine the whole picture visually using a questionnaire. Based on the inputs, it will make a recommendation to the user of the skincare products and routines that fit their own skin condition. Skinsage offers an alternative to everyone who needs to purchase skincare items online with the help of AI and a convenient design that is easy to use, affordable, and reliable enough

1.4 Project Scope

Purpose behind SkinsAge smartphone application

- Make a mobile app allowing users to receive individual guidance on skincare.
- It is possible to analyze photos with the help of AI and determine the skin tone.
- Recommend skincare products and routines based on skin tone and type[2].
- Enhance the process of purchasing cosmetics on the Internet
- Provide an affordable and simple solution for managing skincare concerns.

Project Features

- System helps to detect the colour of the skin of a certain person using the pixels that are in the image.
- A Visual questionnaire which enables the user to utilize multiple choice questions when it comes to skin analysis.
- Instant recommendations for skincare products.
- A wide range of commonly used skincare and beauty products.

User-Friendly Design

- Simple, user-friendly interface that is suitable for all users.
- Image-assisted questionnaire for better understanding[1, 3].
- Simple navigation through the site and knowing the available products.

Administrator Features

- The platform enables you to recommend products to the users.
- Add products, manage products, and manage orders.
- Maintain databases for different skin tone categories.
- Use the Agile development method for iterative improvement.

Main Goals

- Provide an AI-powered skincare recommendation application[4, 5].
- Help users make well-informed decisions when purchasing skincare products.
- Find a solution to seal the gap in offering personalized skincare guidance.

Chapter 2

Literature Review

Using AI on dermatology items on your skin has significantly enhanced the delivery of personalized care through the use of smartphones [6, 7]. The application of AI in cosmetic dermatology has several purposes including the skin observation, identification of the right skin tone. Applications such as SCAI indicate that AI can identify skin changes in real-time that can simplify skin surveillance.

However, it is evident according to reviews that a lot of AI skincare recommendation applications are not appropriately vetted, fail to offer high-quality information[8, 9] and are not acknowledged by regulators which might pose an immediate threat to patients[10]. In the investigation of biases in skin tone recognition, the application of AI-based clustering techniques has been used to produce almost flawless accuracy in skin tone recognition and make AI more inclusive.

This project recommends personalized skincare based on your skin type[3, 4] and your preferences have also come in handy. With all these new technologies, data privacy issues are still present, as well as biases in algorithms and setting of standards of regular professionals. Addressing these issues, SkinSage relies on AI to provide each user with a personalized skincare approach and care on the principles of honesty, tolerance, and the needs of users, offering many of them an opportunity to experience dermatological care.

2.1 Comparative Study

Table 2.1 presents a comparison of existing skincare analysis applications with the proposed SkinSage system.

Table 2.1: Comparative Study of Existing Systems

Feature	**Skin Bliss**	**Right Foundation for You**	**Perfect Skin Tone Analyzer**	**SkinSage (Proposed)**
Skin Tone Detection	Limited	Yes (AI-based)	Yes (Color-based)	Yes (AI-based)
Skin Type Analysis	Basic	No	No	Yes (Visual Questionnaire)
Personalized Product Recommendations	Yes	Yes (Foundation-focused)	No	Yes (Brand-specific)
Real-Time Image Processing	No	Yes	Yes	Yes
Product Diversity	Multi-brand	Single brand	N/A	Single brand (focused)
UI/UX Design	Moderate	Basic	Low	High (Inclusive UI)
Target Users	General skincare users	Foundation shoppers	Tone-matching users	All age groups

2.2 Current System Research

2.2.1 Skin Bliss

Skin Bliss is among the popular AI-based skincare applications on Google Play Store. It provides customized skincare routine by assessing more than 100,000 products and ingredients within them. It then identifies skin problems such as acne, dryness or uneven tone and prescribes the necessary products using facial recognition technology. None of the direct e-commerce features. This application is added to the field with its inclusion of external factors in the analysis of skin health but does not incorporate shopping.[11]

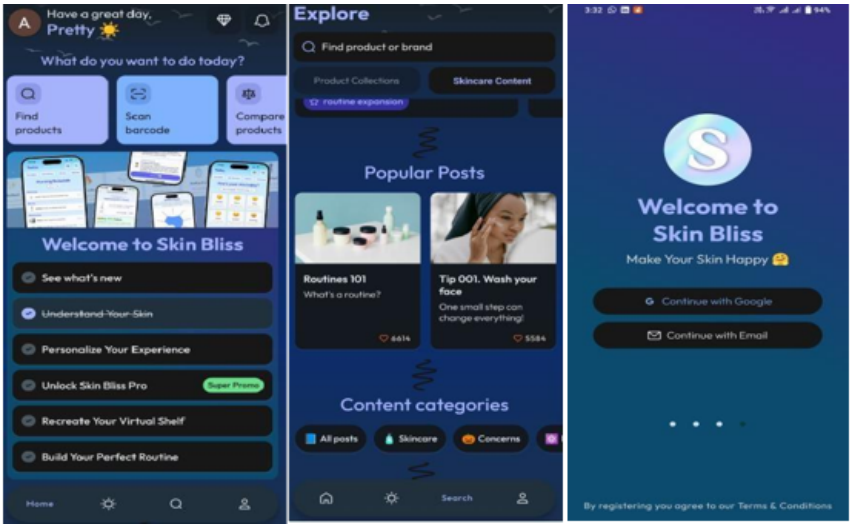

Figure 2.1: Skin Bliss Interface

2.2.2 Perfect Skin Tone Analyzer

Perfect Skin Tone Analysis provides a comprehensive AI-based analysis of skin tone differences and provides appropriate skin and makeup products to balance uneven skin tones.
This application is great at analyzing tones and does not offer a smooth-sailing shopping process. [12]

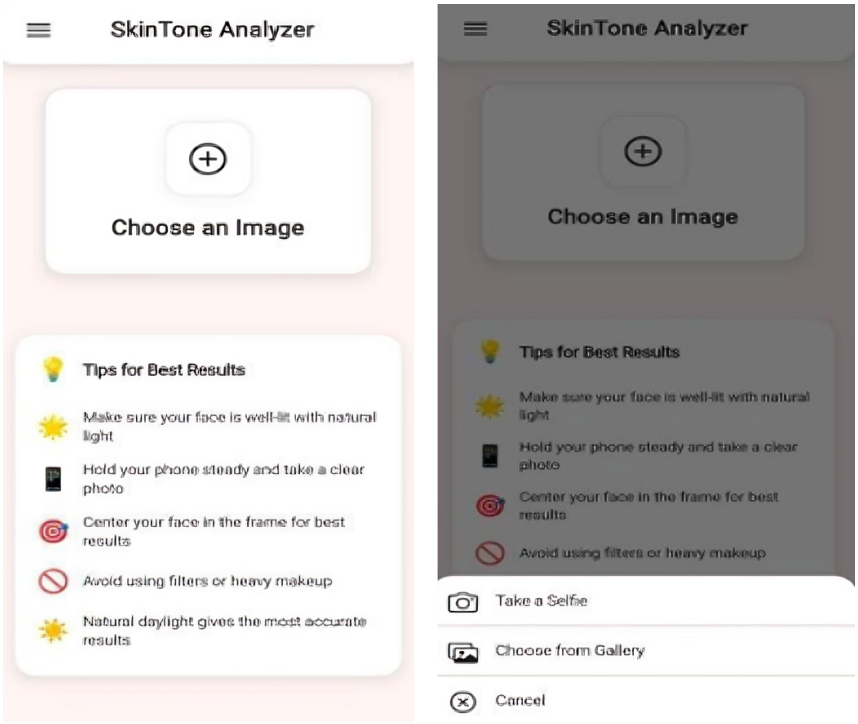

Figure 2.2: Perfect Skin Tone Analyzer Interface

2.2.3 Right Foundation for You

Right Foundation for You focuses on assisting customers in identifying the ideal foundation to use depending on their skin color and type. It does this with the help of AI color-matching technology, which gives a foundation shade match of various brands.
This was an app that was restricted to the foundation goods.[13]

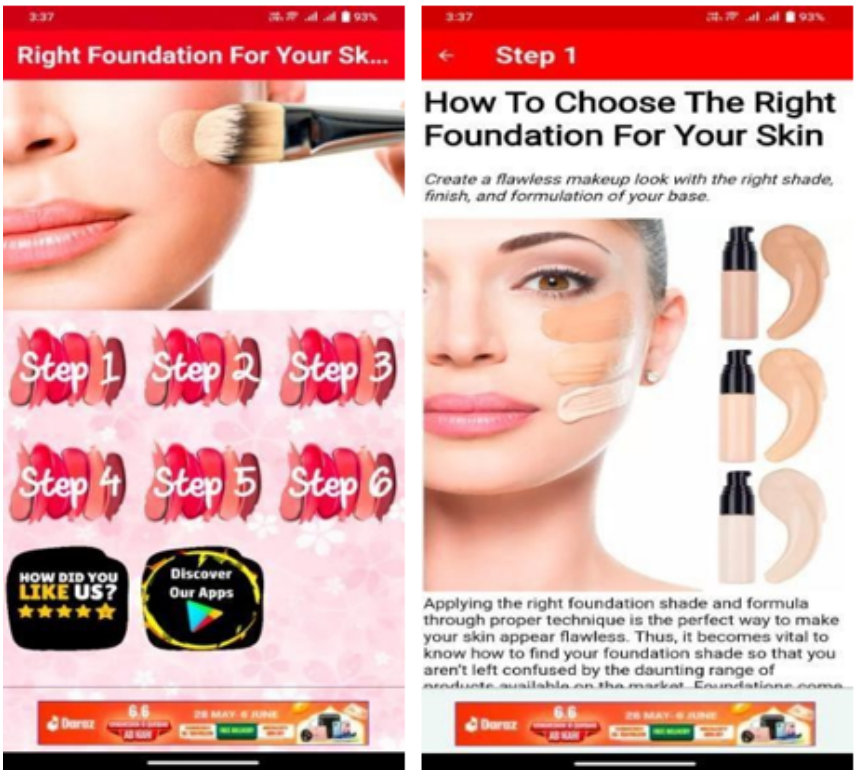

Figure 2.3: Right Foundation for You Interface

Chapter 3

Requirement Specifications

System requirements include specifications of an effective and reliable system. The result of poor implementation of such specifications can be troublesome with the system performance, its stability, or the way it can be used by the users. The following list includes the problems, potential solutions, and breakdown functions and requirements of the SkinSage mobile application.

3.1 Existing System

At this moment, individuals are forced to test products or read online reviews or visit a dermatologist to discover the appropriate skincare products. The reason why they fail to consider the skin of each individual is that such methods leave people dissatisfied with either unused products or damaged skin. The majority of the e-commerce websites do not provide individualized advice using AI and real-time analyzing of the skin type and tone.

Limitations or Drawbacks of the Existing System

Following are the limitations of the existing system:

Lack of Personalization: The majority of the platforms give general product recommendations without taking into consideration unique skin features like skin tone, type, or conditions.
Unavailability of Professional Advice: It is usually costly and unavailable to the general population to meet with dermatologists.
No Real-Time Analysis: A user has no option of analyzing their skin in real time with the help of image processing, which makes it hard to find the appropriate products.
Inefficient Product Selection: The users cannot have a tool that would enable them to

select compatible skincare products with confidence, and this leads to a product return and loss of money.
Lack of Interactive User Engagement: There is no interactive format like a questionnaire (guided) or shade selector to create trust in recommendations.

3.2 Proposed System

The app offers a new way of using the mobile phone through the use of AI to support users in their test quizzes and present them with the good selection of skincare. Artificial intelligence has been applied to this application to recognize your skin tone and the second thing, it does is by making product recommendations to your answers to the quiz. other new functions that it undertakes are the manual checking of shades, checking what customers thought of a product and sorting of the user and database activities through the administration portal.

The Proposed System Overcomes Existing System Limitations

This is how the proposed system overcomes existing system limitations:

AI-Based Skin Tone Detection: Accurate analysis of skin tone makes more accurate recommendations of the product by use of image processing.
Personalized Skincare Routines: Interactive visual questionnaires give the user the ability to self-report their skin concerns and preferences
Inclusive Design: The interface and user experience are simple and visual, and thus can be used by all types of users irrespective of their technological skills.
Manual Control: Delivering the ability to check or manually alter their identified skin shade with the help of the manual shade selection tool
Admin Dashboard: Admin is able to update the information about products, manage orders, and add new products

3.3 Requirements Specifications

These specifications determine the behaviour requirements of the application as well as limitations. They make sure that they are in line with the needs and objectives of development, enable validation, and aid in the control of scope and expectations. The requirements documented in a proper way also contribute to the future maintenance and scalability, as well as to the communication between the stakeholders.

3.4 Functional Requirements

3.4.1 FR-01: User-Side Functionalities

- **User Registration:** The users are allowed to create a new account by email and password.
- **Login/Logout:** The users are allowed to log into the application and log out safely
- **Upload Face Image:** Users are allowed to post a facial photograph on which an AI can identify the skin tone.
- **Take Skin Questionnaire:** The users receive a multiple-choice quiz that is displayed visually to answer and receive a personalized skincare assessment.
- **Manual Shade Selection:** The users can choose or change manually the shade of their skin tone.
- **View Product Details:** The users can read the description of the each of the recommended products, brand.
- **Forgot Password:** The users are given an opportunity to get the password reset through email.

3.4.2 FR-02: Admin Functionalities

- **Admin Login:** Administrators can safely log in to backend features
- **Add/Edit/Delete Products:** Admins are able to handle the skincare product database
- **Manage orders:** The admin can view, track, and update customer orders to ensure smooth order processing.

3.5 Non-Functional Requirements

3.5.1 Accessibility

SkinSage application is available 24/7/365 to the users wherever they go. Because the system is developed based on Firebase and cloud-based technologies, it can maintain 24/7 global coverage. The app also accommodates users with varied network conditions by loading important UI elements even with slow connections. Another aspect of accessibility is the readable fonts, sufficient color contrast and using the simple assistive technologies to enhance usability.

3.5.2 Compatibility

The app is compatible with Android and iOs. It must be responsive in terms of resolutions and screen sizes with Flutter responsive widgets. The interface should adapt automatically to tablets, devices with different aspect ratios, and mobile phones. The application is compatible with the latest stable OS and work on at least two older versions of Android /iOS so that it has a greater scope of compatibility.

3.5.3 Security

The system has powerful security measures to protect data and allow privacy to the users. Firebase storage and Firestore have sensitive information like user profiles, skin images and suggestions encrypted. Firebase Authentication assists in unauthorized access, whereas Firestore rule-based security limits the access to read/write depending on the role of a user (admin vs. user). Despite the fact that the system is not based on SQL, design guarantees protection against the prevalent vulnerabilities including XSS, insecure direct object access, broken authentication, and data manipulation or alteration. ML model results or product databases may only be accessed using authorized processes.

3.5.4 Performance

The SkinSage application provides an excellent and touch-responsive user experience. With an average network connection, the application takes up to 5 seconds to load the main dashboard. Important functions like uploading pictures, determining skin tones, and using recommendations need to take 10 seconds or less. The ML model is trained to achieve high efficiency using Firebase functionality or server processing so that no long delays are experienced in the process of analyzing the skin.

3.5.5 Scalability

The backend architecture is able to manage the increasing traffic using the current number of users without reducing speed. Firebase Firestore, Storage, and Authentication are automatically scaled and able to satisfy a minimum of 1000 simultaneous users. The storage system and policies should also be effective with the growth of the database of products, pictures, and customers. This guarantees that the application can be expanded without any redesign of the core structure.

3.5.6 Maintainability

The codebase of the application is structured under the MVVM architecture, which enables developers to be able to easily update the application, debug, and include new features.

Authentication, product listing, skin detection, and recommendation logic modules are divided to enhance clarity. UI can be changed through the use of reusable Flutter components, which means that business logic is not impacted. Long-term maintainability is assisted by proper commenting, file structuring, and version control.

3.5.7 Usability

It uses user friendly interface with attractive graphics and user user-friendly to the novice. The basic flow of the application, uploading an image, filling out a questionnaire, and getting recommendations, is comprehended by a new user in 10 minutes. There are clear navigation buttons, readable labels, consistent color themes, and meaningful icons, which make the application usable. The user is instructed in each step with error messages and success messages, which makes the app user-friendly and easy to use.

3.6 Sequence Diagram

The Sequence diagram can illustrate the interaction that takes place between the user, SkinSage system, and the product database step-by-step in the product recommendation process. It depicts the operations process, including when an image is uploaded till when final product recommendations are provided, in terms of sequence and time.
The user workflow diagram of the SkinSage system is presented in Figure 3.1.

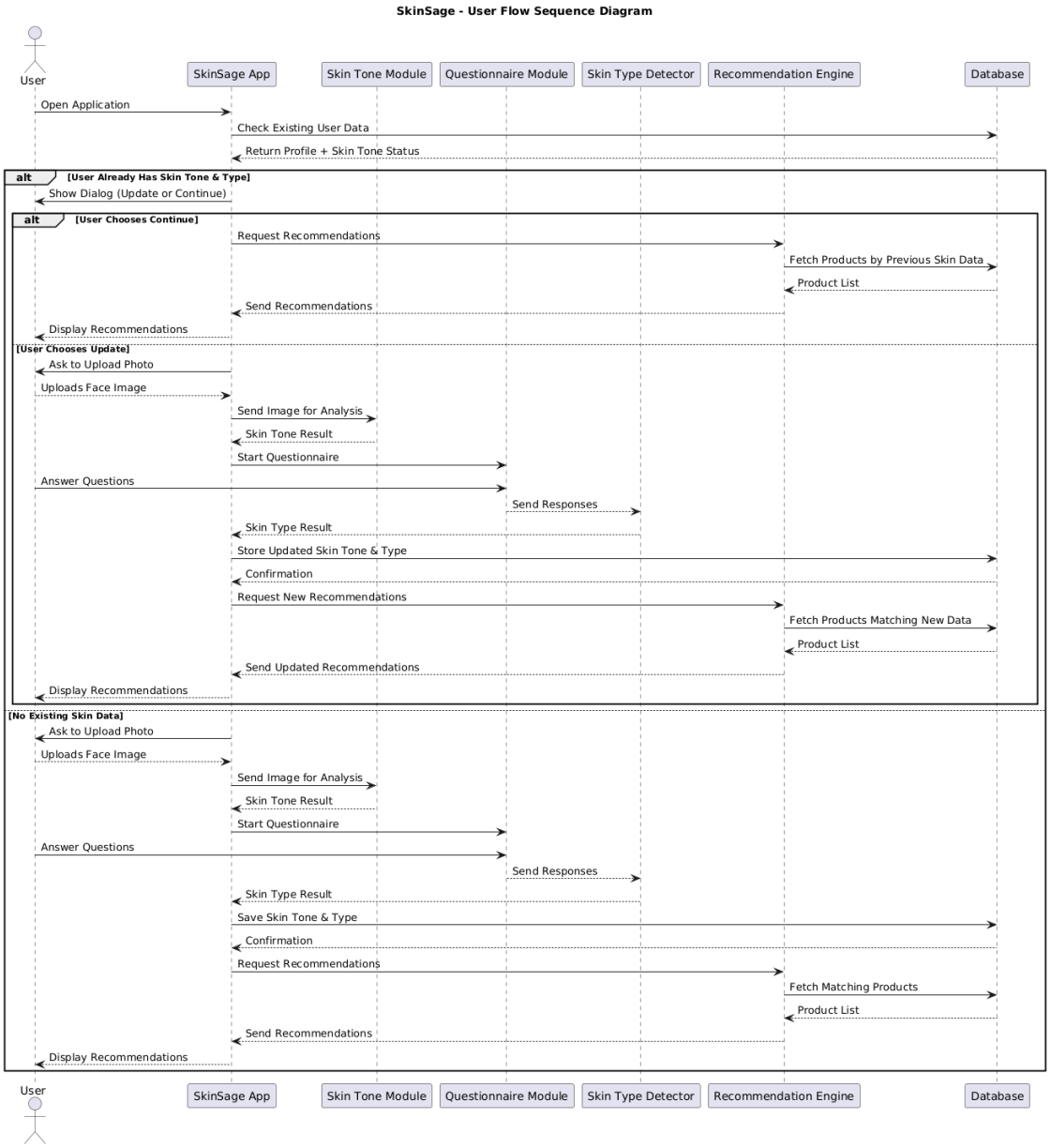

Figure 3.1: Sequence Diagram

3.7 Workflow Diagram

The Workflow diagram shows the entire line of operation of SkinSage application. It provides the description of every significant process, starting with uploading an image and filling out a questionnaire, and ending with an AI-driven analysis and suggested final products. This flow chart can be used to imagine the relationship between various processes that contribute to providing an individual skincare experience.
The user workflow diagram of the SkinSage system is presented in Figure 3.2.

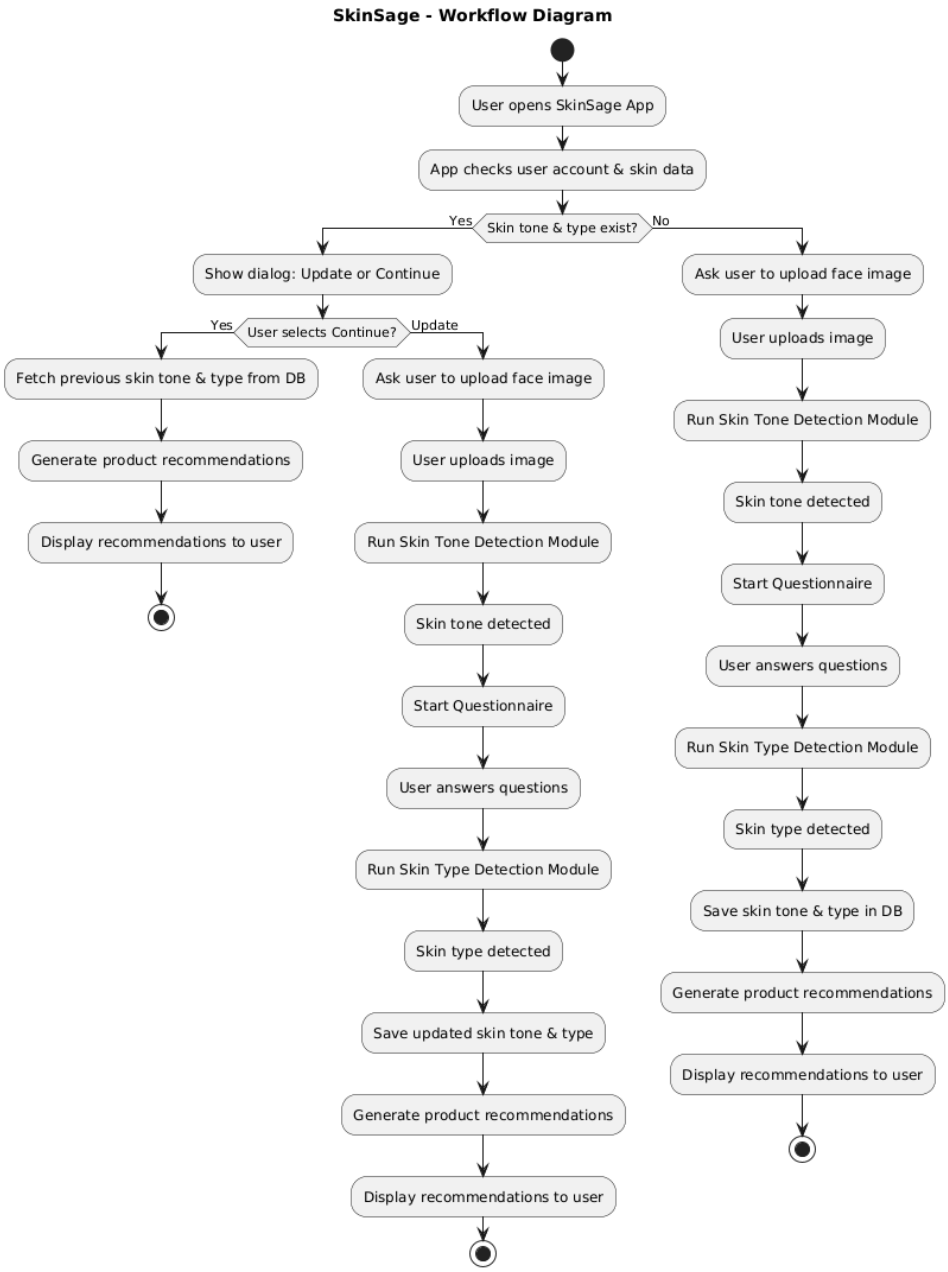

Figure 3.2: User Workflow Diagram

The admin workflow diagram of the SkinSage system is presented in Figure 3.3.

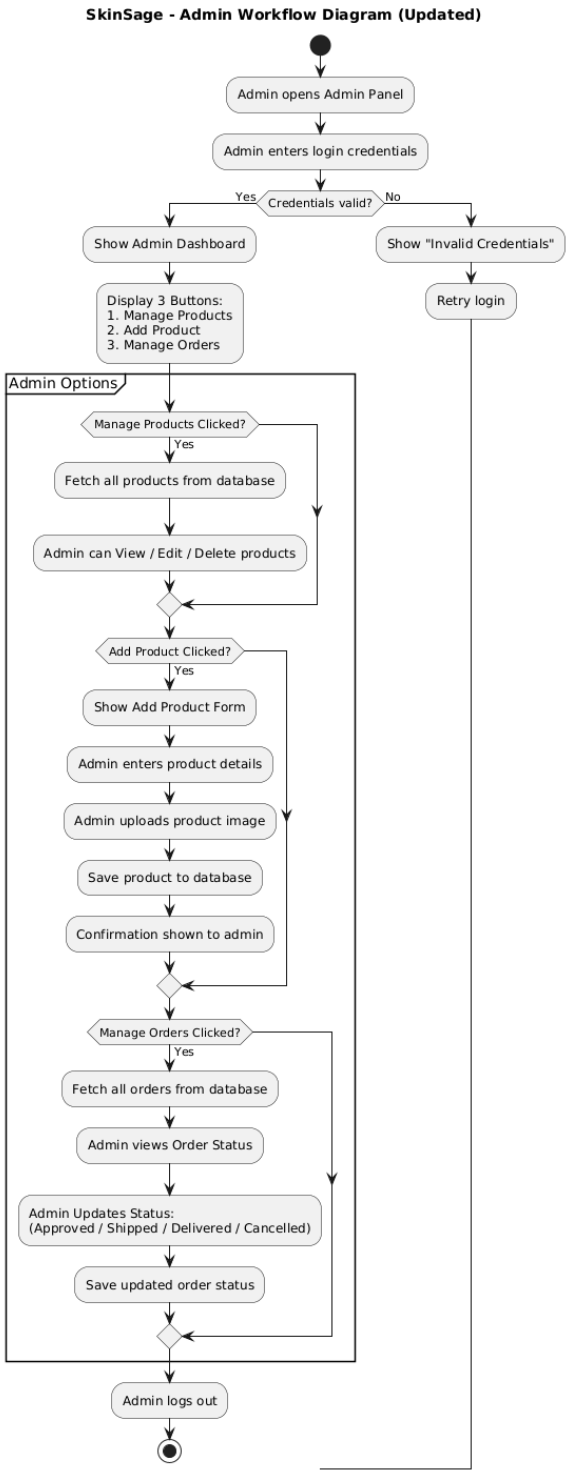

Figure 3.3: Admin Workflow Diagram

3.8 Data Flow Diagram (DFD)

Data flow diagram is a depiction of how the data passes down the SkinSage system. It depicts the communication among external parties, system functions, and data repositories to be utilized in the skin tone analysis, questionnaire outcome and product matching. This diagram gives a systematic perspective of the flow of information in the application.
The data-flow-diagram of the SkinSage system is presented in Figure 3.4.

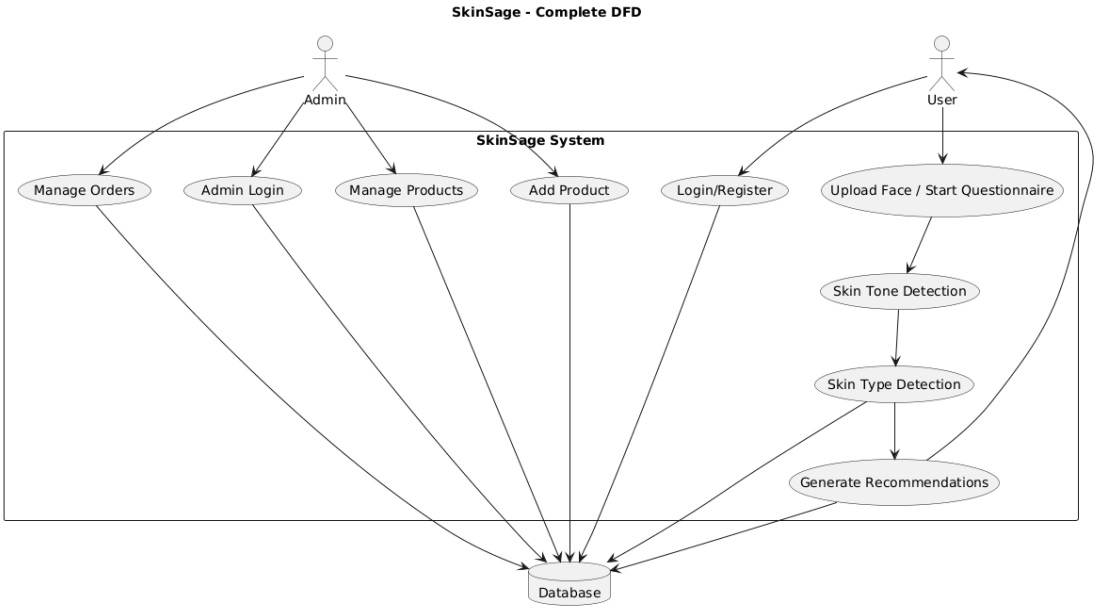

Figure 3.4: Data Flow Diagram

3.9 Use Case Diagram

Use Case Diagram is to illustrate user interaction with the SkinSage system. It also highlights functions such as skin tone detection, filling out questionnaires, and tailored product recommendations. This diagram provides an overview that captures how the system responds to different user's inputs. This allows us to clearly identify what exactly the user and system need to be and how these communicate with each other. This way, the developers get a clear understanding of what the system is able to do and check if it covers all the features that he needed. It also assists in communicating within the team by align members with a collective understanding of how the system behaves externally. In summary, the use case diagram builds a critical base that can be employed for the analysis and design, and testing of core functionalities of the application SkinSage.
The usecase-diagram of the SkinSage system is presented in Figure 3.5.

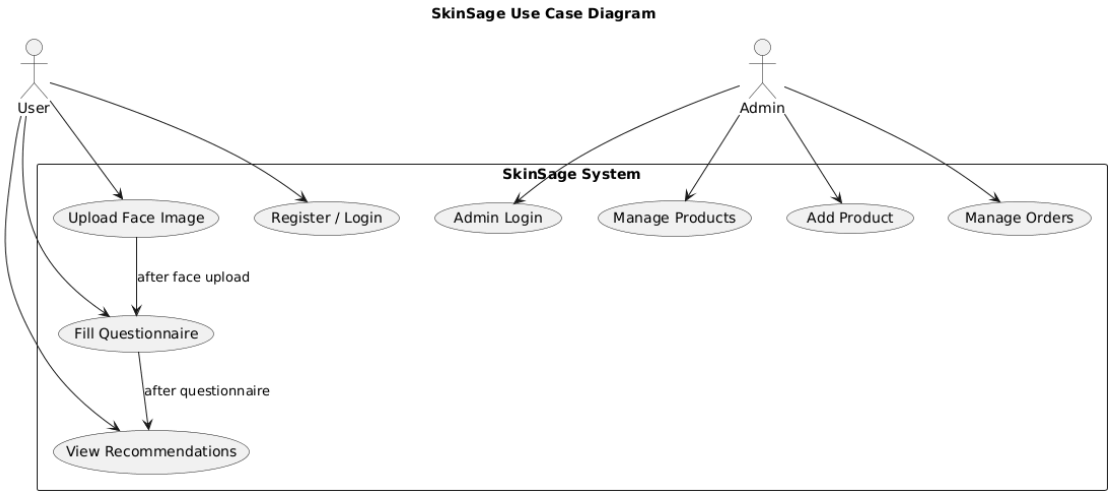

Figure 3.5: Use Case Diagram

Table 3.1: Use Case: Login/Logout

Use Case ID	UC-SS-1.1
Title	Login/Logout
Actor	User, System
Pre-condition	User must have a registered account.
Post-condition	User is successfully logged in or logged out.
Basic Flow	**User:** Enters login credentials / clicks logout. **System:** Validates credentials and logs user in / clears session.
Alternate Flow	**User:** Enters invalid credentials. **System:** Displays error message and denies access.

Table 3.2: Use Case: Register

Use Case ID	UC-SS-1.2
Title	Register
Actor	User, System
Pre-condition	User is not already registered.
Post-condition	New user account is created.
Basic Flow	**User:** Fills in registration form. **System:** Validates data and creates account.
Alternate Flow	**User:** Leaves required fields empty. **System:** Prompts for missing information.

Table 3.3: Use Case: Forgot Password

Use Case ID	UC-SS-1.3
Title	Forgot Password
Actor	User, System
Pre-condition	User must have a registered email.
Post-condition	Password reset link is sent to user's email.
Basic Flow	**User:** Clicks forgot password, enters email. **System:** Verifies email and sends reset link.
Alternate Flow	**User:** Enters unregistered email. **System:** Displays "Email not found" message.

Table 3.4: Use Case: Upload Face Image

Use Case ID	UC-SS-1.4
Title	Upload Face Image
Actor	User, System
Pre-condition	User is logged in.
Post-condition	Face image is uploaded and analyzed.
Basic Flow	**User:** Uploads clear face image. **System:** Processes image and detects skin tone.
Alternate Flow	**User:** Uploads blurry image. **System:** Requests clearer image.

Table 3.5: Use Case: Solve Questionnaire

Use Case ID	UC-SS-1.5
Title	Solve Questionnaire
Actor	User, System
Pre-condition	User is logged in.
Post-condition	User input saved for analysis.
Basic Flow	**User:** Answers visual questionnaire. **System:** Records responses.
Alternate Flow	**User:** Skips questions. **System:** Prompts to complete all questions.

Table 3.6: Use Case: Add to Cart

Use Case ID	UC-SS-1.7
Title	Add to Cart
Actor	User, System
Pre-condition	User is logged in and has viewed recommendations.
Post-condition	Product is added to cart.
Basic Flow	**User:** Clicks "Add to Cart". **System:** Verifies availability and adds item.
Alternate Flow	**User:** Attempts to add out-of-stock item. **System:** Displays "Item not available".

Table 3.7: Use Case: Detect Skin Tone

Use Case ID	UC-SS-1.8
Title	Detect Skin Tone
Actor	System
Pre-condition	User has uploaded face image.
Post-condition	Skin tone is identified and stored.
Basic Flow	**System:** Analyzes image and returns skin tone data.
Alternate Flow	**System:** Returns "Unable to detect" for poor image.

Table 3.8: Use Case: Analyze Questionnaire

Use Case ID	UC-SS-1.9
Title	Analyze Questionnaire
Actor	System
Pre-condition	User has submitted responses.
Post-condition	Skin characteristics derived.
Basic Flow	**System:** Processes answers to identify skin type.
Alternate Flow	**System:** Returns "Insufficient data".

Table 3.9: Use Case: Suggest Foundation or Product

Use Case ID	UC-SS-1.10
Title	Suggest Foundation or Product
Actor	System
Pre-condition	Skin tone + questionnaire results available.
Post-condition	Suitable products shortlisted.
Basic Flow	**System:** Matches skin data to product database.
Alternate Flow	**System:** Returns generic products if no match found.

Table 3.10: Use Case: Manage Product

Use Case ID	UC-SS-1.1
Title	Manage Product
Actor	Admin
Pre-condition	Admin is logged in and has rights.
Post-condition	Product list updated in database.
Basic Flow	**System:** Displays all products and allows editing.
Alternate Flow	**System:** Displays error if products not found.

Table 3.11: Use Case: Add Product

Use Case ID	UC-SS-1.2
Title	Add Product
Actor	Admin
Pre-condition	Admin is on product panel.
Post-condition	New product added to database.
Basic Flow	**System:** Saves new product details.
Alternate Flow	**System:** Shows error for missing/invalid fields.

Table 3.12: Use Case: Manage Orders

Use Case ID	UC-SS-1.3
Title	Manage Orders
Actor	Admin
Pre-condition	Admin can access order panel.
Post-condition	Order status updated/viewed.
Basic Flow	**System:** Displays all orders and allows updates.
Alternate Flow	**System:** Shows error if update fails.

Chapter 4

Design

4.1 System Design

This chapter includes the High-Level Design (HLD) and Low-Level Design (LLD) of SkinSage. HLD includes the data flow diagram, system architecture diagram, use case diagram, and deployment diagram. LLD covers the module-level design such as class structure, API communication, database schema, and UI workflows.

User Interface (UI) design is also an essential part of the system. The GUI was created using a soft baby-pink and neutral skin-tone theme to ensure aesthetic appeal, emotional comfort, and inclusiveness. Special focus was given to readability, simplicity, and smooth navigation so that users of all ages and technical backgrounds could easily use the application.

The following design constraints were also considered:

- **Performance requirements:** rapid TFLite inference and low-latency Firebase reads.
- **Device limitations:** ensuring responsiveness on mid-range smartphones.
- **Usability constraints:** prioritizing short workflows, concise instructions, and friendly graphics.
- **Security controls:** implementing Firebase Authentication and Firestore rules.

The development team ensured that SkinSage is reliable, effective, scalable, and capable of meeting user expectations.

4.2 System Architecture

In this section, the general structure of the SkinSage application is described, with a particular emphasis on interactions of various parts (Flutter frontend, Firebase backend, ML model).

- **Flutter App** (User Interface)
- **Firebase Authentication** (User sign-up/login)
- **Firebase Firestore** (Storage of user profiles, recommendations, analysis history)
- **Firebase Storage** (Image uploads for analysis)
- **Machine Learning Model** (Skin tone detection & recommendation logic using TensorFlow Lite)

The system-architecture-diagram of the SkinSage system is presented in Figure 4.1.

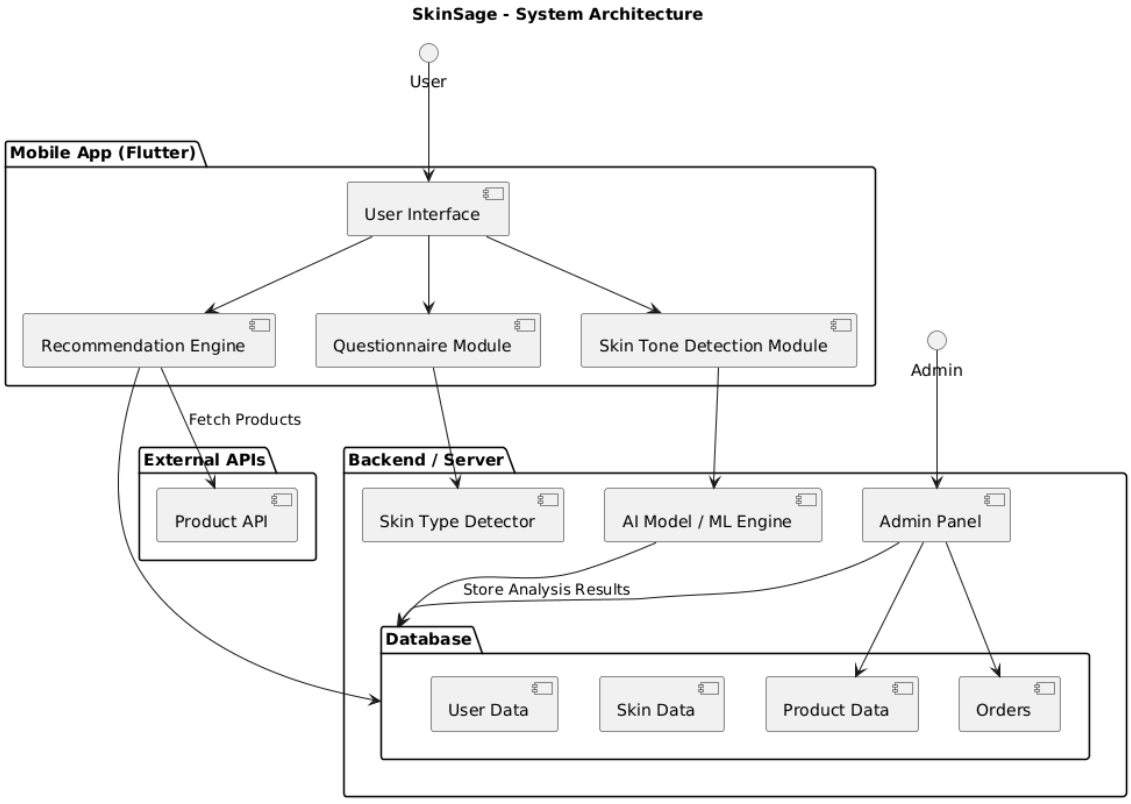

Figure 4.1: System Architecture

4.2.1 Four-Layer Architecture

The four-layer design of the SkinSage application can be described with the help of the high-level architecture diagram. The system consists of the following layers:

1. Frontend Layer (Flutter Mobile App)
    - Purpose: Serves as the user interface.
    - Technology: Developed using Flutter, which supports cross-platform deployment.
    - Functions: Interacts with users, displays recommendations, captures face images, and maintains the overall user experience.
2. Code Server (Firebase Cloud Services)
    - Firebase Authentication: Manages secure user accounts and login sessions.
    - Cloud Firestore: Stores application data including user profiles, product catalog, skin analysis records, and order history.
    - Firebase Storage: Securely stores user-uploaded face images.
    - Cloud Functions: Executes server-side code for complex logic and administrative processes.
3. AI/ML Engine (TensorFlow Lite)
    - Technology: TensorFlow Lite model running on the mobile device.
    - Purpose: Performs real-time skin tone analysis using face images.
    - Major Strength: Processes data locally on the device, ensuring user privacy and delivering fast responses.
4. External Integrations
    - Payment Gateway: Handles secure transactions required for e-commerce features.

4.3 Design Constraints

The following constraints influenced the design of the SkinSage system:

Table 4.1: Design Constraints

Constraint Type	Description
Performance Constraints	Skin image analysis completes within 5-10 seconds, and Firebase queries must return results in real time to maintain application responsiveness.
Security Constraints	User authentication is managed through Firebase Authentication, while sensitive data and user images are securely stored in Firebase Storage with strict access rules.
Usability Constraints	The UI follows a light baby-pink and neutral skin-tone theme for aesthetic appeal. Interfaces are designed to be simple and intuitive for non-technical users.
Technological Constraints	Application requires an active internet connection for Firebase services. It supports only Android and iOS platforms as it is built using Flutter.
Scalability Constraints	Firebase backend efficiently handles multiple users simultaneously without performance degradation.
Reliability Constraints	System maintains uptime and guarantee data consistency across user devices using Firebase real-time synchronization.

4.4 Design Methodology

4.4.1 Agile Scrum

The Agile [14] Scrum framework has been chosen for the development of the SkinSage mobile app because of its adaptability, incremental approach, and emphasis on collaboration and user satisfaction. SkinSage combines a Flutter-based mobile app with a Firebase backend and an AI-powered skintone detection model. These elements mature at different rates and need frequent adjustments, so Agile Scrum works well for controlling this level of complexity. This flexibility means the team can adapt to changes in the development phases (UI tweaks, updated Firestore rules, or upgraded TFLite models) as iterative sprints provide continuous development without the risk of pausing progress.]

With its customer-focused approach, Agile methodology shares development of features such as login module, image upload page, and skin analysis result in each sprint, with early feedback to improve on usability, navigation, and image processing. Scrum further promotes early risk identification; every sprint starts with planning and ends with testing, exposing risks such as model accuracy and Firebase conflicts or UI performance bottlenecks, which can be addressed early, thus reducing the cost of rework.

The incremental nature of Agile Scrum makes it possible to create, test, and integrate various components such as authentication, questionnaires, AI inference, recommendation system, and so forth, early in the process. This supports milestone demonstrations, experimentation, and streamlined progress to keep up with deadlines cite scrum2020.

4.4.2 Sprint Breakdown

Table 4.2: Sprint Breakdown

Sprint	**Deliverables**
Sprint 1	Basic Flutter app structure and setup of Firebase Authentication.
Sprint 2	Image upload module with Firebase Storage integration.
Sprint 3	Skin tone analysis using ML model and product recommendation engine.
Sprint 4	GUI refinements (baby pink + skin-tone theme), profile and history modules.
Sprint 5	Final testing, debugging, and deployment.

4.4.3 Iterative Development Process

The Agile development lifecycle for SkinSage follows these steps in each sprint:

- **Requirement Gathering:** Identify user needs, UI updates, and Firestore schema changes.
- **Design:** Create UI mockups, workflow diagrams, and module outlines.
- **Implementation:** Develop Flutter screens, integrate Firebase services, and connect ML inference.
- **Testing:** Conduct unit testing, integration testing, and user acceptance testing.
- **Deployment:** Deliver the sprint output for review and feedback.

The data-flow-diagram of the SkinSage system is presented in Figure 4.2.

Figure 4.2: Agile Scrum Model

4.5 High Level Design

User

- **Inputs:** Name and passwords, facial photos, survey responses, transaction/shopping data.
- **Outputs:** Authentication status, skin analysis results, personalized product recommendations, and order confirmations.

Administrator

- **Inputs:** Commands to manage products and orders.
- **Outputs:** Management interface confirmations and system reports.

Backend Services

- **Firebase Auth:** Handles secure user authentication.
- **Firestore:** Stores all user profiles, product data, and order information.
- **Firebase Storage:** Stores user-uploaded face images.
- **ML Model:** Processes images and returns skin tone classification.

The high-level context of the SkinSage system is presented in Figure 4.3.

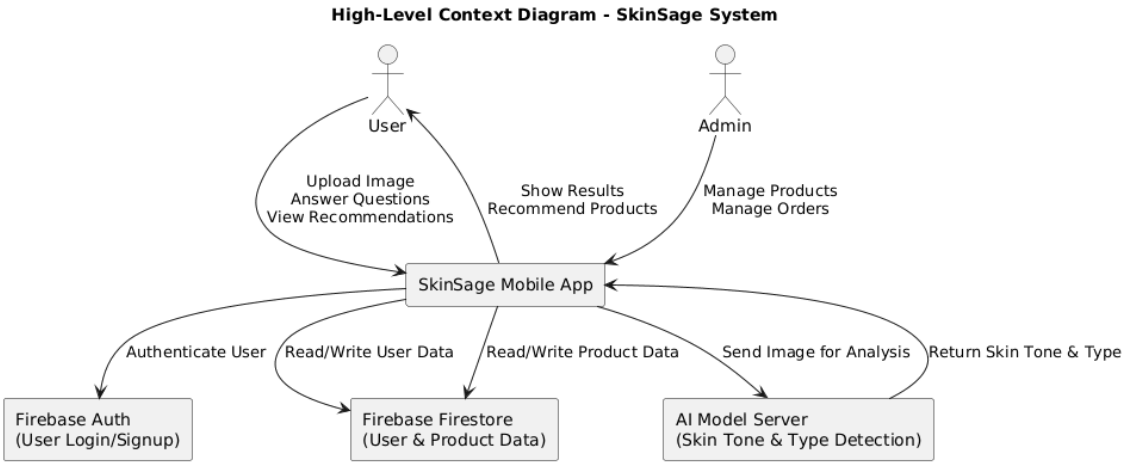

Figure 4.3: High-Level Context Diagram

4.6 Low Level Design

4.6.1 Explanation of Key Components

Processes:

- **Manage User Account & Auth:** Handles all authentication operations such as signup, login, and logout. Communicates with Firebase Authentication and manages basic user profile information (D1).
- **Skin Analysis:** Core AI process. Accepts user inputs (facial image and questionnaire), stores them (D2, D3), and invokes the external ML model to generate a skin tone classification.
- **Make Recommendations:** The personalization engine. Retrieves user profiles (D1), skin analysis results (D3), and product catalog data (D4), then generates personalized product recommendations.
- **Process Orders / Manage Cart:** Handles complete e-commerce operations including cart management and order placement. Interacts with product database (D4) and order storage (D5).
- **Administer System & Products:** Allows the administrator to manage the product catalog (D4) and view/update orders stored in (D5).

Data Stores (System Memory):

- **D1: User Profiles** – Stores user details and preferences.
- **D2: Skin Images** – Stores metadata and references to uploaded facial images.

- **D3: Questionnaire Data** – Stores responses to the skin-analysis questionnaire.
- **D4: Product Database** – Contains the full skincare product catalog along with attributes such as skin type/tone suitability.
- **D5: Orders / Carts** – Stores active shopping carts and completed order history.

Explanation: This figure effectively decomposes the system into major functional processes, illustrates the data flow among modules, and shows how each data store supports the application's overall functionality for both Users and Administrators.

External Entities:

- User
- Administrator
- Firebase Authentication
- ML Model (TensorFlow Lite / Firebase ML Kit)

The low-level context of the SkinSage system is presented in Figure 4.4.

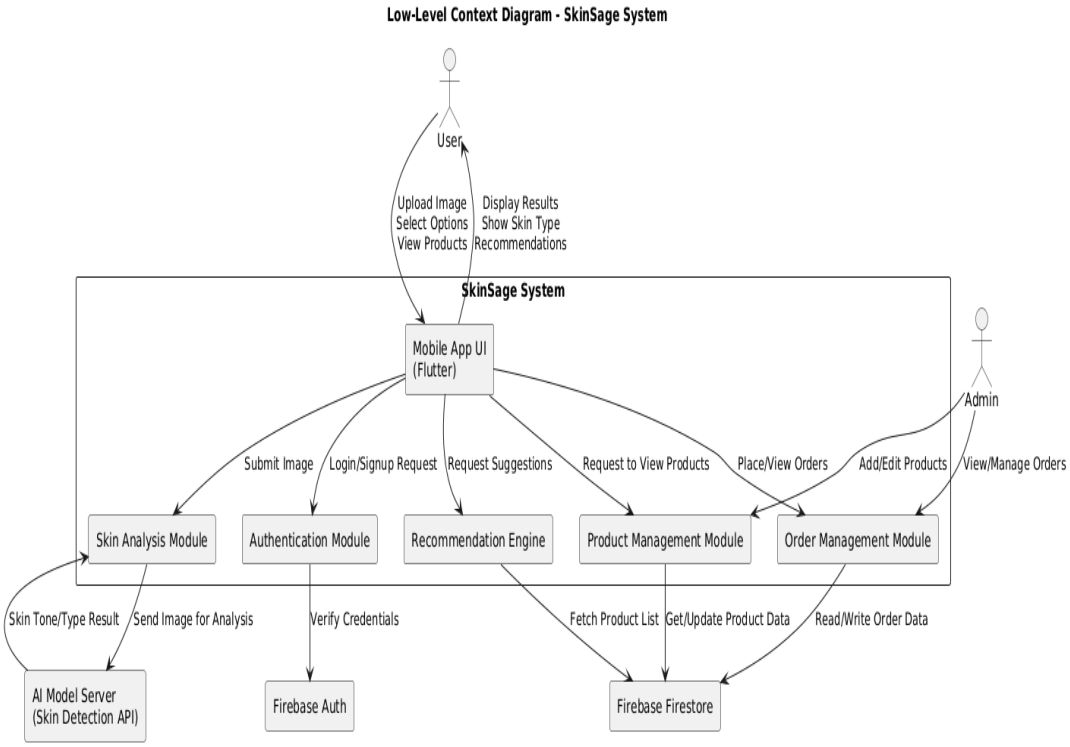

Figure 4.4: Low-Level Context Diagram

4.7 Database Design

The design of the SkinSage system database has been developed to well reinforce the main functions of the system. It keeps users' information, records the skin analysis results, the product information, and personalized recommendations. The system lets users take skin tests, and the results are saved in the database to provide customized skin care recommendations. Through the platform, the administrator can also add products, view orders, and track user activity. By storing the data in an orderly fashion, the database makes sure all system operations, including user authentication, product management, recommendation generation, and order tracking, are performed smoothly and reliably. It is this organization that enables the SkinSage system to provide a flawless user experience and at the same time keep highly accurate information readily available for the users and the administrators

The Entity-Relationship Diagram of the SkinSage system is presented in Figure 4.5.

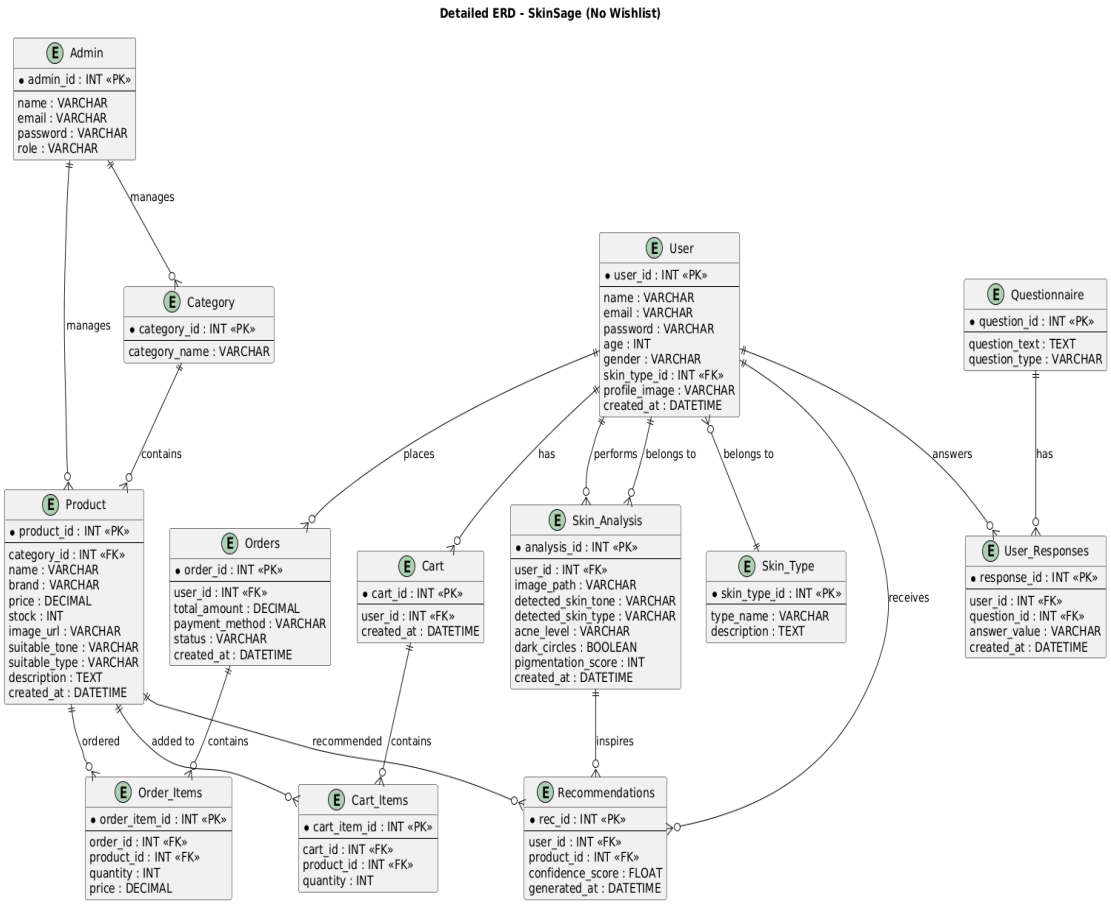

Figure 4.5: ERD Diagram

Chapter 5

System Implementation

The key stage is system implementation which is a period where the conceptual design of a system is converted to a fully operational functional solution. The implementation of the SkinSage mobile application proceeded in the following way: mobile development technologies, machine learning models, cloud service, and user-interface elements were combined into a single platform and could analyze the skin tones and suggest appropriate skincare and cosmetic products. The mobile app has been created with Flutter because of its cross-platform, good performance, and the ability of a flexible system of widgets to create interactive and beautiful interfaces. In the case of backend services, the Firebase Authentication, Firestore Database and Cloud Storage are utilized to administer user accounts, save results of analysis, and keep products data safely. Python, TensorFlow/Keras, and image processing tools, including OpenCV[15] , were used to create the model of skin tones detection. The trained model was finally turned into a TensorFlow Lite (TFLite) which can be used to operate rapidly on mobile devices assuring real-time performance on both Android and iOS. The model is used to determine the category of skin tones of a user and relay it to the mobile application to be displayed. Several security measures were observed in order to achieve safety and privacy. Firebase Authentication secures the account user by using secure user log-in, password encryption, and session management. The Graphical User Interface (GUI) design and development are also described in the chapter and is highly important in ensuring the user-friendly and intuitive experience.

5.1 Dataset

Large-scale face attributes dataset (CelebA) with approximately 200,000 images, used for skintone detection.

5.1.1 Images

- **Total Images:**
  - Training: 144,784 images
  - Validation: 36,194 images
  - **Total: 180,978 images**
- **Image Size:** 128 × 128 pixels
- **Batch Size:** 32

5.1.2 Classes / Labels

The dataset contains 5 skin tone categories:

- Beige
- Dark
- Fair
- Light
- Medium

5.1.3 Preprocessing and Data Augmentation

The dataset used Keras ImageDataGenerator with the following preprocessing steps:

- **Rescaling:** Pixel values normalized to range 0–1 `rescale=1./255`
- **Validation Split:** 20% of data for validation
- **Augmentation Techniques:**
  - `rotation_range = 10`
  - `width_shift_range = 0.1`
  - `height_shift_range = 0.1`
  - `zoom_range = 0.1`
  - `horizontal_flip = True`

This helps the model generalize better by introducing variations in angle, zoom, and orientation.

5.1.4 Data Balancing

- Class distribution was checked using utility function in scikit-learn.
- Class weights were applied during training to balance underrepresented classes, ensuring equal importance for all skin tones.

The data balancing is presented in Figure 5.1

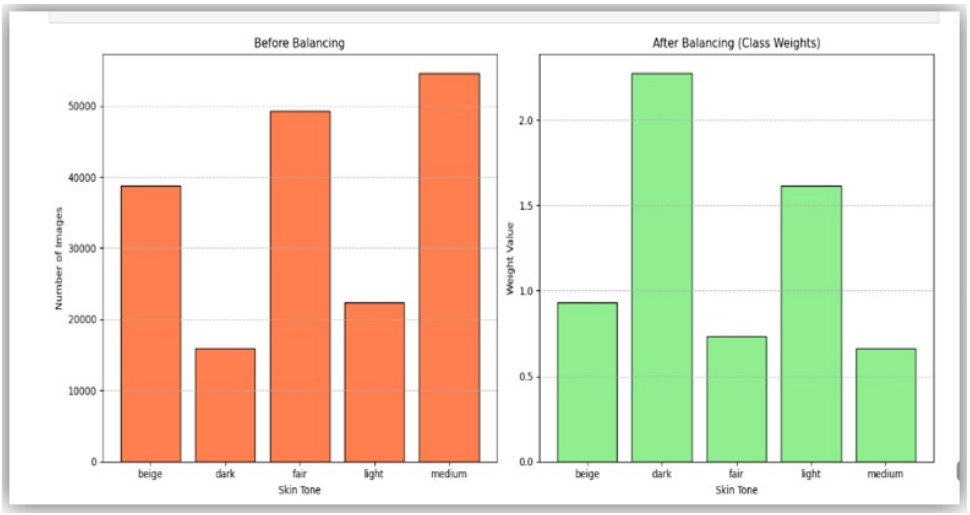

Figure 5.1: Data Balancing Before and After

Before Balancing (Left Graph) there were a lot of images (very tall bars) of medium and fair dark has fewer images (small bar). That's what we call class imbalance, meaning your model could learn more about fair/medium tones and less about dark tones.
After Balancing (Right Graph) it shows computed class weights — these are not image counts, they are importance multipliers used during training. That's why:

- dark has the highest weight (2.27) — it means the model gives 2x more importance to dark images because they were fewer.
- medium and fair have lower weights (<1) — since they already have many samples, the model gives them less emphasis.

5.1.5 Model Overview

- **Model Type:** Convolutional Neural Network (CNN)
- **Layers Used:**
  - Convolutional Layers (Conv2D)
  - MaxPooling2D

  - Flatten Layer
  - Dense (Fully Connected) Layers
  - Dropout Layer (to prevent overfitting)

- **Loss Function:** categorical_crossentropy
- **Optimizer:** Adam
- **Metric:** Accuracy

5.1.6 Training of Data

The training data of the SkinSage model was a set of labeled facial images that were divided into five categories: Fair, Light, Medium, Beige and Dark. The preprocessing of the data involved resizing every image to dimensions 128 x 128 and normalization of the pixel values in the range 01. Data augmentation: rotation, horizontal flipping, zooming, and contrast adjustment were done to enhance generalization of the model and address the changes in light, angle, and background. The Convolutional Neural Network (CNN) architecture based on the combination of convolutional and pooling layers, and then fully connected dense layers was used to train the model. The model performance was optimized with the Adam optimizer and categorical cross-entropy loss function. The data was divided into 80 percent training and 20 percent validation data. In training, classes weights were used to accommodate the imbalance in the classes and make sure that underrepresented quality of the skin tones classes had an equal contribution to the learning process. The model underwent training up to 25 epochs, and the initial learning rate of the model was 0.001 then refined to 0.00001 as the model would achieve better accuracy and stability. The training was implemented on TensorFlow with Keras API and the trained model was subsequently converted to a mobile-deployable format, TensorFlow Lite (.tflite), to use in the SkinSage application.

5.1.7 Testing and Validation

To test the model, the model was tested on the validation dataset to determine its accuracy and loss per epoch. The testing stage was a measure of the effectiveness of the model to classify previously unknown images into the right skin tones. The validation accuracy plateaued at 82-87 and the validation loss decreased steadily with epochs, which meant that the learning ability was high and the overfitting was low.

The model further was tested with random unknown samples which confirmed that it was able to pick the skin tone of the user in different lighting conditions and at different angles of the picture. The dataset accuracy is presented in Figure 5.2

	precision	recall	f1-score	support
beige	0.84	0.85	0.84	7752
dark	0.81	0.94	0.87	3186
fair	0.88	0.81	0.84	9867
light	0.82	0.93	0.87	4480
medium	0.86	0.83	0.85	10909
accuracy			0.85	36194
macro avg	0.84	0.87	0.86	36194
weighted avg	0.85	0.85	0.85	36194

Figure 5.2: Dataset Accuracy

5.1.8 Result and Analysis

The trained CNN model[16] was able to recognize and classify skin tones in five categories namely Fair, Light, Medium, Beige, and Dark. The overall validation performance reached about 85 per cent, proving a good capability of the model to generalize and be very robust. Data balancing and augmentation played an important role in enhancing accuracy in the classes because it decreased bias to common classes. The results confirmed that:

- The model can distinguish between close skin tones.
- Balancing and data augmentation increased the stability of the model.
- The model was lightweight and efficient to be applied in a mobile device due to the integration of TensorFlow Lite.

All in all, the trained model offers valid findings of analysis of real-time skin tones, the basis of SkinSage recommendation system, which proposes appropriate beauty and skincare products based on skin tone of each user. The dataset accuracy is presented in Figure 5.3

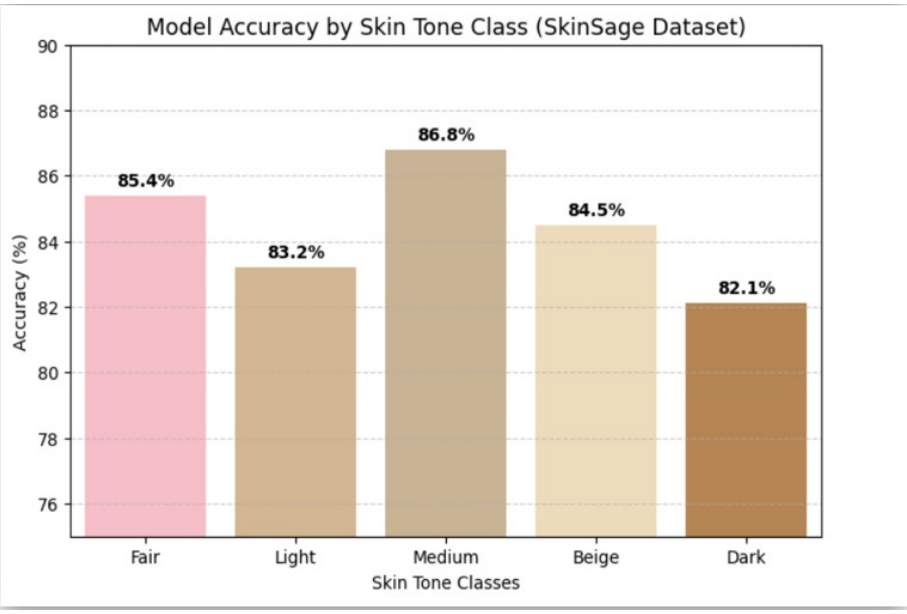

Figure 5.3: Results

5.2 System Architecture

SkinSage application is based on a client-server architecture with machine learning services. The architecture will provide a smooth communication between the mobile client, backend services and machine learning model in order to give real-time skincare recommendations.

5.2.1 System Components

- **Mobile Application (SkinSage Client):**
  This is the main user interface that is developed in Flutter and is used when uploading pictures, solving a questionnaire, browsing products, and getting AI-based tips on how to use skincare products.
- **Machine Learning Model (Skin Tone Detection Engine):**
  Implemented using TensorFlow Lite / Firebase[17, 18] ML Kit, the ML model[16, 19] analyzes the uploaded facial image to detect the skin tone.
- **Firebase Authentication:**
  Used for secure user registration, login, and account management.
- **Firebase Firestore (Cloud Database):**
  The store stores user information, history of skin analysis, questionnaires and customized products.
- **Firebase Storage:**
  Responsible for securely storing user-uploaded images used for skin analysis.

- **Admin Dashboard:**
  Gives the administrators the capacities to control product data, track reviews, retrain the AI set, and track user activity.

5.2.2 Functionality of Components

Flutter Mobile Application

The SkinSage system uses the Flutter app as the main interface where users can interact with the system. It enables the users to create an account or log in, post their face photos, complete skin-related tests, and see their own product suggestions. The app interacts with machine learning model and Firebase services over the secure API calls. The reason is that it is cross-platform, making it work well on Android-based devices, provides a responsive interface, easy navigation, and real-time feedback when analyzing skin tone. Flutter takes care of preprocessing tasks such as image resizing prior to transmitting them to the ML model making it an efficient processing and results production.

Machine Learning (ML) Model

The ML model has the mandate of detecting and classifying skin tone. It is made with the help of TensorFlow and is compiled into a small TensorFlow Lite (TFLite) platform, which directly processes the uploaded face images on the device. This is to guarantee quick inference, reduce latency and improve data privacy as images are not transmitted to third parties servers to be predicted. The model will remove prominent features of skin color, map them to a set of standardized skin tones, and transfer the outputs to the app. Personalized recommendations are fueled by the output of the ML model, which matches the identified skin tone with the products that are available in Firebase.

Firebase Firestore (Cloud Database)

Firestore is the main cloud database where organized data like product information, foundation shade libraries, users and their profiles, questionnaire feedback, and AI-generated outputs will be stored. This database is structured to be a collection-document design so that it can be quickly queried and maximally retrieved. Once the ML model recognizes a skin tone, the data is correlated with the products that are saved in the Firestore to make individualized suggestions. The user preferences and the previous analyses are also stored in Firestore that will allow a more personalized user experience in the future. Product syncing in real-time makes sure that the app is always updated with the most recent product data.

Firebase Authentication

Firebase Authentication will make sure that registered and authorized users will be able to enjoy the features of the app. It manages secure user management, which includes email/password registration, log in, password recoveries, and session management. This will ensure that personal skin analysis results are not accessed or the user profile information is not

accessed without permission. Firebase has default password policies that are automatically enforced and the system offers token authentication to ensure communication between the app and the backend services have high integrity. Consequently, the information of the user is secure during the process of interacting with the system.

Firebase Storage

Firebase storage holds uploaded face images of users in a safe place where they can be accessed in case they need to be processed using the ML. The FirebaseStorage has stringent rules of security of stored images to avoid illegal access or leakage of data. As an illustration, the user deleting an image can only see the image and removes it. Unique paths are also assigned to images so that they are not overwritten and their retrieval is effective. The app uploads images in compressed format in order to save latency and storage space consumed. This is a safe storage system to facilitate proper analysis without the loss of user privacy and data security.

5.2.3 Communication between Components

- The mobile application is connected to the Firebase backend through HTTPS request.
- The pictures are posted in Firebase storage to be analyzed with the help of an ML.
- The ML model either works on the cloud (Firebase ML Kit) or locally (TensorFlow Lite).
- The logic will be to store processed results and questionnaire responses in Firestore and then provide personalized recommendations on skincare.
- Data within the components is encrypted to ensure the privacy of the users.

5.3 Tools and Technologies Used

5.3.1 Flutter Framework

Used to build a cross-platform mobile application for both Android and iOS, ensuring smooth UI/UX design with aesthetic baby pink and skin-tone themes[17].

5.3.2 Firebase Services

The application utilizes Firebase services[18] for backend functionalities, including:

- Firebase Authentication for sign-up/login.
- Firestore Database for user and product data.
- Firebase Storage for image uploads.

- Firebase Cloud Functions for handling recommendation logic and notifications.

5.3.3 TensorFlow Lite / Firebase ML Kit

For real-time skin tone detection and analysis of face images.

5.3.4 Visual Studio Code

The main development environment for writing and debugging both Flutter frontend and backend logic.

5.4 Processing Logic / Algorithms

5.4.1 Skin Tone Detection Algorithm

The Skin Tone Detection Algorithm is responsible for analyzing the user's uploaded facial image and determining the most accurate skin tone category.

- **User uploads a facial image:**
  The image is first preprocessed by the app—resized, normalized, and slightly enhanced to improve detection accuracy.
- **ML model analyzes pixel values:**
  The TensorFlow Lite model identifies the prevalent color data of the region of the cheeks, forehead, and chin. It dwells on the RGB values, undertones (warm, cool, neutral), and other elements of skin color without paying attention to the background and noise in lighting.
- **Comparison of shade identified with product database:**
  After identifying the category of the skin tone (e.g. fair, wheatish, deep) the app compares the tone to pre-stored product shades in Firebase Firestore. This makes sure that the user is provided with shade- precise foundation and skincare suggestions.

5.4.2 Questionnaire Analysis Algorithm

This part examines the responses of the user to short skin-type questionnaire.

- **User maintains the visual quiz:**
  The questionnaire has questions based on the oiliness, dryness, acne propensity, sensitivity, and daily routine.
- **Reactions are correlated with skin type:**
  Each response is rated with a fixed number which is added to one or more skin related characteristics. For example:

  - Shiny T-zone raises the score of oily skin.
  - The post-washing tightness raises the score of dry skin
  - Redness or itching enhances sensitive skin score.

  The algorithm determines the maximum score and classifies the user as Dry, Oily, Normal, Combination or Sensitive.

5.4.3 Recommendation Logic

This step combines both results—skin tone and questionnaire skin type—to form a complete, personalized product recommendation.

- **ML skin tone result (Shade):** Determines the color-based recommendations such as foundation, concealer, tint, powder shade.
- **Questionnaire result (Skin type):** Determines skincare needs such as
  - Oily $\rightarrow$ oil-free moisturizers
  - Dry $\rightarrow$ hydrating creams
  - Sensitive $\rightarrow$ fragrance-free products
  - Acne-prone $\rightarrow$ salicylic acid recommendations
- **Personalized Recommendation Generation:**
  This part examines the responses of the user to short skin-type questionnaire. User fills out the visual quiz: The questionnaire has questions based on the oiliness, dryness, acne propensity, sensitivity, and daily routine. There are skin type categories of responses: Each response is rated with a fixed number which is added to one or more skin related characteristics. For example:
  - Shiny T-zone raises the score of oily skin.
  - The post-washing tightness raises the score of dry skin.
  - Sensitive skin score is higher when it is erythematous or itchy.
  - Additional tips for improvement

  The algorithm determines the maximum score and classifies the user as Dry, Oily, Normal, Combination or Sensitive.

5.5 Application Access Security

5.5.1 Authentication and Authorization

- Users must register/login via Firebase Authentication (email + password).

- Admins have higher privileges to access dashboards and manage datasets.

5.5.2 Firestore Security Rules

- Role-based access ensures that only the user can view their data.
- Admins can update and manage product information securely.

5.5.3 Logging and Monitoring

- All login attempts, product updates, and user interactions are logged for monitoring.

5.6 Database Security

One of the most significant issues of the SkinSage mobile application is the data security compliance, as the system keeps such sensitive user data as the emails, profile details, and the data connected with skin-analysis. Firebase Firestore and Firebase Storage were authenticated, authored by rules, encrypted, and monitored in order to ensure integrity, confidentiality, and accessibility to the stored information. The security controls that were applied were as follows:

5.6.1 Remote Access Control

Access to Firestore and Storage is highly controlled remotely. Only the requests, which are based on authenticated and verified Firebase sessions, can be used to access the database. Illegal requests or anonymous requests are automatically blocked. All connections are by default secured using HTTPS through Firebase Console, which is resistant to attacks by hackers who may potentially intercept or modify data during transmission. That is to make sure that the users who would be connecting to the database remotely (through mobile app) are communicating through a secure, encrypted medium. Furthermore, Firebase relies on the authentication protocol of a token, i.e., it can only admit data accessing tokens that are valid IDs that are created when a user is properly logged in. These tokens have a time limit after which they are deleted and this minimizes chances of misuse due to stale sessions.

5.6.2 Authentication

Verification of user identities is done through firebase authentication before it is allowed any privilege in the database. The system also maintains good password policies that will help in making it resistant to brute force attacks and unauthorized access. Some of the POC steps Involve:

- **Password strength checking:** Firebase will enforce password length and urge the use of alphanumeric combinations, which lessen weak-password vulnerability.
- **Email verification:** Before some actions (e.g. changing password) users are required to verify their email to be sure that the user is legitimate.
- **Session management:** Active sessions are safely stored, and failure to use tokens will not allow exploitation over a long period.
- **Automated attacks protection:** Firebase has integrated features to track any multiple failed attempts to log in to a system as well as throttle requests.

Automated attacks protection: Firebase has integrated features to track any multiple failed attempts to log in to a system as well as throttle requests.

5.6.3 Authorization (Role-Based Access Control)

Firestore Security Rules can also be applied to users after authentication determining what each role can read or write. **Implemented Authorization Structure:**
Regular User

- Can only read/write data belonging to their own profile.
- Cannot access other users' skin analysis, recommendations, or cart data.
- Cannot modify administrative content.

Admin User

- Managing product listings.
- Managing orders.
- add new products

These Firestore rules serve as an electronic firewall in the database, and even authorized users cannot go beyond their authorized limits. This significantly lowers the potential risk of information leakage or overwrite or malicious access.

5.6.4 Auditing and Activity Logging

Firebase has an automatic maintenance of important operations logs. These logs are the logs that can be used to monitor and question any suspicious or abnormal behavior associated with authentication, access patterns and so on. **Logging Includes:**

- Login and logout attempts

- Account creation and deletion
- Unauthorized access attempts
- Log of every modification by the accounts of the administrators.
- Abnormal request bursts or abnormal behavior.

Such logs facilitate incidents discovery and assist the developers to analyze the possible vulnerabilities or misuse. Given that Firebase is also connected with Google Cloud Logging, the administrators will be able to configure custom alerts when it comes to the high-threat actions, which are repeated attempts to log in or attempts to write unauthorized data.

5.7 Application Pages

5.7.1 Splash Screen

Initial screen greeted to user upon app opening. It shows the name of the app, SkinSage, and the logo of the app. This screen will appear as a welcome page which will remain visible approximately 5 seconds as the app starts and loads the necessary resources. When the initiation process is finished, the screen fades off to the login page and users get a smooth start to their experience.The splash screen is presented in 5.4

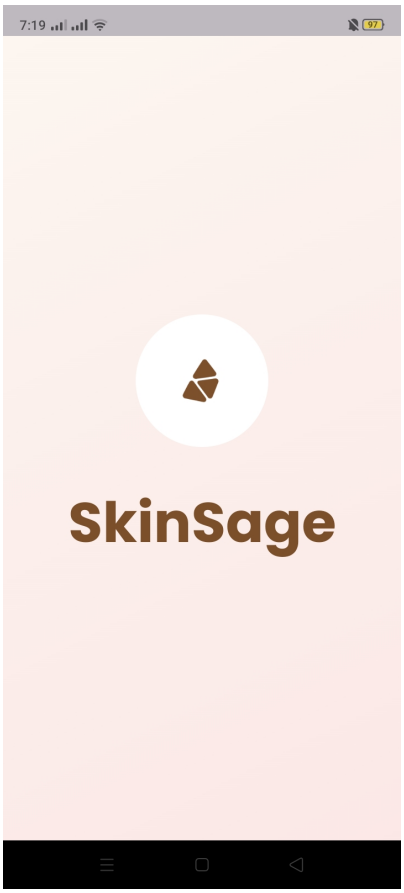

Figure 5.4: Splash Screen

5.7.2 Sign-up and Login

5.7.2.1 Sign-up Page

The users will create a user account using an email and password. The Sign-up Page 5.5 gives the new users an opportunity to subscribe to SkinSage application by entering their email address and password. After entering valid credentials, the user is safely saved in Firebase Authentication, and thus creates a safe and reliable method of registration. The system checks to ensure that there is no duplicate account using the same email address. When the user registers successfully, they will be redirected to the Login Page where they will access their account. The page allows increased security and lets users access SkinSage features personally.

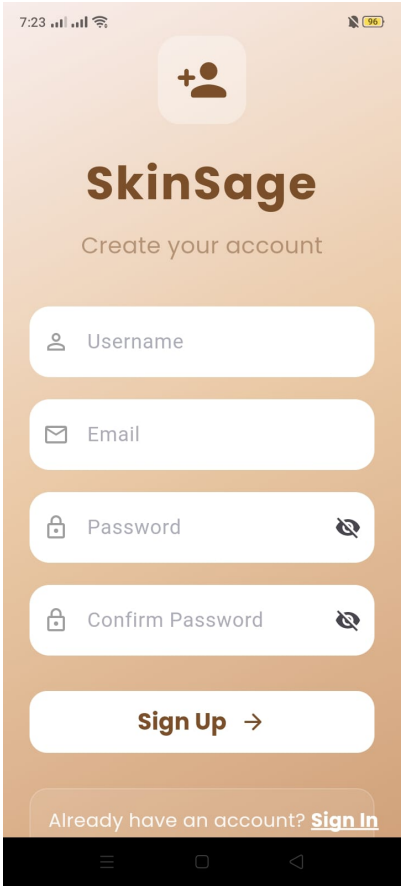

Figure 5.5: Sign-Up Page

5.7.2.2 Login Page

Users who have used the system before type in their credentials to gain access into the system. The Login Page 5.7 enables the registered users to access their SkinSage account in a secure manner. Users will need to fill in registered email and password. This is done through Firebase Authentication in which the credentials are checked to ensure that the user who logs in is authorized. Where wrong credentials are entered, a proper error message is shown. After authentication the user gets redirected to the Home Page where he can utilize the main functions of the application like the skin tone analysis and product recommendations.

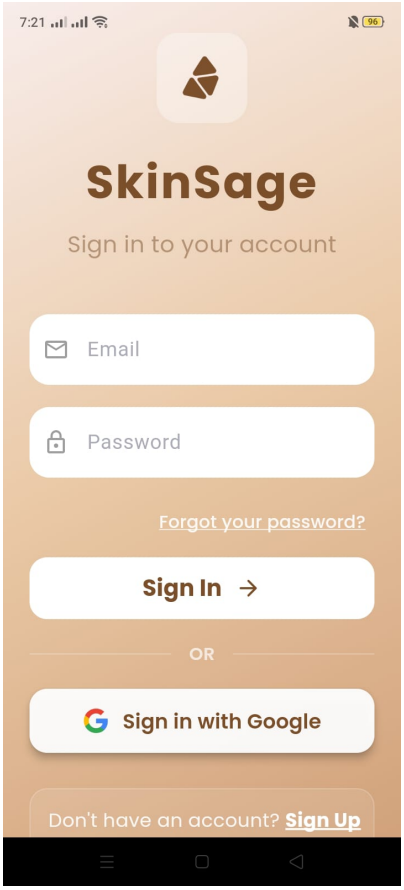

Figure 5.6: Login Page

5.7.2.3 Forgot Password

Users are able to recreate an account by firebase email verification. Forgot Password Page will help users to recover their accounts in case they have forgotten their password. Users will be given a password reset link that is created by Firebase Authentication by entering their registered email address. It is this connection that enables them to generate a new password safely. The attribute guarantees the recovery of the accounts without undermining the security of user information, which enhances the overall user experience and the reliability of the app.

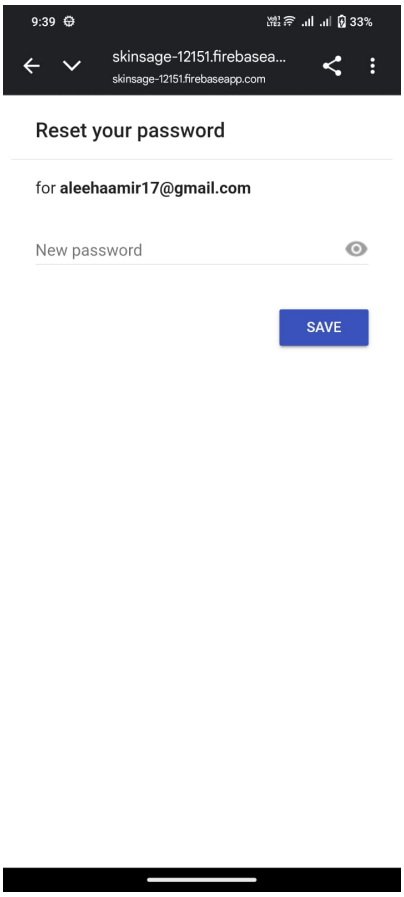

Figure 5.7: Password Reset Page

5.7.3 Home Page and Main Functionalities

The Home Page 5.8 provides access to the main features:

- Upload Face Image for AI skin tone detection.
- Solve Questionnaire for skin analysis.
- View Personalized Product Recommendations.
- Browse Reviews and Add to Cart.

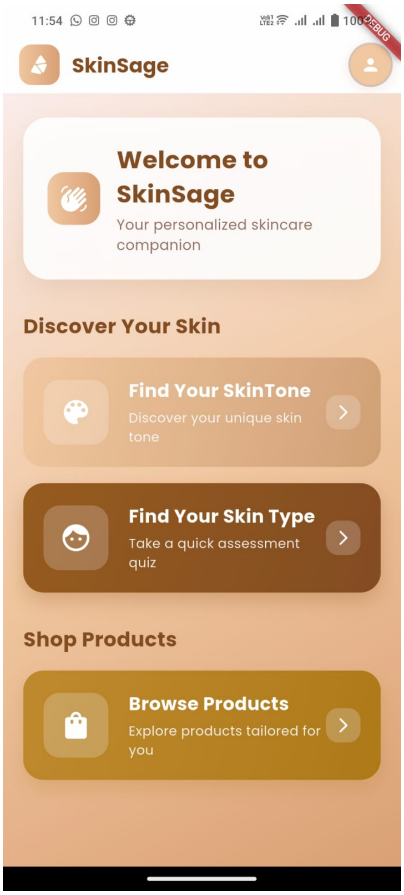

Figure 5.8: Home Page

5.7.4 Skin Analysis and Recommendations

Skin Analysis Page: A user posts the picture to have it analysed by AI. The main functionality of the SkinSage application is the Skin Analysis Page 5.9. It enables a user to take or post a picture of his face to be automatically analyzed in terms of the skin tones and types of the individual. This is done by running the trained model of TensorFlow Lite (TFLite) which classifies the skin tone of the user as Fair, Medium, or Dark. According to this analysis, the system identifies the most appropriate foundation shades and skincare products. The page is created with the convenient interface which presupposes easy interaction, quick results and retains the theme of light pink and skin-tone of the app.

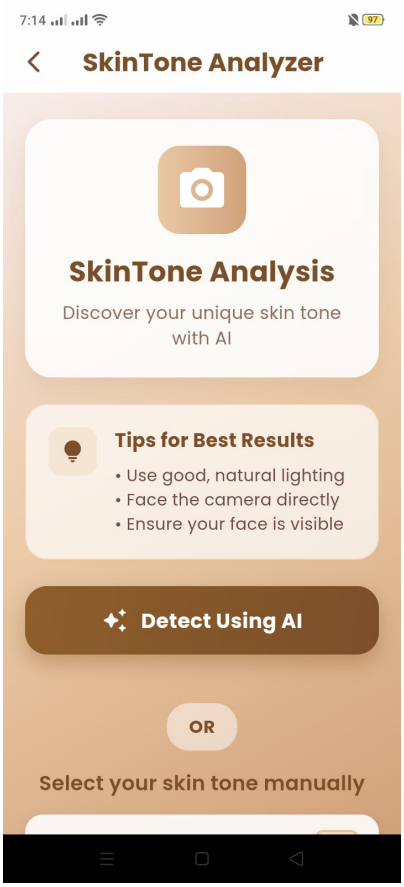

Figure 5.9: Skin Analysis Page

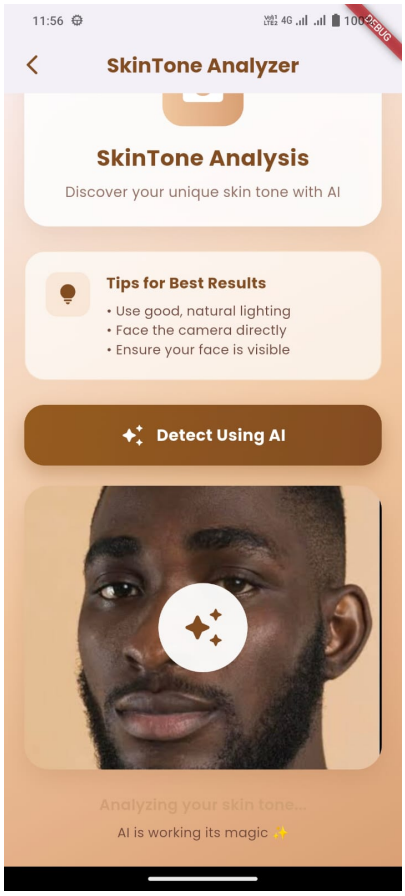

Figure 22: Skin Analysis Page

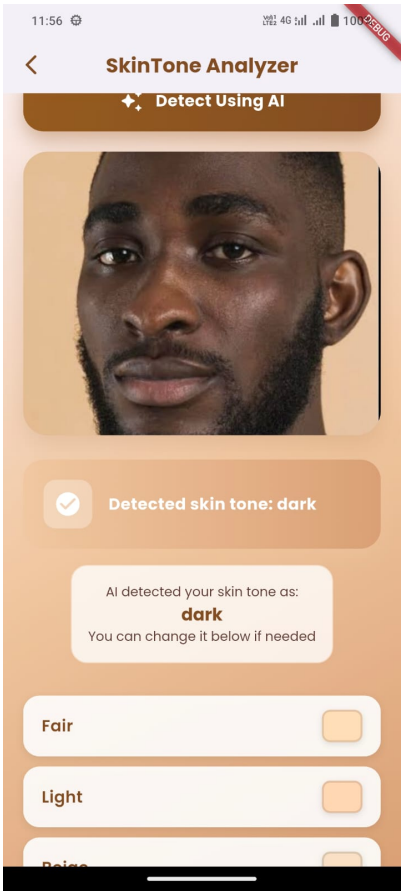

Figure 23: Skin Analysis Page

Recommendation Page: It has displays that suggest products, such as foundation shades, moisturizers, and skin care routines. The Recommendation Page 5.10 shows the recommendations on the products to buy according to the findings of the analysis of the skin. Based on the established skin tone and type, the system suggests foundations, moisturizers, and skincare products that best fit the complexion of the user. Every suggestion has a picture of the product, description, and the price. The database is dynamically accessed to fetch the recommendations and have the relevant and up-to-date recommendations. This aspect contributes to the user experience by offering personalized solutions of beauty care.

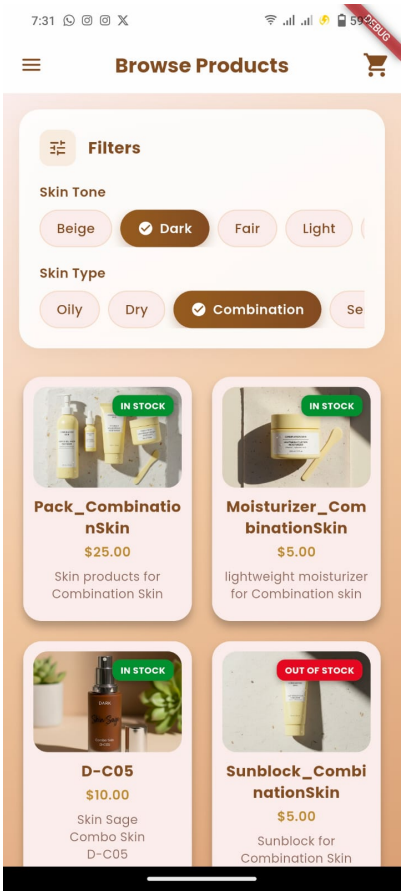

Figure 5.10: Recommendation Page

5.7.5 Product Details

The given application page is a product detail page (PDP) 5.11 of an online retail shop, which is aimed at informing the user about a particular skincare set. The advertisement displays product images of the set of "Pack-CombinationSkin" of the brand "Skin Sage" offered at a price of 25.00. The presence of the statement that the item is In Stock (10 available), the information regarding the suitability of the item to different skin tones (Beige, Dark, Fair, Light, Medium) and types (Combination), and the interactive section with the button of Add to Cart and the option of the quantity are the main details. And plus to wishlist button on the right upper part.

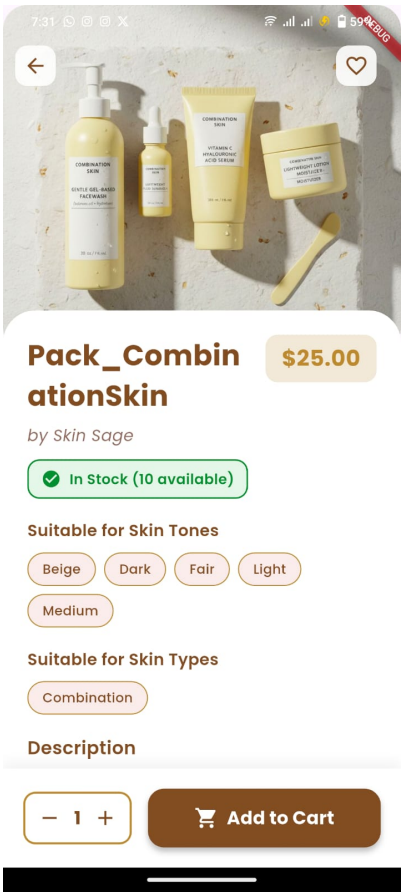

Figure 5.11: Product Details Page

5.7.6 Profile and Admin Dashboard

Profile Page: It is an application page that serves as an all-encompassing user profile and settings, which is intended to enable the user to control his or her application experience and manage his or her identity. It has two major parts, the upper part shows core profile data, such as the name of the user, email , and application-specific data, such as Skin Tone (fair) and Skin Type (combination). The lower half contains default standard account management options in a settings section, which contains a customization option of Editing Preferences, a security update option of Change Password, and a safe termination option of Logout. The general design is clean and modern with a soft beige palette which makes it easy to navigate. The profile page is presented in 5.12

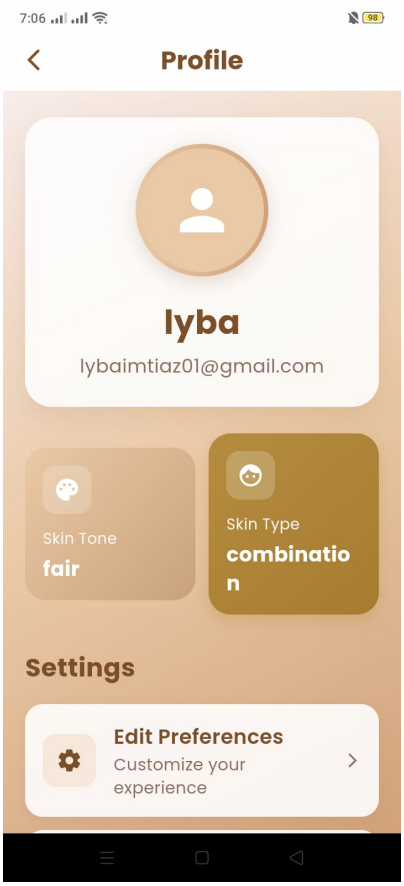

Figure 5.12: Profile Page

Admin Dashboard: The admins can add/edit/delete products, retrain AI datasets and manage users. Admin Dashboard 5.13 is created to allow application administrators to control users, products, and reviews. It will give information on the activity of users, product performance, and system statistics. Admins have access to addition, revision, or deletion of products, tracking of feedback, and data accuracy. The dashboard is constructed using a user-friendly interface, which allows easy management of the system, and easy operations of the back-end to maintain SkinSage application.

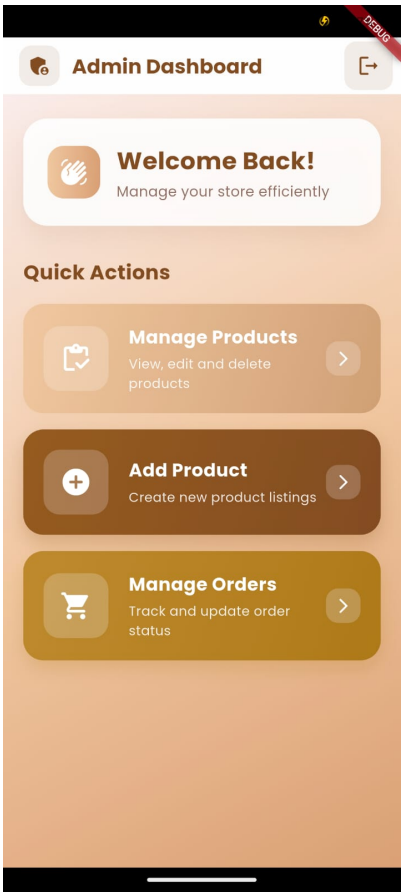

Figure 5.13: Admin Dashboard

5.7.7 Admin Product Management Pages

Manage Products Page

The Manage Products page 5.14 is the dash board on which the administrators can view, update, and delete products in the Firebase Firestore database. There is a clean card-based layout that shows all the products with necessary information as product name, image, price, stock and skin tones that they support. Two action buttons are also found on each card:

- **Edit (pencil icon):** Opens the Edit Product Page having the already filled product details.
- **Delete (trash icon):** Deletes the product selected on Firebase.

In this screen, the admins are able to handle the whole product inventory effectively. The soft pink/beige theme is very attractive and helps to be very clear, and such tags as fair, beige and dark help to understand on which skin tones the product helps to work. In general, the page provides a fast overview of the entire range of products available, which allows effectively tracking and maintaining stocks.

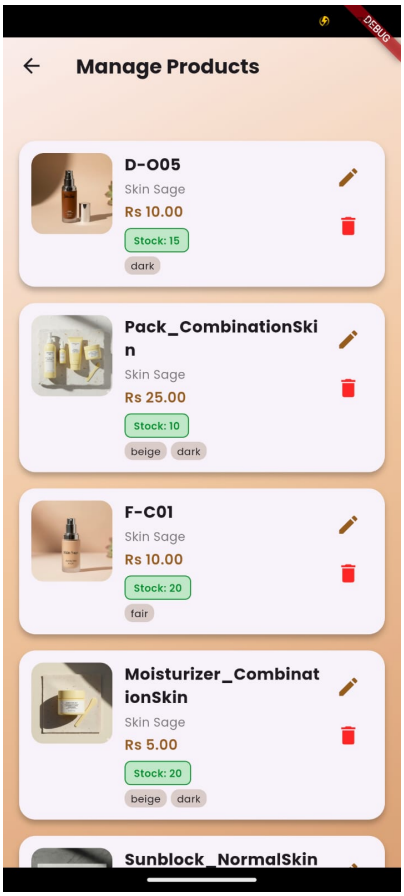

Figure 5.14: Manage Products Page

Edit Product Page

The Edit Product page 5.15 enables the admins to make changes to the existing product details with ease. After clicking on a product on the list of Manage Products, the admin is taken to this form where the existing information on the product is filled.

- Product Name
- Price
- Product Image (tap to re-upload)
- Brand Name
- Stock Quantity
- Description
- Skin Tones Supported
- Skin Types Supported

The interface is designed according to principles of edit-friendly interface, with big fields to input data and preview of the image of the product. This assists administrators that they have been correct with the details prior to updating them. After the edits have been saved, the updates will be reflected real-time on Firebase firestore. On this page, inventory management becomes easy since the admins can easily update inventory, prices or type of skin supported, including tone category without having to switch through the screens.

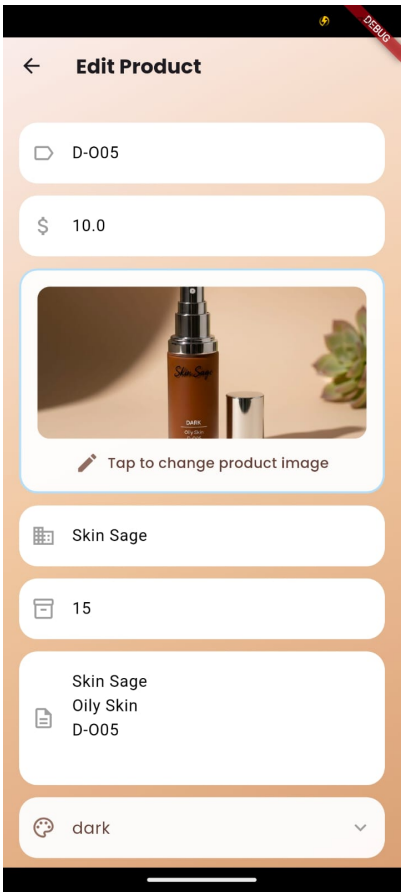

Figure 5.15: Edit Product Page

Add Product Page

Add Product page 5.16 helps admins to add new products to SkinSage product catalog. It offers a user friendly form whereby all the product specifications can be typed. Fields include:

- Product Name
- Price
- Product Image Upload
- Brand Name
- Stock Quantity
- Product Description
- Select Skin Tones (e.g., fair, beige, dark)
- Select Skin Types (e.g., oily, dry, combination, sensitive)

The data is sent to Firebase Firestore and the product image uploaded to Firebase storage with the help of the special Add Product button. The form will guarantee that nothing is overlooked and incomplete details will not be included.

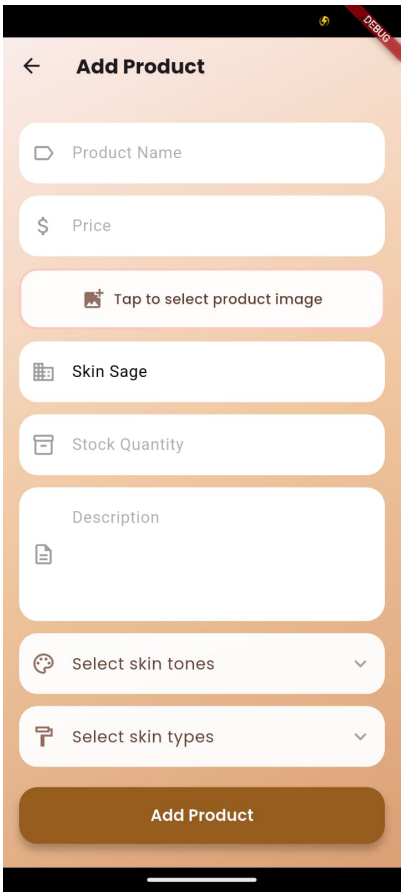

Figure 5.16: Add Product Page

Manage Orders Page

The Manage Orders page 5.17 gives an administrator the full view of all the order taken by customers in the SkinSage application. It should have an order-handling process that is easy to manage, giving admins a chance to track, update, and handle orders easily under one interface. **Order Statistics Overview**

At the top of the page, a summary dashboard displays:

- Total Orders
- Pending Orders
- Delivered Orders

These statistics allow the admin to quickly assess overall system activity and monitor daily order performance. **Order Filtering Options**

Below the statistics, multiple filter buttons help the admin sort orders based on their current status:

- All
- Pending
- Processing
- Shipped
- Delivered

This filtering system makes it easier to locate specific orders and handle time-sensitive tasks efficiently. **Order Card Details**

Each order appears in a card layout showing:

- Order ID
- Order Status (Pending/Processing/Delivered)
- Order Date
- Customer Details: Name, Phone Number, Address

The clean design allows easy accessibility of all the relevant details of the customer making it easy to verify and hence communicate when necessary. **Order Items Section**

The order card also lists:

- Product Image
- Product Name

- Brand
- Price
- Quantity Ordered

This allows the admin to review the contents of the order at a glance. The Total Amount is displayed clearly at the bottom for billing and accounting purposes. **Order Status Management**
The admin can manage order flow through two action buttons:

- Start Processing: Moves the order from Pending to Processing.
- Cancel Order: Cancels the order and updates the status in Firebase.

These buttons guarantee smooth work process between taking orders and delivery and real time update of status on the part of both the administration and the user.

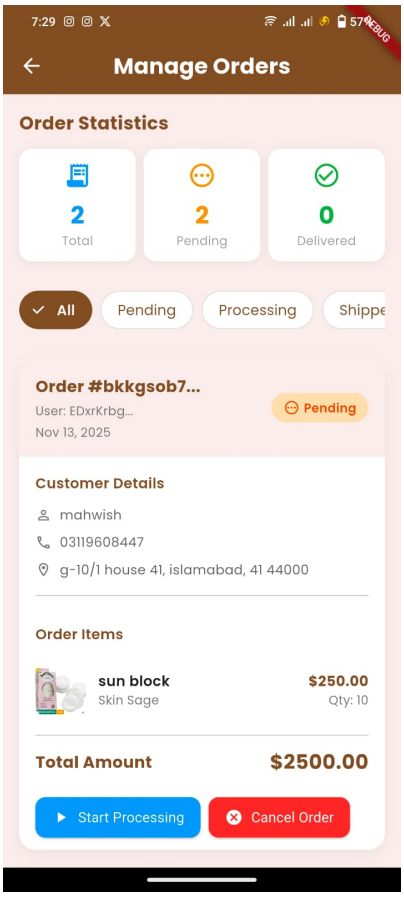

Figure 5.17: Manage Orders Page

5.7.8 Additional Features

The SkinSage application incorporates other few features to make it easier to use, customizable and the overall user experience good. These attributes make it convenient, more interactive and easy to use the system.

Manual Shade Selector

Once the AI has identified the skin color of the user, there is a manual shade chooser that enables the user to confirm, adjust, or remedy the shade identified. This aspect is particularly useful when there is a chance of lighting, camera position, or clarity of the image to influence the results of the detection.

Saved User Preferences

- Last detected skin tone
- Questionnaire results
- Recommended products

This allows users to revisit their previous analysis without repeating the entire process. The saved data improves convenience and enhances the personalized experience.

Update Prompt on Login

Whenever any user is logged in and after some time, the system examines whether:

- Their skin tone has changed
- New products are available
- Recommendations should be updated

A pop-up update message will be shown in which a user can either re-analyze or proceed with stored results. This will make the recommendations relevant and updated.

Stock and Inventory Status

The app provides real-time stock visibility, helping users make informed decisions:

- **In Stock** badge-product available
- **Out of Stock** label-product unavailable

For admins, this feature helps in inventory planning, while users avoid selecting unavailable products.

Chapter 6

System Testing and Evaluation

6.1 Testing Techniques

In a bid to guarantee the accuracy, reliability and the overall quality of SkinSage mobile application, there was a mixture of several testing methods employed during the development lifecycle. These techniques were meant to be able to find the defects at an earlier stage, authenticate the functional requirements, and measure the performance as well as ensuring that the system could act uniformly across various situations. Functional Testing initiated the testing process, and its aim was to certify the correctness of main modules, including Sign-up,Login, Forgot Password, Skin Tone Detection, Questionnaire Analysis, Recommendations Display,Add-to- Cart and Profile Management. The black-box testing was performed to test every feature individually and to make sure that the behavior of the system in terms of input and output was corresponding to the predefined use cases. Also, the Integration Testing analyzed the level of interaction between the modules. To illustrate, the image upload to preprocessing to tone detection to product recommendation was tested as one system to establish the flow of data and collaboration of the systems. It was also essential to do non-functional testing. This involved the performance testing, usability testing, reliability testing, compatibility testing and interface testing. Firebase authentication, TFLite model inference and loading product data were paid special focus as these aspects required real time calculations and online communication. All of these tests combined were necessary to be sure that the application was not only functional but also secure, scalable and easy enough to use, and to be rolled out and used in the real world.

6.2 Graphical User Interface (GUI) Testing

The GUI testing was conducted to determine the efficiency of the users in using the SkinSage application based on its visual components. This type of testing was necessary to verify that all the elements were working properly such as the buttons, icons, images,

text boxes, navigation menus, and output screens were delivering a smooth experience to the user. In the GUI testing, every screen of the application was reviewed on the following:

- **Layout Consistency:** The layout of all screens was clean and pleasing to the eyes with equal spacing, margins, and alignments. The use of the baby-pink and soft-neutral color palette was experimented with to ensure the comfort of reading and visualization.
- **Responsiveness:** The interface was tried with mobile devices of various screen sizes (since small smartphones to larger Android devices). Every section was dynamically adjusted without being overlapped and cropped.
- **Input Validation:** The presence of buttons and text fields was also tested and it was needed to respond to taps accordingly. The error messages were displayed in the correct location when invalid input was provided (e.g., fields were not filled during registration, email address was not a proper format).
- **Visual Clarity & Accessibility:** The company must have clearly displayed information to prevent misunderstanding of the website's content and information.
- **Navigation Flow:** Some typical user paths were put to test, e.g.:
- Login → Home Dashboard
- The next step is to upload image and then Skin Analysis and Recommendations.
- Profile → Edit Profile → Save Changes

All tests of the GUI were positive, which proves the fact that SkinSage has a smooth, appealing, and intuitive interface, which allows easy interaction with users.

6.3 Software Performance Testing

To evaluate the performance of the SkinSage application in terms of speed, stability, and efficiency of resources, performance testing was performed. As the system is based on machine learning models and image processing, performance testing was an important factor in making it react in real time.

- **Response Time:** TFLite skin tone recognition model was experimented in different conditions. The system was able to give correct tone predictions at an average rate of 2-3 seconds which is satisfactory when it comes to mobile AI uses. Response time to product recommendation in Firestore was also very fast (1–2 s).
- **Stress and Load Testing:** The system was tested with:

  - Multiple rapid face scans
  - High-resolution images
  - Consecutive logins/logouts
  - Repeated database queries

  The application continued to be stable in all situations, which proves that the existing architecture can support the normal user loads.

- **Network Performance:** Because SkinSage relies on Firebase to perform authentication and product information, network testing was conducted at different internet speeds. Results indicated:

  - Smooth performance on stable Wi-Fi
  - Acceptable performance on 3G/4G
  - Appropriate error messages on low/no connectivity

- **Resource Utilization:** The memory and CPU usage were monitored when uploading images and during model inference. The usage was optimal, and there were no crashes or serious frame drops. In general, it was possible to state that performance testing showed that the SkinSage system is very robust and efficient under various conditions of real-life use.

6.4 Usability Testing

Usability testing was intended to determine the ease with which the SkinSage mobile application could be learned, understood, and used by the users. The test was carried on with a real users who were not technical in nature in order to make sure that the interface was easy to use as well as the application process was natural. The participants were requested to carry out major tasks that included:

- Creating an account
- Logging in
- Uploading a face image
- Completing the questionnaire
- Viewing recommendations
- Editing their profile

The following aspects were tested during these tests:

- **Learnability:** Majority of the users could work without any help. The arrangement of buttons and icons was intuitive and navigation was not very difficult even to first-time users.
- **Efficiency:** Customers spent less time on the tasks with fewer steps. The redesigned button to analyze skin and recommendation cards that were simplified greatly cut confusion and enhanced precision.
- **Error Handling:** The application alerted the users when they missed some information or in cases where there was no clarity when uploading images. These hints served to prevent wrong submissions and led the users effectively.
- **User Satisfaction:** Feedback highlighted that:
- The color scheme was relaxing and appealing to the eyes.
- The guidelines on the individual screens were understandable.
- Overall, it was a professional modern experience.
- **Testing Results and Improvements Made:**
- Added the descriptions of tooltips under some of the icons.
- Better spacing between analysis screen.
- Improved error messages to enhance communication.
  Usability problems were also confirmed by the fact that SkinSage is an extremely user-friendly device and has a low learning curve and generally pleasant experience among users with different levels of technical expertise.

6.5 Test Cases for SkinSage Application

Test Case: Login / Logout

Table 6.1: Test Case: Login / Logout

Test Case ID	TC-SS-1.1
Actors	Login / Logout
Pre-Conditions	User must have a registered account.
Post-Conditions	User is logged in or logged out successfully.
Main Flow	User logs in or logs out successfully.
Alternate Flow	User enters login credentials or clicks logout. System validates credentials and logs user in or clears session.
Failure Scenario	User enters invalid credentials. System displays error message and denies access.
System Response	Login fails — incorrect email or password entered. System displays 'Invalid credentials'. Logout fails if session not cleared correctly.

Test Case: Register

Table 6.2: Test Case: Register

Test Case ID	TC-SS-1.2
Actors	Register
Pre-Conditions	User must not already be registered.
Post-Conditions	New account created.
Main Flow	New user account is created.
Alternate Flow	User fills in registration form. System validates data and creates account.
Failure Scenario	User leaves required fields empty. System prompts for missing information.
System Response	Registration fails — incomplete or invalid data entered. System displays 'Please fill required fields'.

Test Case: Forgot Password

Table 6.3: Test Case: Forgot Password

Test Case ID	TC-SS-1.3
Actors	Forgot Password
Pre-Conditions	User must have a registered email.
Post-Conditions	Reset link sent.
Main Flow	System sends password reset link.
Alternate Flow	User clicks 'Forgot Password' and enters email. System verifies email and sends reset link.
Failure Scenario	User enters unregistered email. System displays 'Email not found'.
System Response	System Response Password reset fails — invalid or unregistered email. System shows 'Email not registered'.

Test Case: Upload Face Image

Table 6.4: Test Case: Upload Face Image

Test Case ID	TC-SS-1.4
Actors	Upload Face Image
Pre-Conditions	User, System
Post-Conditions	User is logged in.
Main Flow	Face image is uploaded and analyzed.
Alternate Flow	Alternate Flow User uploads clear face image. System processes image and detects skin tone.
Failure Scenario	User uploads blurry or incomplete image. System requests clearer image.
System Response	Upload fails — poor lighting or incomplete image. System shows 'Unable to process image'.

Test Case: Solve Questionnaire

Table 6.5: Test Case: Solve Questionnaire

Test Case ID	TC-SS-1.5
Actors	Solve Questionnaire
Pre-Conditions	User, System
Post-Conditions	Post-Conditions User is logged in.
Main Flow	User input saved for analysis.
Alternate Flow	User answers visual questionnaire. System records responses for analysis.
Failure Scenario	User skips questions. System prompts to complete all questions.
System Response	Questionnaire incomplete — user skips mandatory fields. System prevents submission.

Test Case: View Recommendations

Table 6.6: Test Case: View Recommendations

Test Case ID	TC-SS-1.6
Actors	View Recommendations
Pre-Conditions	User, System
Post-Conditions	Skin analysis must be completed.
Main Flow	User sees recommended products.
Alternate Flow	User sees recommended products.
Failure Scenario	Failure Scenario User tries accessing recommendations before analysis. System prompts to complete skin analysis.
System Response	Failure Scenario User tries accessing recommendations before analysis. System prompts to complete skin analysis.

Test Case: Add to Cart

Table 6.7: Test Case: Add to Cart

Test Case ID	TC-SS-1.7
Actors	Add to Cart
Pre-Conditions	User, System
Post-Conditions	User logged in and viewed recommendations.
Main Flow	Product added to user's cart.
Alternate Flow	Product added to user's cart.
Failure Scenario	User adds out-of-stock item. System shows 'Item not available'.
System Response	Add to cart fails — stock unavailable or internet issue. System alerts user.

Test Case: Detect Skin Tone

Table 6.8: Test Case: Detect Skin Tone

Test Case ID	TC-SS-1.8
Actors	Detect Skin Tone
Pre-Conditions	System
Post-Conditions	User uploaded face image.
Main Flow	Skin tone is identified and stored.
Alternate Flow	System analyzes image and returns tone data.
Failure Scenario	System unable to detect tone. Displays 'Unable to detect'.
System Response	System Response Detection fails — low-quality image or model error. System logs issue.

Test Case: Analyze Questionnaire

Table 6.9: Test Case: Analyze Questionnaire

Test Case ID	TC-SS-1.9
Actors	Analyze Questionnaire
Pre-Conditions	System
Post-Conditions	User submitted responses.
Main Flow	Skin characteristics extracted.
Alternate Flow	System processes answers to identify skin type.
Failure Scenario	Incomplete answers. System returns 'Insufficient data'.
System Response	Analysis fails — model receives incomplete or corrupted data.

Test Case: Suggest Foundation or Product

Table 6.10: Test Case: Suggest Foundation or Product

Test Case ID	TC-SS-1.10
Actors	Suggest Product
Pre-Conditions	System
Post-Conditions	Skin tone and questionnaire data available.
Main Flow	Products shortlisted and ranked.
Alternate Flow	Products shortlisted and ranked.
Failure Scenario	No matching products found. System displays generic recommendations.
System Response	Recommendation fails — database unavailable or query error.

Test Case: Admin Dashboard

Table 6.11: Test Case: Admin Dashboard

Test Case ID	TC-SS-1.11
Actors	Admin Dashboard
Pre-Conditions	Admin, System
Post-Conditions	Admin logged in successfully.
Main Flow	Admin views and manages user data.
Alternate Flow	Admin navigates to dashboard and performs operations like view or delete user.
Failure Scenario	Admin action denied due to invalid credentials or permissions.
System Response	Admin panel fails to load — database connectivity issue or authentication error.

Chapter 7

Conclusions

7.1 Introduction

he development of the SkinSage mobile app represents a major milestone in bringing artificial intelligence technology and skincare personalization to a single digitally accessible platform. The main objective of this project—to create a practical, accurate, and easy-to-use system to identify a user's skin tone and type and recommend suitable makeup and skincare products was achieved. This chapter provides a brief review of the project's success in terms of its principal achievements and potential reform of the beauty and well-being market. System Achievements and Recap

The SkinSage platform is a seamless integration of advanced machine learning and accessible mobile technology. Its strength was in having an organized software development life cycle, which emphasised user centred design, system reliability and the accuracy of the models beneath. Key achievements include:

- **Our Personalized Skincare Advice:** system is unique in the fact that it gives you your own Skincare Expert, literally. Such advice is made possible by combining two functionalities:
  - **A**nalyzing images with machine learning to accurately assess the skin tone.
  - **U**sing questionnaire to determine the skin type precisely.
- **Robust Machine Learning Model:** The model was trained and fine-tuned on a large, clean, and balanced dataset to guarantee robust performance over various skin tones.
- **On-device, Responsive Inference:** Leveraging the TFLite model, this app performs on-device inference on mobile providing fast and good user experience in skin check without losing accuracy.

- **Scalable and Secure Infrastructure:** Using Firebase we were able to provide seamless and secure user authentication, data storage, and user management, and make the system secure and scalable for implementation in the real world.
- **Stable and Useful:** Functional, GUI, performance and usability tests confirmed that all modules (user authentication, skin analysis, product recommendation, and administrative functionality) behaved consistently and as expected even in the presence of errors. The User Interface was tested at friendly and easy to use and attractive to users with different levels of technical background.

7.2 Future Expansion Opportunities

Although the current SkinSage model serves as a robust core, there are multiple directions for future improvements and additions to keep it up-to-date and cover all aspects:

- **Larger product database:** The model can be improved by adding a wider range of products to be recommended in the beauty and skincare space.
- **Overall Skin Health Assessment in Real Time:** Additional functionality could be added, such as monitoring general skin health on a real-time basis, which would elevate the app from a simple recommendation-based, to a continuous wellness-based one.
- **Options for dermatological consultation:** The system can be further developed to allow for direct or virtual consultation with a dermatologist, introducing an element of professional medical supervision.

7.3 Final Summary

To conclude, the SkinSage mobile app accomplishes its goal and demonstrates what can be achieved when artificial intelligence is baked into a solution to bring practical benefit to everyday concerns of users. Delivering a unique data-driven experience in the field of skincare, the project not only increases user convenience but also illustrates the disruptive capabilities of intelligent mobile applications in the beauty and well-being domain. The developers' dedication to accuracy and user experience provides a solid foundation for future additions, making SkinSage a substantial leap forward in personalized digital health.

Appendix A

User Manual

Appendices are provided to give supplementary information, which is included in the main text may serve as a distraction and cloud the central theme.

- Author 1: Aleeha Amir (ID 01-135221-005)
- Author 2: Lyba Imtiaz (ID 01-135221-024)
- Title of Dissertation: SkinSage: An AI-Based Skincare Recommendation Mobile Application.
- Supervisor Mehwish Pervaiz

References

[1] Leslie Baumann. Skin type classification systems old and new. *Dermatologic Clinics*, 26(3):359–373, 2008. Cited on pp. 1 and 3.

[2] Thomas B. Fitzpatrick. Soleil et peau. *Journal of Medicine Esthétique*, 1975. Cited on p. 3.

[3] Wei Zhang and Hong Liu. Automatic skin type classification based on facial images. *IEEE Access*, 8:154310–154320, 2020. Cited on pp. 3 and 5.

[4] Muhammad Ali and Usman Khan. Facial skin type detection using computer vision techniques. *Journal of Ambient Intelligence and Humanized Computing*, 12(4):4567–4576, 2021. Cited on pp. 4 and 5.

[5] Jong-Hyun Park and Sung-Min Kim. Measurement of facial skin oiliness using image-based analysis. *Skin Research and Technology*, 22(4):447–453, 2016. Cited on p. 4.

[6] Andre Esteva, Alexandre Robicquet, and Bharath Ramsundar. A guide to deep learning in healthcare. *Nature Medicine*, 25(1):24–29, 2019. Cited on p. 5.

[7] Hien Pham and Jung-Hoon Kim. Artificial intelligence in cosmetic dermatology. *Journal of Cosmetic Dermatology*, 19(9):2185–2192, 2020. Cited on p. 5.

[8] Roxana Daneshjou et al. Disparities in dermatology ai performance on a diverse dataset. *Science Advances*, 7(31):eabj1859, 2021. Cited on p. 5.

[9] Gary Marcus and Ernest Davis. Ethical and societal risks of ai in healthcare. *MIT Technology Review*, 2020. Cited on p. 5.

[10] M. R. Hassan et al. Ai-based skin disease detection using image processing. *International Journal of Dermatology*, 58(9):1081–1089, 2019. Cited on p. 5.

[11] Maria Otworowska. Skinbliss: Ai-powered skincare analysis app. Online, 2021. Accessed: 2025-12-13. Cited on p. 6.

[12] Perfect Corp. Perfect skin tone analyzer: Ai-based skin tone detection app. Online, 2024. Accessed: 2025-12-13. Cited on p. 7.

[13] Right Foundation Team. Right foundation for you: Skin tone based foundation shade analyzer. Online, 2022. Accessed: 2025-12-13. Cited on p. 7.

[14] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for agile software development. Online, 2001. Accessed: 2025-11-26. Cited on p. 25.

[15] OpenCV Team. Opencv library. Online, 2024. Accessed: 2025-12-13. Cited on p. 31.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105. NIPS, 2012. Cited on pp. 35 and 36.

[17] Google. Flutter: Build apps for any screen. Online, 2024. Accessed: 2025-12-13. Cited on pp. 36 and 38.

[18] Google. Firebase documentation. Online, 2024. Accessed: 2025-12-13. Cited on pp. 36 and 38.

[19] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*. O'Reilly Media, 3 edition, 2023. Cited on p. 36.