AURA SKIN

SOFTWARE DESIGN DOCUMENT (SDD)

AI 7993 – Section W01 - Fall 2025

September 21, 2025

Nyah Robinson

Table of Contents

1. Introduction

1.1 Purpose

This Software Design Document describes the architecture, components, interfaces, and data flows of AuraSkin. It provides a detailed blueprint for developers and engineers to implement, integrate, and test the system. This document also ensures that design decisions align with the requirements defined in the Software Requirements Specification.

1.2 Scope

AuraSkin is a cross-platform mobile application designed to analyze skin conditions using AI-based image processing and provide personalized skincare recommendations. The system includes the following components:

- Image acquisition and preprocessing module.
- Skin concern detection via trained ML model.
- Recommendation engine with product/routine mapping.
- Mobile user interface.
- Backend database for products and historical data.

2. Design Considerations

2.1 Assumptions and Dependencies

- Users provide clear, front-facing images with proper lighting.
- The device must have a functional camera.
- TensorFlow Lite is available for ML model deployment on mobile devices.
- Flutter SDK and Dart are used for UI development.
- Backend relies on a cloud-hosted database.

2.2 General Constraints

- The app must run on iOS $\geq$ 13 and Android $\geq$ 10.
- Mobile device storage is limited; images must be processed efficiently.
- Processing should take less than 3 seconds for usability.
- Privacy requirements are images must be deleted immediately after inference.

2.3 Goals and Guidelines

- Achieve a minimum model accuracy of 85%.
- Prioritize lightweight models optimized for mobile deployment.
- Ensure secure handling of personal images.
- Provide a clean, intuitive user interface.

3. System Architecture

AuraSkin follows a client-server hybrid architecture:

- **Client-side (mobile app):** Image capture, preprocessing, on-device ML inference, UI.
- **Server-side (backend):** Recommendation database, product updates, user history storage.
- **Framework:** Flutter for cross-platform development.
- **Backend:** Firebase Authentication for login, Firestone for data persistence.
- **ML Integration:** TensorFlow Lite for efficient on-device model inference.
- **Cloud Hosting:** Google Cloud for backend services and updates.
- **Error Recovery:** Retry mechanism for network requests and graceful error handling for image uploads.
- **UI Strategy:** Minimalist design with guided steps for image capture, analysis, and results display.

4. Detailed System Design

4.1 Module Descriptions

Module 1: Image Acquisition & Preprocessing

- Captures or uploads images.
- Preprocesses: resize 256x256, normalize pixel values, detect face region.
- Validates image quality; prompts retake if blurry or underexposed.

Module 2: Skin Concern Detection

- **Model:** CNN optimized for mobile inference.
- **Input:** 256x256 RGB image.
- **Output:** Probability distribution over skin conditions (acne, dryness, hyperpigmentation, oiliness, healthy).
- **Framework:** TensorFlow Lite.

Module 3: Recommendation Engine

- Maps detected condition(s) to personalized skincare products and routines.
- Rule-based mapping with potential for ML-driven personalization.
- Retrieves relevant entries from the product database.
- Ranks recommendation based on confidence scores and product suitability.

Module 4: Mobile App UI/UX

- Built using Flutter with responsive layouts for iOS and Android.
- Main screens:
    - **Home:** Capture/Upload image.
    - **Analysis Results:** Detected conditions with confidence scores.
    - **Recommendations:** Products and routines displayed with descriptions.
    - **Saved History:** User's saved recommendations.
- UX emphasizes simplicity and clarity.

Module 5: Database Design

- Schema includes:
    - Users (user_id, preferences, history).
    - Conditions (condition_id, description).
    - Products (product_id, name, type, brand, skin_type, condition_id).
    - Recommendations (mapping between conditions and products/routines).
- Hosted on Firebase/Firestore (NoSQL).

4.2 Classification

- UI Layer: Flutter components (login screens, capture screens, results, results page, history page).
- ML Layer: TensorFlow Lite model for skin concerns detection.
- The recommendation layer helps database queries mapping concerns to products/routines.
- The backend layer manages user data, authentication, and secure access to stored history.

5. Data Design

5.1 Data Flow

1. The user captures or uploads an image.
2. Preprocessing cleans and resizes images.
3. ML model classifies skin condition(s).
4. Classification results sent to the recommendation engine.
5. Engine queries database and retrieves recommendations.
6. Results displayed on UI.

5.2 Data Dictionary

Entity	**Attributes**	**Description**
User	user_id, email, preferences, history	Stores user info & saved results.
Condition	condition_id, name, description	Skin concern categories.
Product	product_id, name, brand, type, skin_type, description	Skincare products.
Recommendation	rec_id, condition_id, product_id, priority	Maps conditions to products.

6. Interface Design

6.1 User Interfaces

- **Image Capture Interface:** Camera access, gallery upload.
- **Analysis Results Screen:** Detected conditions with confidence scores.
- **Recommendation Screen:** Scrollable list of product/routine suggestions.

- **History Screen:** Saved results, option to delete.

6.2 External Interfaces

- **APIs:** Database APIs for product retrieval.
- **Camera API:** Device camera integration.
- **Cloud Services:** Firebase/PostgreSQL.

6.3 Error Handling

- If an image is low-quality the system prompts users to retake.
- If ML confidence is less than threshold then the system outputs it uncertainly and suggests retake.
- If no internet connection the analysis proceeds locally, but recommendations are cached or delayed.
- If the database query fails the system fallback message is "Unable to retrieve recommendations."

6.4 Security and Privacy Design

- User images processed locally; temporary storage cleared after inference.
- No raw images stored on cloud servers.
- Database stores anonymized user IDs and history only.
- Data encrypted in transit (HTTPS) and at rest (AES_256).
- Compliance with GDPR/CCPA for handling sensitive user data.

6.5 Future Enhancements

- Multi-condition simultaneous detection.
- Personalized recommendations via user profile learning.
- Dermatologist consultation integration (via API).
- AR feature to track skin improvements over time.

7. Appendices

7.1 Glossary

Acne – a common skin condition that causes pimples, blackheads, whiteheads, and other blemishes.

AI – Artificial Intelligence, the capability of computational systems to perform tasks typically associated with human intelligence

API – Application Programming Interface, are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols.

CCPA – California Consumer Privacy Act, is a landmark California law that grants consumers the right to know what data is collected, the right to delete it, and the right to opt-out of its sale or sharing.

CNN – Convolutional Neural Network, a type of artificial neural network used for tasks like image and video recognition.

Dryness – a lack of moisture in the skin.

GDPR – General Data Protection Regulation, a European Union regulation that sets strict rules for how organizations collect, process, and store the personal data of individuals.

Hyperpigmentation – a condition where the skin develops areas of increased pigmentation, resulting in darker patches or discoloration.

ML – Machine Learning, a field of study in AI concerned with the development and study of statistical algorithms that can learn from data.

Noise Reduction – the process of removing unwanted noise from a signal, like images, to improve clarity and quality.

Normalize – the process of transforming image pixel intensity values into a common and consistent range or statistical distribution.

Oiliness – the condition of containing or being covered with oil.

Resize – alter the size of the image.

UI – User Interface, the point of interaction between a human and a digital product, such as a website, application, or device.

UX – User Experience, the overall feeling a person has when interacting with a product, system, or service, encompassing every aspect of their interaction from discovery to post-use.

7.3 Diagrams

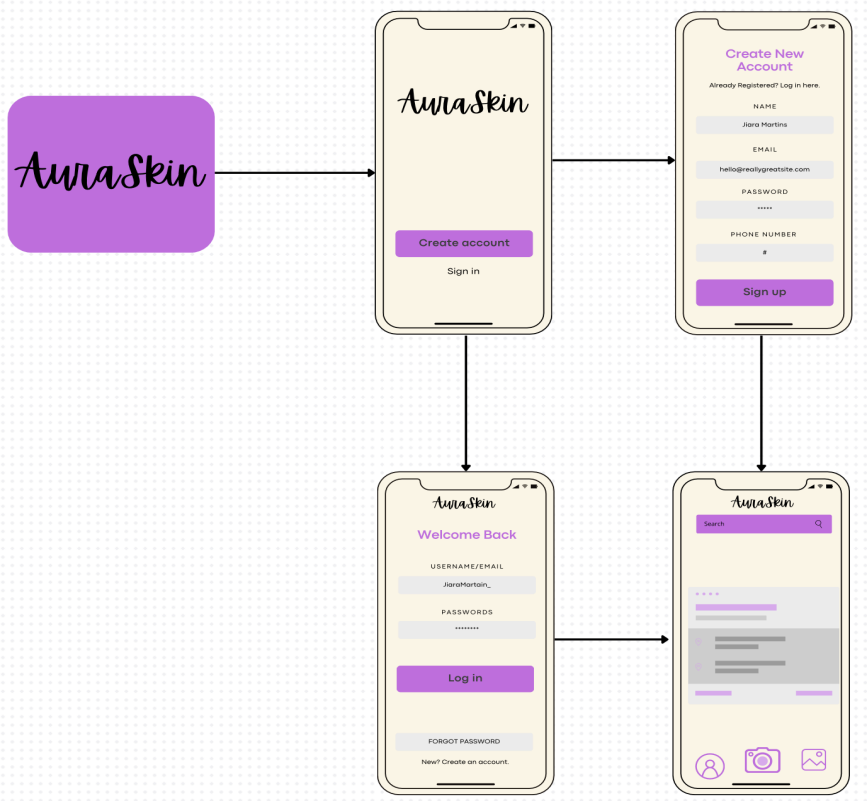

Figure 1: Mobile App Mock-Up

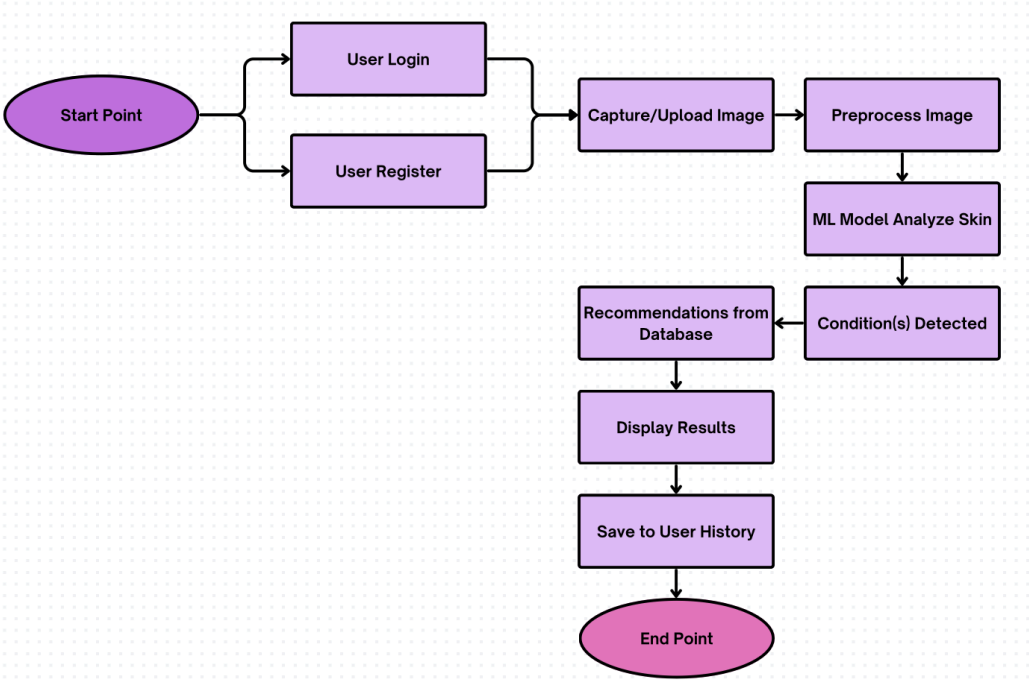

Figure 2: Process Flow Diagram

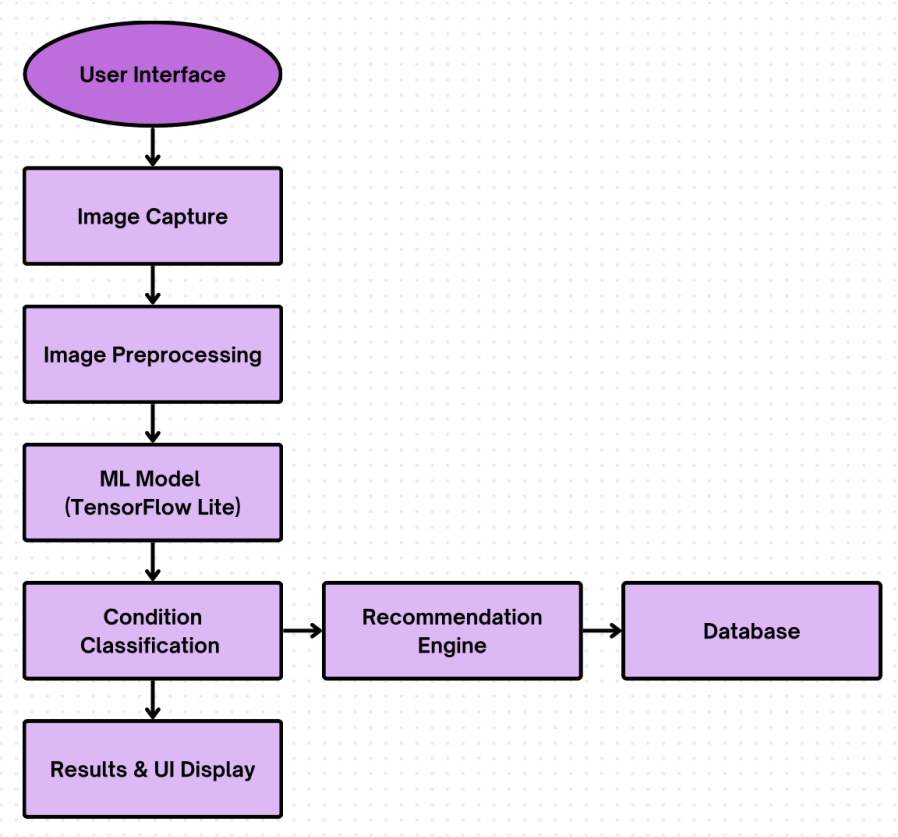

Figure 3: High-Level Architecture

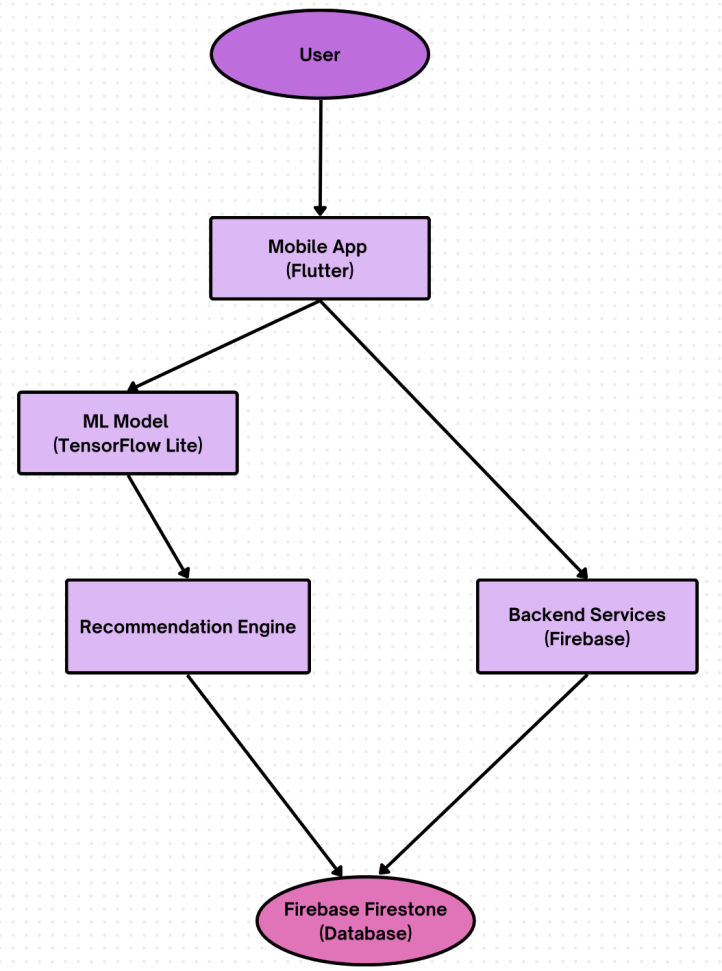

Figure 4: System Architecture

7.4 Tutorials

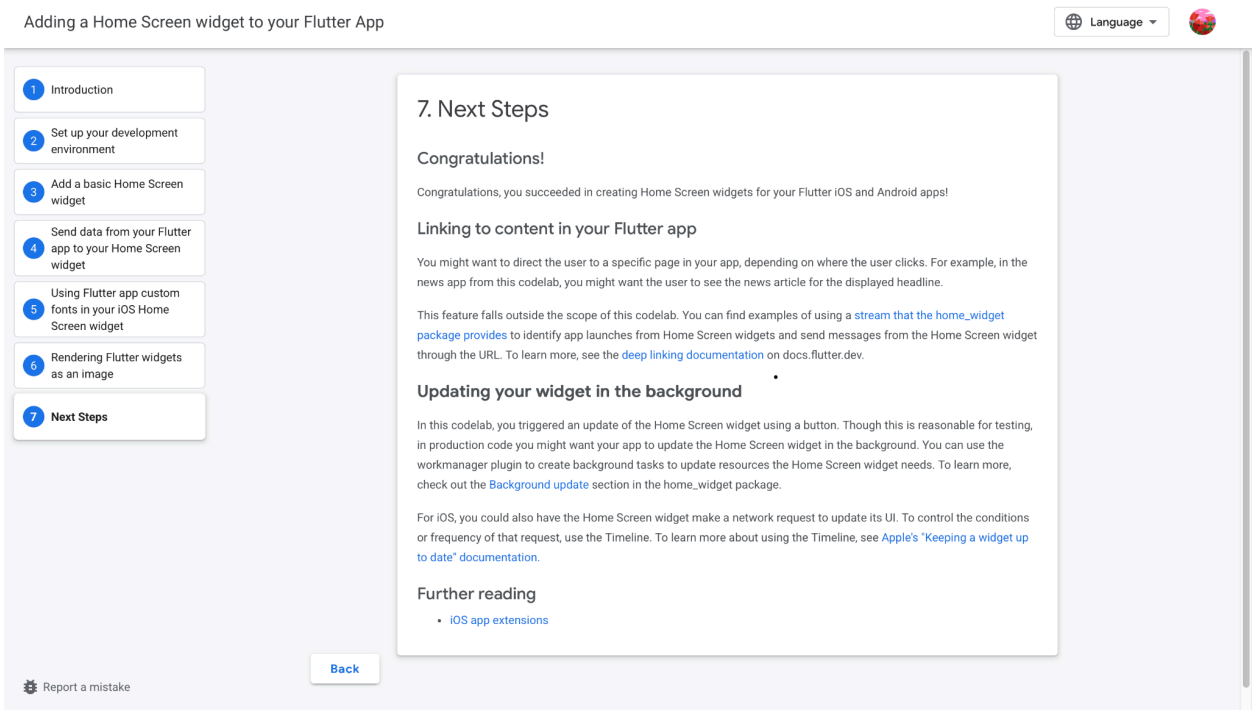

Figure 1: Adding a Home Screen Widget to your Flutter App

Figure 2: Read and Write Files