Riphah International University Lahore, Pakistan

Riphah School of Computing & Innovation

Final Year Project

PROJECT REPORT (Part-1)

HAIRLYTIC

Project ID: -

Project Team

Student Name	Student ID	Program	Contact Number	Email Address
Minahil Sharif	49004	BSSE	03094900205	Minahilsharif28@gmail.com
Fizza Ijaz	49260	BSSE	03229419580	Fizzaijaz11@gmail.com

Kanza Zafar
Senior Lecturer

Chapter 1

Introduction

Chapter 1: Introduction

Alopecia, a condition leading to partial or complete hair loss, affects millions worldwide and often causes both physical and emotional distress. Traditional diagnosis requires dermatological expertise and specialized equipment, making early detection difficult for many individuals.

Hairlytic addresses this challenge through an **AI-based web platform** that enables users to **diagnose alopecia from home**. The system combines **live scalp image analysis**, **medical questionnaires**, **report uploads**, and **guided self-assessment tests** such as the Hair Pull and Hair Count tests to identify the **type and severity of alopecia**.

1.1. Background

Alopecia is a common dermatological disorder caused by genetic, autoimmune, hormonal, and environmental factors. Early detection and timely treatment significantly improve patient outcomes, but conventional diagnostic methods are resource-intensive and time-consuming. Modern AI techniques and mobile/web applications provide an opportunity to bring accurate and accessible diagnosis directly to patients, empowering them to monitor their hair health from home. Hairlytic integrates AI, guided tests, and medical data analysis into a single ecosystem to address this gap.

1.2. Motivations and Challenges

Motivations:

- Provide accessible and accurate alopecia diagnosis for patients at home.
- Reduce dependency on in-person dermatologist visits for initial assessments.
- Combine AI with guided hair tests and patient medical history for precise results.

Challenges:

- Capturing high-quality scalp images remotely.
- Ensuring patient adherence to guided self-tests.
- Integrating AI models with real-time feedback and reporting mechanisms.

1.3. Goals and Objectives

Goals:

- Develop an AI-powered platform for at-home alopecia detection.
- Provide an end-to-end solution combining AI analysis, questionnaires, report uploads, and guided self-assessment tests and dermatologist appointment bookings.

Objectives:

- Enable patients to perform Hair Pull, Tug, and Hair Count tests with guided instructions.
- Collect and analyze patient questionnaires and medical reports to enhance AI diagnostic accuracy.
- Offer appointment booking and recommendations for verified dermatologists.
- Deliver real-time notifications and downloadable diagnostic reports.

1.4. Existing Solutions

Features	TrichoScan	MyHair.ai	MDhair	DermaQ	iHairium	HairSnap / HairLoss AI	Regrow Hair AI	Hairlytic
AI-Based Alopecia Detection	✓	✗	✓	✓	✗	✗	✗	✓
Live Camera Scanning	✗	✓	✓	✗	✓	✓	✓	✓
Questionnaire-Based Diagnosis	✗	✗	✓	✓	✗	✗	✗	✓
Medical Report Upload & Analysis	✗	✗	✗	✗	✗	✗	✗	✓
At-Home Test Simulation (Hair Pull, Tug, Count)	✗	✗	✗	✗	✗	✗	✗	✓
Real-Time Analysis & Results	✗	✓	✓	✗	✓	✓	✓	✓
Dermatologist Recommendation & Appointment Bookings	✗	✗	✗	✗	✗	✗	✗	✓
All-in-One Alopecia Diagnosis Ecosystem	✗	✗	✗	✗	✗	✗	✗	✓

1.5. Gap Analysis

Despite the availability of various hair loss diagnostic tools, most solutions lack integration of AI, patient questionnaires, medical report analysis, and at-home guided tests in a single platform. Patients often need to visit clinics for initial assessments, and existing tools do not provide a complete ecosystem for self-monitoring, appointment management, and actionable feedback. Hairlytic fills this gap by providing an **all-in-one, patient-centric, AI-powered solution**.

1.6. Proposed Solution

Hairlytic is an **AI-powered web platform** developed to enable **at-home diagnosis and management of alopecia**. The system uses **live camera scanning** with guided head movements to capture clear scalp images, which are analyzed by **machine learning models** to detect alopecia presence, type, and severity.

In addition to image-based analysis, Hairlytic incorporates a **questionnaire and report analysis module** that collects user medical history, hereditary information, and uploaded diagnostic reports (such as blood tests or dermatology notes) to enhance the precision of diagnosis. The platform also simulates **at-home versions of standard hair assessments**, including the Hair Pull Test, Tug Test, and Hair Count Test, allowing users to perform simple guided evaluations through their camera.

1.7. Project Plan

This section describes how the Hairlytic project will be managed and executed through clearly defined milestones and deliverables. The project is divided into two major phases: documentation and implementation. The documentation phase spans from **6th October 2025 to 10th January 2026**, focusing on research, requirement gathering, system design, and final report preparation. The implementation phase begins on **12th January 2026 and continues until 18th May 2026**, during which the actual development, integration, and testing of the Hairlytic system will be completed. An agile approach will be followed, ensuring continuous feedback and progress tracking through milestones and sprints.

1.7.1 Work Breakdown Structure

A Work Breakdown Structure (WBS) provides a deliverable-based decomposition of the entire project scope, covering 100% of the project work including documentation, system design, development, testing, and deployment. The duties are distributed between two team members **Fizza** and **Minahil** ensuring balanced workload and clear accountability.

WBS Hierarchy

1. **Project Management**
   - 1.1 Work Breakdown Structure (WBS)
   - 1.2 Roles & Responsibility Matrix
   - 1.3 Change Control System

2. **Reports / Documentation**
   - 2.1 Final Documentation Introduction
   - 2.2 Literature Review / Market Survey
   - 2.3 Requirements Analysis
   - 2.4 System Design
   - 2.5 Implementation Plan
   - 2.6 Testing & Performance Evaluation Plan

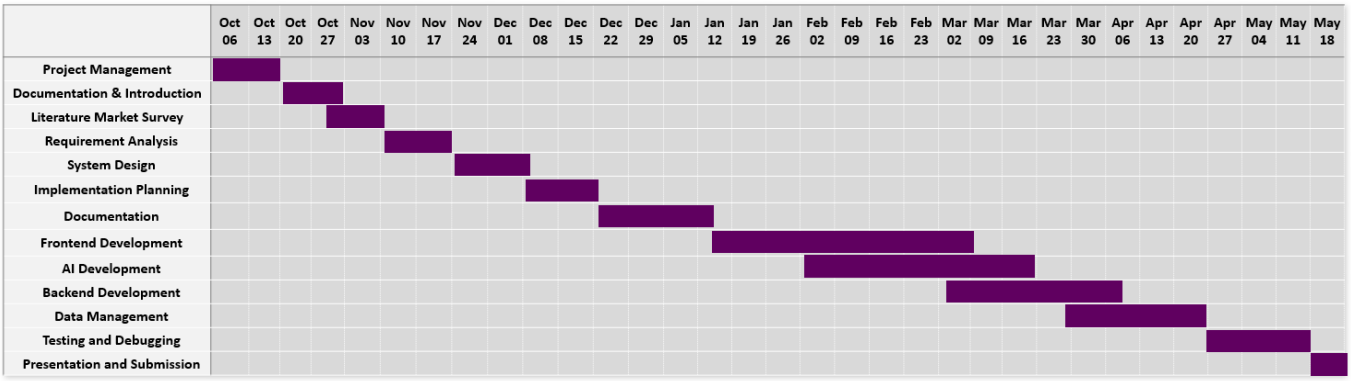
1.7.2. Roles & Responsibility Matrix

The purpose of roles & responsibility matrix is to identify who will do what.

#	WBS Deliverable	Activity #	Activity to Complete Deliverable	Duration (Days)	Responsible Team Member(s) & Role(s)
1	Project Management	1.1–1.3	Define WBS, roles, and change system	7	Minahil , Fizza
2	Documentation Introduction	2.1	Draft executive summary & objectives	14	Minahil
3	Literature / Market Survey	2.2	Collect and summarize previous studies & competitors	12	Minahil
4	Requirements Analysis	2.3	Define functional/non-functional requirements	14	Minahil & Fizza
5	System Design	2.4	Create architecture diagrams & design specifications	17	Fizza
6	Implementation Planning	2.5	Define module-based plan & milestones	14	Both
7	Documentation	3.1	Creating SRS and SDS Documents	23	Fizza
8	Frontend Development	3.2	Design UI, build responsive interface, user flow	60	Fizza
9	AI Development	3.3	Development and integrate AI	50	

10	Backend Development	3.3	Build APIs, integrate, secure backend	30	Fizza
11	Data Management	3.4	Schema creation, API linking, encryption	28	Both
12	Testing & Debugging	2.6	Perform unit, integration & system testing	20	Both
13	Final Documentation & Reports	2.7–2.10	Prepare final report, user & admin manuals	14	Minahil
14	Presentation & Submission	-	Prepare final presentation & demo	7	Both

1.7.3. Gantt Chart



1.8. Report Outline

Chapter 1: Introduction

Provides background on alopecia, motivations, project objectives, existing solutions, and the proposed Hairlytic platform.

Chapter 2: Software Requirements Specification (SRS)

Defines the functional and non-functional requirements, user roles, system features, and validation criteria.

Chapter 3: Use Case Analysis

Illustrates patient, dermatologist, and admin workflows using use cases, flowcharts, and role-specific interactions.

Chapter 4: System Design

Covers system architecture, database schema, UI/UX design, API integration, and security mechanisms.

Chapter 5: Implementation

Details the development environment, frontend/backend implementation, AI integration, at-home test simulation, and data management.

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1 Introduction

2.1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to define the functional and non-functional requirements for **Hairlytic Web Platform v1.0**. Hairlytic is an **AI-powered web application** designed to enable **at-home diagnosis and management of alopecia** through **live scalp image analysis**, **medical questionnaires**, **report uploads**, and **guided self-assessment tests**.The platform also features a **dermatologist recommendation module** that helps users locate **nearby certified clinics** for professional consultation.

This SRS serves as a guide for developers, testers, and stakeholders to understand the complete system behavior and scope for the initial **web-based release** of Hairlytic. Mobile app development and wearable integration are excluded from this phase.

2.1.2 Document Conventions

This Software Requirements Specification (SRS) for **Hairlytic** follows the **IEEE 830 SRS standard** to ensure that all requirements are documented in a clear, structured, and traceable manner. The IEEE format provides consistency across sections, making it easier for developers, testers, and stakeholders to understand the system requirements and design decisions.

The following conventions and standards are used throughout this document:

- **Bold text** is used for module names, headings, and key terms (e.g., *AI Detection Module*).
- Numbered headings (e.g., 1.1, 1.1.1, 3.2.4) indicate document hierarchy.
- Each requirement is assigned a unique ID such as **FR-01** (Functional Requirement) or **NFR-01** (Non-Functional Requirement).
- **Tables** are used to present structured information—such as requirement lists, priorities, and data relationships—for better readability and traceability.
- Requirement priorities are categorized as **High (H)**, **Medium (M)**, or **Low (L)**, helping developers plan implementation order.

Using the IEEE standard ensures that the **Hairlytic SRS** remains professional, consistent, and easy to maintain, while allowing future extensions such as new modules or system upgrades to be documented seamlessly.

2.1.3 Intended Audience and Reading Suggestions

The Hairlytic SRS is intended for developers, project managers, testers, end users, and documentation writers. Developers focus on **system features, interfaces, and non-functional requirements** for implementation. Project managers use the document to **monitor scope, plan tasks, and allocate resources**. Testers refer to **functional and non-functional requirements to design test cases**, while end users review the **product overview, user classes, and interfaces** to understand system capabilities. Documentation writers use it to prepare manuals and training materials. The SRS is organized into Introduction, Overall Description, External Interfaces, System Features, Non-functional Requirements, and Other Requirements, and readers can follow this sequence or consult relevant sections based on their role.

2.1.4 Product Scope

Hairlytic is an **AI-powered web platform** designed to assist users in the **at-home diagnosis and management of alopecia** through intelligent scalp image analysis, guided self-assessment tests, and data-driven recommendations. The system leverages **machine learning** and **image processing** to detect the type and severity of alopecia, providing users with accurate, accessible, and real-time insights without the need for physical consultations.

Beyond diagnosis, Hairlytic offers an integrated **dermatologist recommendation portal** that suggests **nearby certified clinics** for further consultation, bridging the gap between at-home evaluation and professional care.

The main objectives of Hairlytic are to:

- Provide users with an **accessible and affordable** way to detect and understand alopecia.
- Combine **AI-based diagnosis** with **guided self-assessment tests** to improve accuracy.
- Offer **personalized recommendations** for treatment and products.
- Enhance **user convenience** through an integrated health ecosystem.

By aligning with modern **digital health and wellness strategies**, Hairlytic contributes to early alopecia detection, personalized treatment support, and improved user confidence—ultimately promoting **scalable, AI-driven healthcare innovation** in the dermatology domain.

2.1.5 References

The following references were used in preparing this SRS and provide additional guidance on system design, AI implementation, and domain knowledge:

1. **Ahn, S. Y., & Lee, S. H. (2022).** *Advances in hair loss diagnosis using artificial intelligence: From image processing to deep learning applications.* Journal of Dermatological Science, 105(2), 95–103. https://doi.org/10.1016/j.jdermsci.2022.01.006
2. **Cheng, L., & Kim, H. J. (2021).** *AI-based scalp analysis: A modern approach to personalized hair care.* International Journal of Cosmetic Science, 43(4), 378–386. https://doi.org/10.1111/ics.12742
3. **DermEngine (2024).** *AI-powered scalp and hair analysis system.* Retrieved from https://www.dermengine.com
4. **HairScan AI (2024).** *Hair health assessment using machine learning and image recognition.* Retrieved from https://www.hairscan.ai
5. **WHO (2022).** *Global trends in dermatological conditions: Hair and scalp health report.* Retrieved from https://www.who.int/publications
6. **Pantene Research Institute (2023).** *The science of hair fall: Factors, patterns, and solutions.* Retrieved from https://www.pantene.com/science
7. **L'Oréal Research & Innovation (2023).** *AI in personalized beauty and hair care diagnostics.* Retrieved from https://www.loreal.com/en/research-innovation/
8. **Kumar, R., & Yadav, M. (2021).** *Machine learning for dermatology and scalp disorder detection.* IEEE Access, 9, 118743–118754. https://doi.org/10.1109/ACCESS.2021.3107812

These references provide technical, clinical, and AI-related foundations that guided the design, features, and implementation of Hairlytic.

2.2 Overall Description

2.2.1 Product Perspective

Hairlytic is a **self-contained, AI-powered web platform** developed to address the growing issue of hair loss and alopecia awareness among individuals. The system originated from the observation that many people experience severe hair fall but remain unaware of underlying conditions such as **alopecia**. Additionally, users often face difficulties finding **trusted dermatologists** for treatment.

Hairlytic bridges this gap by providing a unified solution where users can:

- Diagnose hair loss and identify the **type and severity of alopecia** through **AI-based image analysis**.
- Access **guided self-assessment tests** (e.g., Hair Pull, Tug, Hair Count).
- Receive **personalized dermatologist recommendations** and locate **nearby certified clinics** for professional consultation.

The system is completely **independent** and not a replacement for any existing software. It integrates multiple services: AI detection, medical data processing, and dermatologist recommendation within a single digital ecosystem to enhance accessibility, trust, and health awareness.

2.2.2 Product Functions

Hairlytic provides a range of AI-driven and user-interactive functions that enable users to diagnose, understand, and manage alopecia from home. The system integrates multiple modules that work together to provide accurate analysis, recommendations, and access to trusted hair-care solutions.

Major Functions:

- **User Registration & Authentication:**
  Allows users to create secure accounts, log in, and manage their profiles.
- **AI-Based Alopecia Detection:**
  Captures scalp images via the user's camera and analyzes them using trained AI models to detect the type and severity of alopecia.
- **Questionnaire & Report Upload Module:**
  Lets users fill out medical and lifestyle questionnaires and upload medical or dermatological reports to improve diagnostic accuracy.

- **At-Home Hair Test Simulation:**
  Guides users through simple camera-based self-tests (Hair Pull Test, Tug Test, and Hair Count Test) to assess hair strength and shedding activity.
- **Dermatologist Recommendation:**
  Suggests nearby certified dermatologists and clinics for users needing professional consultation or treatment.
- **Notification & Reporting System:**
  Sends diagnostic summaries, test results, and relevant treatment recommendations to users in real-time.

2.2.3 User Classes and Characteristics

Hairlytic is designed for a variety of user groups who interact with the system for different purposes, based on their needs, expertise, and privileges. The main user classes and their characteristics are described below:

1. General Users / Patients

- **Description:**
  Individuals experiencing hair loss or scalp issues who use Hairlytic for at-home alopecia detection and management.
- **Characteristics:**
  - Non-technical background; prefer simple, guided instructions.
  - Use features like AI scalp analysis, medical questionnaires, and at-home test simulations.
  - Can view personalized diagnostic reports, and nearby dermatologist options.
- **Privileges:**
  - Register, log in, upload images/reports, perform self-tests, view results, explore products, and book dermatologist appointments.

2. Dermatologists / Hair Specialists

- **Description:**
  Certified dermatologists who register on Hairlytic to offer consultation services. Users can search for dermatologists nearby and book appointments directly through the platform.
- **Characteristics:**
  - Medical professionals with verified credentials.
  - Use the platform to manage profiles, view appointment requests, and provide medical advice or treatment plans.
- **Privileges:**
  - Create professional accounts, set consultation timings, accept/reject appointments, and manage patient interactions.

3. System Administrator

- **Description:**
  Technical personnel responsible for system maintenance, security, and user management.
- **Characteristics:**
  - High-level access and technical expertise in web systems and database management.
- **Privileges:**
  - Full control over all system modules, user verification (especially dermatologists), Monitor and Manage Microservices, and AI updates.

2.2.4 Operating Environment

- **Platform:** Web-based AI application for hair and scalp analysis.
- **Client Devices:** Works on desktops, laptops, tablets, and smartphones (with camera).
- **Frontend:** React.js, HTML5, CSS3, Tailwind CSS.
- **Backend:** Python (Flask/FastAPI) with MongoDB database.
- **AI Tools:** TensorFlow, PyTorch, OpenCV.
- **Supported OS:** Windows, macOS, Android, iOS (browser-based).
- **Security:** OAuth 2.0.
- **Other Features:** Web camera access, cloud storage.

2.2.5 Design and Implementation Constraints

- **Software & Technology Constraints:**
  - Must use **Python (Flask/FastAPI)** for backend and **React.js** for frontend.
  - **MongoDB** is fixed as the primary database for storing user, product, and clinic data.
  - AI modules rely on **TensorFlow** or **PyTorch**, limiting flexibility in framework choice.
- **Integration Constraints:**
  - Must integrate with **Google Maps API** for nearby dermatologist recommendations.
- **Operational Constraints:**
  - Requires **constant internet connectivity** for AI processing and real-time updates.
- **Design & Standards:**
  - Must follow **IEEE SRS standards** for documentation and design structure.
  - Consistent **UI/UX** guidelines across all modules for usability and accessibility.

2.2.6 User Documentation

The following user documentation will be provided with the **Hairlytic** system to ensure ease of use and understanding for all users:

- **User Manual:** Step-by-step guide explaining app installation, account setup, and feature usage (AI detection, test performing and clinic booking).
- **Online Help Section:** Built-in FAQs and troubleshooting tips accessible within the app.
- **Video Tutorials:** Short demo videos showing how to scan the scalp, view AI results, and book appointments.

Delivery Format:
All documentation will be available in **digital format (PDF, HTML, and in-app help)** for easy access on **web.**

2.2.7 Assumptions and Dependencies

1. **Assumptions:**

- Users will have devices with a **camera** and **stable internet connection** for AI-based hair analysis.
- Dermatologists and clinics will **register and maintain accurate information** within the system.
- The **AI model** will perform accurately if trained with sufficient and diverse scalp image datasets.
- Users will provide **authentic data and images** for reliable diagnosis and recommendations.

2. **Dependencies:**

- Relies on **third-party APIs** such as **Google Maps** for clinic recommendations
- Dependent on **TensorFlow/PyTorch frameworks** for AI image processing.
- Requires **MongoDB** for secure data storage and retrieval.
- Application performance depends on **browser and OS compatibility** across devices.

2.3 External Interface Requirements

2.3.1 User Interfaces

The **Hairlytic application** offers a **highly user-friendly and responsive interface** tailored for both **end-users and dermatologists**, ensuring **consistency, accessibility, and clarity** throughout all screens while adhering to standard **UI/UX design principles**. The **Login & Signup screens** provide **secure authentication** via email or Google accounts, while the **Dashboard** prominently displays the **user profile**, **AI scan history**, and allows **quick navigation** to key features. The **AI Detection Screen** enables users to **upload scalp images or capture live photos** for **AI-based hair analysis**, presenting **personalized results and treatment suggestions**. The **Clinic Finder Module** offers a **map-based interface** to locate nearby dermatologists, displaying **ratings, distance, and appointment booking options**. For administrators, the **Admin Control Panel** allows **efficient management of users, clinics**. The interface is **fully responsive**, optimized across **mobile, tablet, and web platforms**, ensuring an intuitive and engaging experience for all users.

2.3.2 Hardware Interfaces

The Hairlytic web platform interacts with client and server hardware for AI-based scalp analysis, at-home tests, and dermatologist operations.

Logical Characteristics:

- Client devices capture images, inputs, and test data, sending them to the server for AI processing.
- Servers handle AI inference, data storage, returning results to clients.
- Real-time communication for image analysis and tests; asynchronous for notifications and reports.

Physical Characteristics:

- **Client Devices:** Desktops, laptops, tablets, smartphones with cameras.
- **Peripherals:** Camera, touchscreen, keyboard, mouse.

Data & Control Interactions:

- Client sends images and questionnaires to the server; server returns AI results, test feedback, and notifications.
- Hardware input (e.g., camera activation) triggers software processes.

2.3.3 Software Interfaces

Hairlytic interacts with various software components, databases, tools, and libraries to enable AI-based scalp analysis, at-home tests, and e-commerce.

- **Databases:** MongoDB stores user profiles, medical reports, AI results.
- **Operating Systems:** Compatible with Windows, macOS, Android, and iOS (browserbased).
- **Libraries & Tools:** TensorFlow/PyTorch for AI, OpenCV for image processing, React.js, Tailwind CSS for frontend, Flask/FastAPI for backend APIs.
- **Integrated Components:** Google Maps API for dermatologist locations.

Data Flow:

- **Incoming:** Scalp images, questionnaires, test results, report uploads, user input.
- **Outgoing:** AI analysis results, test evaluations, notifications, dermatologist recommendations.

2.3.4 Communications Interfaces

Hairlytic requires reliable communication between client devices, cloud servers, and external services.

- **Protocols & Standards:** HTTP/HTTPS for web communication, RESTful APIs for data exchange, OAuth 2.0 for secure authentication.
- **Data Types:** Scalp images, questionnaire responses, test results, notifications, and report files.
- **Message Formatting:** JSON format for API requests and responses, ensuring structured and consistent data transfer.
- **Network Requirements:** Stable internet connection with minimum broadband speed for real-time AI processing; asynchronous handling for notifications and report generation.
- **Synchronization:** Real-time updates for AI results and background sync for notifications and report uploads.

2.4. System Features

2.4.1 User Registration & Authentication

2.4.1.1 Description and Priority

This feature allows users to create accounts, log in securely, and manage their profiles. High priority, as it is essential for accessing the system and personalizing the user experience.

2.4.1.2 Stimulus/Response Sequences

- User opens the registration page → System displays form for email/Google login.
- User submits credentials → System verifies and creates account or returns error.
- User logs in → System authenticates and grants access to dashboard.

2.4.1.3 Functional Requirements

REQ-SF1-1: The system shall allow users to register using email or Google accounts.
REQ-SF1-2: The system shall validate credentials and prevent duplicate accounts.
REQ-SF1-3: The system shall enable users to reset passwords and update profile information.

2.4.2 AI-Based Alopecia Detection

2.4.2.1 Description and Priority

This feature enables users to capture scalp images via their camera and receive AI-based analysis for alopecia detection, type, and severity. High priority due to core diagnostic functionality.

2.4.2.2 Stimulus/Response Sequences

- User opens AI Detection screen → System prompts for camera access.
- User positions head as guided → System captures images and sends them to AI module.
- AI processes images → System returns diagnosis and severity results to user.

2.4.2.3 Functional Requirements

REQ-SF2-1: The system shall capture live scalp images using the device camera.
REQ-SF2-2: The system shall process images using trained ML models to detect alopecia type and severity.
REQ-SF2-3: The system shall display real-time diagnostic feedback and suggestions.

2.4.3 Questionnaire & Medical Report Upload

2.4.3.1 Description and Priority

Allows users to fill questionnaires and upload medical reports to improve diagnostic accuracy. High priority for enhancing AI precision.

2.4.3.2 Stimulus/Response Sequences

- User navigates to questionnaire → System displays forms for medical and lifestyle data.
- User uploads reports → System stores files securely and uses data in AI analysis.
- System returns updated diagnostic results considering questionnaire and reports.

2.4.3.3 Functional Requirements

REQ-SF3-1: The system shall provide forms for medical and lifestyle questionnaires.
REQ-SF3-2: The system shall allow uploading of lab or dermatology reports (PDF, JPG, PNG).
REQ-SF3-3: The system shall integrate questionnaire and report data to improve AI results.

2.4.4 At-Home Hair Test Simulation

2.4.4.1 Description and Priority

Guides users through self-assessment tests (Hair Pull, Tug, Hair Count) via camera for hair strength and shedding evaluation. High priority for user engagement and detailed diagnostics.

2.4.4.2 Stimulus/Response Sequences

- User selects a hair test → System shows instructions and live camera view.
- User performs test → System analyzes video/images and computes test metrics.
- System displays test evaluation along with AI analysis.

2.4.4.3 Functional Requirements

REQ-SF4-1: The system shall guide users through Hair Pull, Tug, and Hair Count tests.
REQ-SF4-2: The system shall analyze captured images/videos to assess hair condition.
REQ-SF4-3: The system shall provide a report of test results and recommendations.

2.4.5 Recommendation & Dermatologist Module

2.4.5.1 Description and Priority

Recommends certified dermatologists or nearby clinics and allows appointment booking. Medium priority for professional support.

2.4.5.2 Stimulus/Response Sequences

- User requests dermatologist → System searches database and returns nearby certified specialists.
- User selects a doctor → System allows appointment booking and confirms with notifications.

2.4.5.3 Functional Requirements

REQ-SF5-1: The system shall provide a list of nearby certified dermatologists.
REQ-SF5-2: The system shall allow users to book and confirm appointments.
REQ-SF5-3: The system shall send notifications for appointment confirmation and reminders.

2.4.6 Notification & Reporting System

2.4.6.1 Description and Priority

Sends diagnostic summaries, test results, and dermatologist recommendations. High priority for keeping users informed.

2.4.6.2 Stimulus/Response Sequences

- System generates diagnostic report → Sends notification to user.
- User receives notification → Can view report and dermatolgist suggestions.

2.4.6.3 Functional Requirements

REQ-SF7-1: The system shall send real-time notifications for diagnostic results.
REQ-SF7-2: The system shall deliver dermatologist recommendations to patients.
REQ-SF7-3: The system shall maintain a history of notifications and reports for reference.

2.5. Other Non-functional Requirements

2.5.1 Performance Requirements

- **Image Processing:** AI-based scalp image analysis must process and return diagnostic results within **5–10 seconds** for standard images (~1–2 MB).
- **Concurrent Users:** The system must support at least **1,000 concurrent users** without significant performance degradation.
- **Response Time:** All page navigations and dashboard updates should load within **3 seconds** on devices
- **Scalability:** The system must allow horizontal scaling on cloud infrastructure (AWS/Google Cloud) to maintain performance during traffic spikes.

2.5.2 Safety Requirements

- **Data Safety:** Users' personal, medical, and image data must be protected against accidental loss. Regular cloud backups must be performed daily.
- **Operational Safety:** AI-based test guidance must include clear instructions to prevent misuse (e.g., excessive hair pulling causing scalp damage).
- **Medical Disclaimer:** The system must display a disclaimer stating that AI diagnosis is **supportive only** and does not replace professional medical advice.
- **Regulatory Compliance:** The platform should comply with local and international health data protection regulations (e.g., HIPAA, GDPR for applicable regions).

2.5.3 Security Requirements

- **Authentication:** User authentication must use **OAuth 2.0**, enforcing strong passwords and optional two-factor authentication (2FA).
- **Access Control:**
  - General Users can access their own profiles, AI results.
  - Dermatologists can access only patient appointment requests and consultation data.
  - Admins have full control for maintenance, user verification, and AI updates.
- **Privacy Compliance:** The system must adhere to GDPR or equivalent privacy policies, including user consent for data collection and processing.

2.5.4 Software Quality Attributes

- **Usability:** The UI/UX must be intuitive for non-technical users, with guided instructions for AI scans and hair tests.
- **Reliability:** The system should maintain **99% uptime**, with automatic failover in case of server failure.
- **Portability:** The web platform must work seamlessly on modern browsers (Chrome, Edge, Safari, Firefox) across desktop, tablet, and mobile devices.
- **Maintainability:** Modular code design using **React.js (frontend)** and **Python/Flask (backend)** ensures maintainable updates and future feature integration.
- **Interoperability:** Integration with **Google Maps API** for clinic recommendations.
- **Testability:** All functional modules must be unit-testable, with automated test scripts for AI, database operations, and front-end navigation.
- **Robustness:** The system must handle invalid inputs, network interruptions, and partial failures without crashing.

2.5.5 Business Rules

- Users must **register** before accessing AI diagnostics, or appointment booking.
- Only **verified dermatologists** can appear in clinic recommendations and accept patient appointments.
- Admins approve new user and dermatologist accounts to ensure system trustworthiness.
- AI analysis results cannot be edited by users; only authorized system updates or retraining of AI models can modify diagnostic logic.

2.6 Other Requirements

- **Database Requirements:**
  - MongoDB will be used for storing user data, AI results, clinic information.
  - Backups must occur daily and must support point-in-time recovery.
- **Internationalization Requirements:**
  - The system should support multiple languages (starting with English) with easy extensibility for future language packs.
- **Legal Requirements:**
  - Comply with local medical and data privacy regulations in operating regions.
  - Display terms of service and consent forms for medical data.
- **Reuse Objectives:**
  - AI image processing and at-home test simulation modules should be designed for reuse in future mobile app or wearable integrations.

Chapter 3

Use Case Analysis

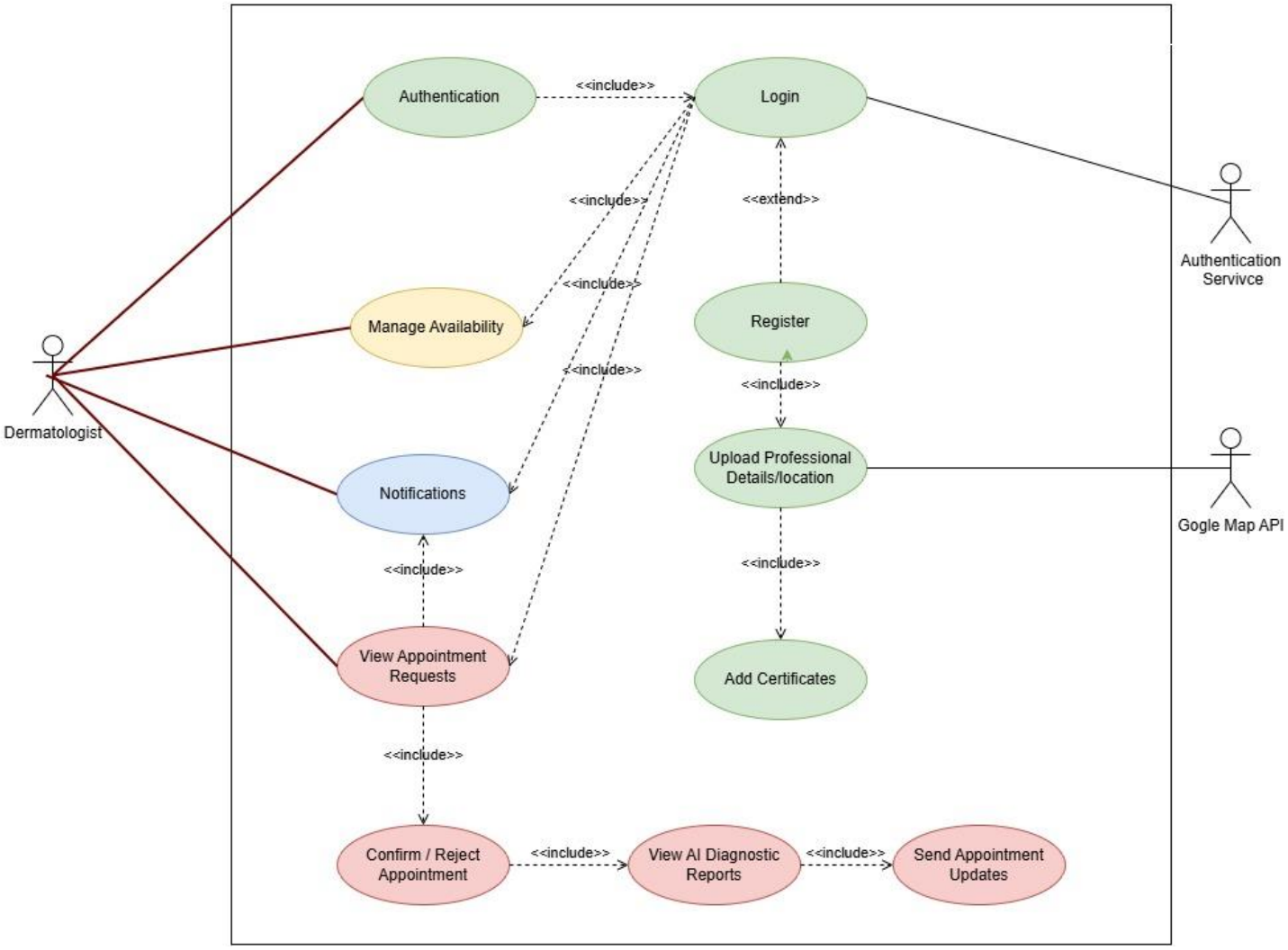
Chapter 3: System Analysis

Chapter 3 explains how the Hairlytic system works by breaking the project into smaller, understandable parts. It includes the use case model, which shows what different users can do in the system and how they interact with it. This chapter also presents fully dressed use cases that describe each feature in detail, including actors, steps, and expected results. The purpose of this analysis is to understand the system's functionality before development begins. By studying user needs, actions, and system behavior, this chapter provides a clear foundation for design and implementation.

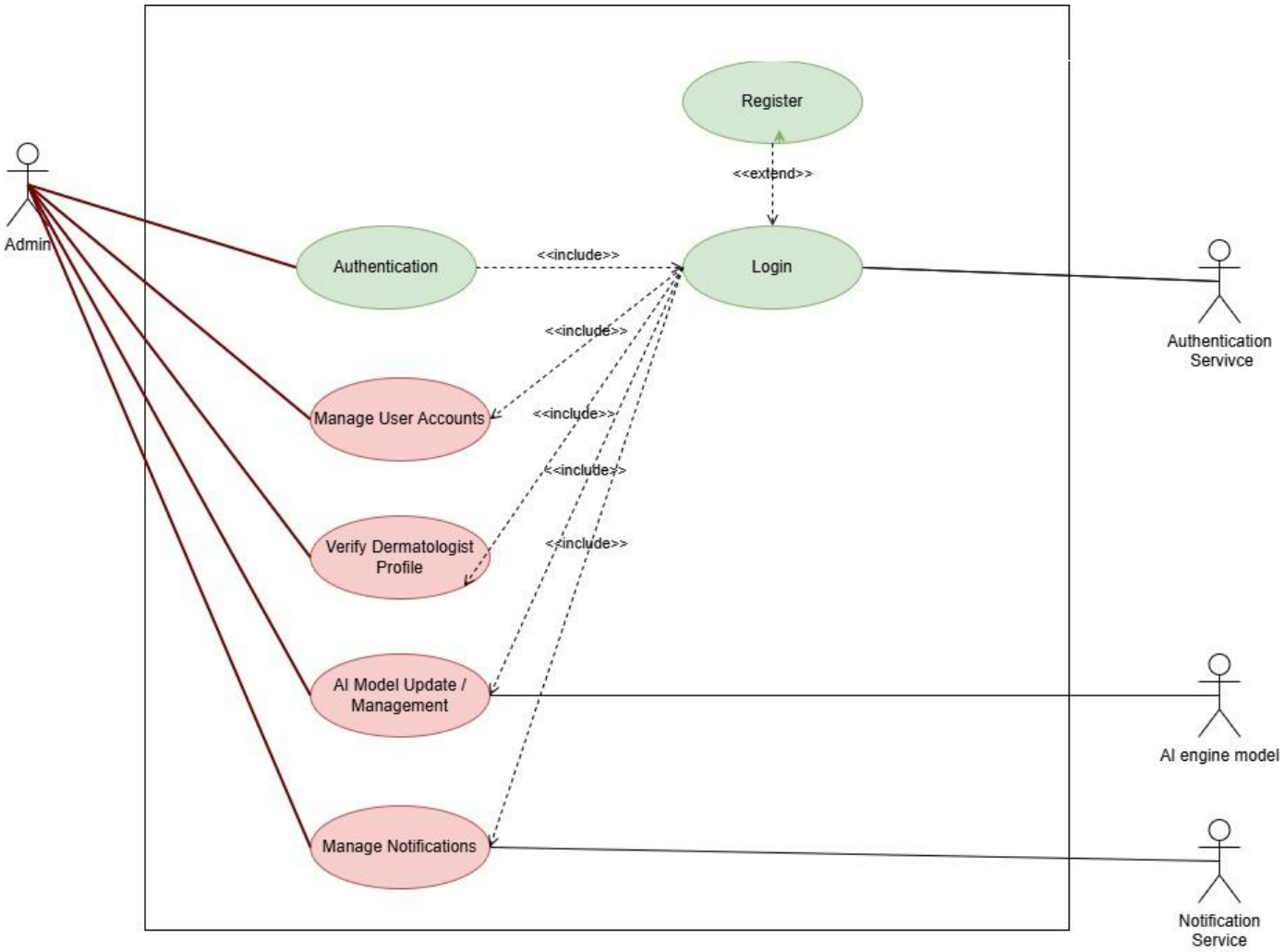
3.1. Use Case Model

Role: User

Role: Dermatologist



Role: Admin


Register
<<extend>>
Admin
Authentication
<<include>>
Login
Authentication Servivce
<<include>>
Manage User Accounts
<<include>>
<<include>>
Verify Dermatologist Profile
<<include>>
AI Model Update / Management
AI engine model
Manage Notifications
Notification Service

3.2. Fully Dressed Use Cases

Use Case 1: Register Account

Attribute	Description
Actors	General User
Trigger	User wants to create a new HairLytic account.
Preconditions	User is on the registration page; internet connection is available.
Postconditions	Account is created and verified; user can log in.
Normal Flow	1. User opens registration page. 2. User enters name, email, password, and other required details. 3. System validates input and checks for duplicates. 4. System sends verification email. 5. User clicks verification link. 6. System confirms registration.
Alternate Flows	3A1: Email already exists → System displays error message. 4A1: Verification email fails → System allows resending email.
Summary	Enables users to securely create an account to access HairLytic features.

Use Case 2: Login

Attribute	Description
Actors	General User
Trigger	User wants to access their HairLytic account.
Preconditions	User has a registered account; system is accessible.
Postconditions	User is logged in and navigated to dashboard.
Normal Flow	1. User opens login page. 2. User enters email and password or uses Google login. 3. System validates credentials. 4. User is redirected to dashboard.
Alternate Flows	2A1: Forgot password → System provides reset option. 3A1: Invalid credentials → System shows error message.
Summary	Provides secure access to HairLytic features for registered users.

Use Case 3: Capture Scalp Image / Perform AI Detection

Attribute	Description
Actors	General User
Trigger	User wants AI-based diagnosis for alopecia.
Preconditions	User is logged in; device camera is available; internet is active.
Postconditions	AI processes image; diagnosis results are stored and displayed.
Normal Flow	1. User navigates to AI Detection module. 2. System prompts camera access. 3. User captures scalp image. 4. System sends image to AI engine. 5. AI analyzes and returns type/severity of alopecia. 6. System displays results and recommendations.
Alternate Flows	3A1: Camera unavailable → System allows upload from gallery. 5A1: Image unclear → System prompts retake.
Summary	Provides real-time, AI-based scalp analysis for personalized diagnostic feedback.

Use Case 4: Fill Lifestyle Questionnaire / Upload Medical Reports

Attribute	Description
Actors	General User
Trigger	User wants to enhance AI diagnosis by providing lifestyle and medical data.
Preconditions	User is logged in.
Postconditions	Data is stored securely and integrated with AI analysis.
Normal Flow	1. User opens Questionnaire or Report Upload module. 2. User fills questionnaire or uploads PDF/JPG/PNG report. 3. System validates data and stores securely. 4. AI incorporates data into analysis.
Alternate Flows	2A1: Unsupported file format → System rejects upload. 3A1: Incomplete questionnaire → System prompts completion.
Summary	Improves accuracy of AI-based diagnostics using user-provided data.

Use Case 5 : Hair Pull Test

Attribute	Description
Actors	Patient, System, Dermatologist
Trigger	Patient experiences hair loss or thinning and wants diagnostic analysis.
Preconditions	Patient is logged in and has scheduled a diagnostic session.
Postconditions	Number of hairs pulled is recorded; results available in diagnostic report.
Normal Flow	1. Dermatologist selects Hair Pull Test in the system.2. Dermatologist instructs patient on procedure.3. Dermatologist gently pulls a few hair strands from affected area.4. System records the count of hairs pulled and severity based on predefined threshold.5. Results stored in patient's diagnostic report.6. System updates patient profile with test outcome.
Alternate Flows	3A: Patient experiences discomfort – Dermatologist pauses test.4A: Hair count exceeds threshold – System flags as abnormal for further consultation.4B: Hair count below threshold – System marks as normal.
Summary	Detects hair shedding severity by counting the number of hairs pulled from scalp to aid in diagnosis.

Use Case 6 – Hair Tug Test

Attribute	Description
Actors	Patient, Dermatologist, System
Trigger	Patient complains of fragile hair or suspected hair breakage.
Preconditions	Patient is logged in, diagnostic session scheduled, hair prepared (clean, dry).
Postconditions	Hair elasticity/flexibility results are stored in system for further recommendations.
Normal Flow	1. Dermatologist selects Hair Tug Test in the system.2. Dermatologist isolates a small section of hair.3. Gently tugs hair to assess elasticity and breakage resistance.4. Dermatologist observes hair behavior (stretches, breaks, snaps back).5. System records results as flexible, brittle, or abnormal.6. Test results added to patient's diagnostic report.7. System updates recommendations based on hair condition.
Alternate Flows	3A: Hair breaks easily – System flags severe fragility.3B: Hair stretches normally – System records as healthy.4A: Patient experiences pain – Test stopped, session noted.
Summary	Assesses hair strength and elasticity to identify hair fragility and potential underlying conditions.

Use Case 7– Perform Hair Count Test

Attribute	Description
Actors	General User
Trigger	User wants to check hair shedding activity using Hair Count Test.
Preconditions	User is logged in and has camera access.
Postconditions	Test is performed and results are stored for AI analysis.
Normal Flow	1. User navigates to Hair Count Test. 2. System displays instructions and live camera view. 3. User follows steps and captures hair count images. 4. System processes images and updates diagnostic data.
Alternate Flows	3A: Camera not available – System displays error message. 3B: Blurry images – System prompts user to retake images.
Summary	Enables users to monitor hair shedding quantitatively for improved diagnosis.

Use Case 8 – View Diagnostic Report

Attribute	Description
Actors	General User
Trigger	User wants to view AI and test-based diagnostic results.
Preconditions	User has completed AI detection or hair tests.
Postconditions	User views full diagnostic summary with recommendations.
Normal Flow	1. User selects “View Diagnostic Report.” 2. System retrieves AI and test results. 3. Report is displayed including alopecia type, severity, and suggestions.
Alternate Flows	2A: Report not available – System informs the user to complete tests first.
Summary	Consolidates AI and hair test results into a single user-friendly report.

Use Case 9– Book Dermatologist Appointment

Attribute	Description
Actors	General User, Dermatologist
Trigger	User wants to schedule a consultation.
Preconditions	Dermatologist profiles and time slots are available.
Postconditions	Appointment is confirmed and notifications sent.
Normal Flow	1. User searches and selects dermatologist. 2. System shows available slots. 3. User selects slot and books. 4. System confirms booking and sends notification.
Alternate Flows	2A: Slot unavailable – System prompts alternative slots. 4A: Dermatologist rejects appointment – System updates status.
Summary	Enables secure and convenient scheduling of dermatologist consultations.

Use Case 10 – View Near Clinics

Attribute	Description
Actors	General User, Google Maps API
Trigger	User wants to find nearby certified dermatologists.
Preconditions	User location is shared / accessible.
Postconditions	List of nearby clinics displayed on map.
Normal Flow	1. User opens “Nearby Clinics.” 2. System fetches location and queries Google Maps API. 3. Clinics are displayed with distance and ratings.
Alternate Flows	2A: Location unavailable – System asks for permission. 3A: No clinics nearby – System displays message.
Summary	Helps users locate dermatologists for physical consultation.

Use Case 11– Receive Notifications

Attribute	Description
Actors	General User, Notification Service
Trigger	System sends updates regarding AI results, appointments.
Preconditions	User is registered and logged in.
Postconditions	Notifications received in-app and/or via email.
Normal Flow	1. System generates notification. 2. Notification service delivers it to the user. 3. User views notification in dashboard or email.
Alternate Flows	2A: Delivery fails – System retries or queues notification.
Summary	Keeps users informed in real-time about relevant updates.

Use Case 12 – Login (Dermatologist)

Attribute	Description
Actors	Dermatologist
Trigger	Dermatologist wants to access the system.
Preconditions	Dermatologist has registered credentials.
Postconditions	Dermatologist is authenticated and redirected to dashboard.
Normal Flow	1. Dermatologist enters email/username and password.2. System verifies credentials.3. Access granted and dashboard displayed.
Alternate Flows	2A: Invalid credentials – System displays error message.2B: Forgot password – System provides reset option.
Summary	Provides secure access to dermatologist functionality.

Use Case 13 – Manage Availability / Time Slots

Attribute	Description
Actors	Dermatologist
Trigger	Dermatologist wants to update available consultation slots.
Preconditions	Dermatologist is logged in.
Postconditions	Updated slots reflected for patient booking.
Normal Flow	1. Dermatologist navigates to availability settings.2. Adds, updates, or removes slots.3. System saves changes and updates calendar.
Alternate Flows	2A: Overlapping slots – System prompts to correct.
Summary	Ensures patients can book appointments only at valid times.

Use Case 14 – View Appointment Requests

Attribute	Description
Actors	Dermatologist
Trigger	New appointment requests are received.
Preconditions	Dermatologist has active time slots.
Postconditions	Requests are displayed for action.
Normal Flow	1. Dermatologist opens “Appointment Requests.”2. System fetches pending requests.3. Requests displayed with patient details.
Alternate Flows	2A: No pending requests – System shows “No new requests.”
Summary	Centralizes pending appointments for dermatologist review.

Use Case 15 – Confirm / Reject Appointment

Attribute	Description
Actors	Dermatologist, Patient
Trigger	Dermatologist wants to manage appointment requests.
Preconditions	Appointment requests exist.
Postconditions	Appointment status updated and patient notified.

Attribute	Description
Normal Flow	1. Dermatologist selects request.2. Chooses Confirm or Reject.3. System updates status and sends notification.
Alternate Flows	2A: Slot no longer available – System suggests rescheduling.2B: Patient cancels – System updates automatically.
Summary	Allows dermatologists to manage appointments efficiently.

Use Case 16– View Uploaded Medical Reports

Attribute	Description
Actors	Dermatologist
Trigger	Patient uploads medical reports for consultation.
Preconditions	Reports have been uploaded by patients.
Postconditions	Dermatologist can access and review reports.
Normal Flow	1. Dermatologist navigates to “Medical Reports.”2. System displays patient-uploaded files.3. Dermatologist opens and reviews files.
Alternate Flows	2A: No reports uploaded – System displays message.
Summary	Provides a centralized location to view patient medical history.

Use Case 17 – View AI Diagnostic Reports

Attribute	Description
Actors	Dermatologist
Trigger	Dermatologist wants to see AI hair/scalp analysis results.
Preconditions	Patient has completed AI diagnostic test.
Postconditions	AI reports accessible to dermatologist.
Normal Flow	1. Dermatologist selects patient.2. System fetches AI report.3. Report displayed with insights and severity levels.
Alternate Flows	2A: Report not ready – System notifies dermatologist.
Summary	Supports informed decision-making for dermatologists.

Use Case 18 – Send Appointment Updates (Notifications)

Attribute	Description
Actors	Dermatologist, Patient
Trigger	Appointment is confirmed, rejected, or updated.
Preconditions	Appointment exists in the system.
Postconditions	Patient receives real-time notification.
Normal Flow	1. Dermatologist updates appointment.2. System generates notification.3. Notification delivered to patient.
Alternate Flows	2A: Notification fails – System retries delivery.
Summary	Ensures communication between dermatologists and patients.

Use Case 19 – Login (Admin)

Attribute	Description
Actors	Admin
Trigger	Admin wants access to the system.
Preconditions	Admin has credentials.
Postconditions	Admin authenticated and dashboard accessible.
Normal Flow	1. Admin enters credentials.2. System verifies info.3. Access granted and Admin dashboard displayed.
Alternate Flows	2A: Invalid credentials – Error displayed.
Summary	Provides secure access to administrative features.

Use Case 20 – Manage User Accounts

Attribute	Description
Actors	Admin
Trigger	Admin wants to add, update, or delete users.
Preconditions	Users exist in the system.
Postconditions	User accounts updated as per actions.
Normal Flow	1. Admin navigates to Users section.2. System displays all accounts.3. Admin adds/edits/deletes user.4. System updates database.
Alternate Flows	3A: Invalid data – System prompts correction.
Summary	Maintains control over registered users.

Use Case 21 – Verify Dermatologist Profile

Attribute	Description
Actors	Admin
Trigger	Admin wants to verify credentials of registered dermatologists.
Preconditions	Dermatologist signed up and uploaded credentials.
Postconditions	Profile approved or rejected.
Normal Flow	1. Admin opens dermatologist list.2. Reviews credentials.3. Approves or rejects profile.4. System updates status and notifies dermatologist.
Alternate Flows	2A: Missing documents – System prompts dermatologist to upload.
Summary	Ensures only verified dermatologists appear to patients.

Use Case 22– AI Model Update / Management

Attribute	Description
Actors	Admin
Trigger	Admin wants to update AI model used for hair/scalp diagnosis.
Preconditions	AI model exists in system.
Postconditions	Model updated and operational.
Normal Flow	1. Admin uploads new model version.2. System validates and deploys model.3. Model becomes active for diagnostics.
Alternate Flows	2A: Validation fails – Admin notified, rollback applied.
Summary	Ensures AI model stays accurate and up-to-date.

Use Case 23 – Manage Notifications

Attribute	Description
Actors	Admin
Trigger	Admin wants to configure system-wide notifications.
Preconditions	Notification service active.
Postconditions	Notifications settings updated.
Normal Flow	1. Admin selects notification type (user, dermatologist).2. Updates content or delivery rules.3. System applies settings.
Alternate Flows	2A: Delivery configuration invalid – System prompts correction.
Summary	Ensures timely, accurate notifications for all actors.
Alternate Flows	2A: Invalid configuration – System prompts correction.
Summary	Enables control over system-wide configurations and policies.

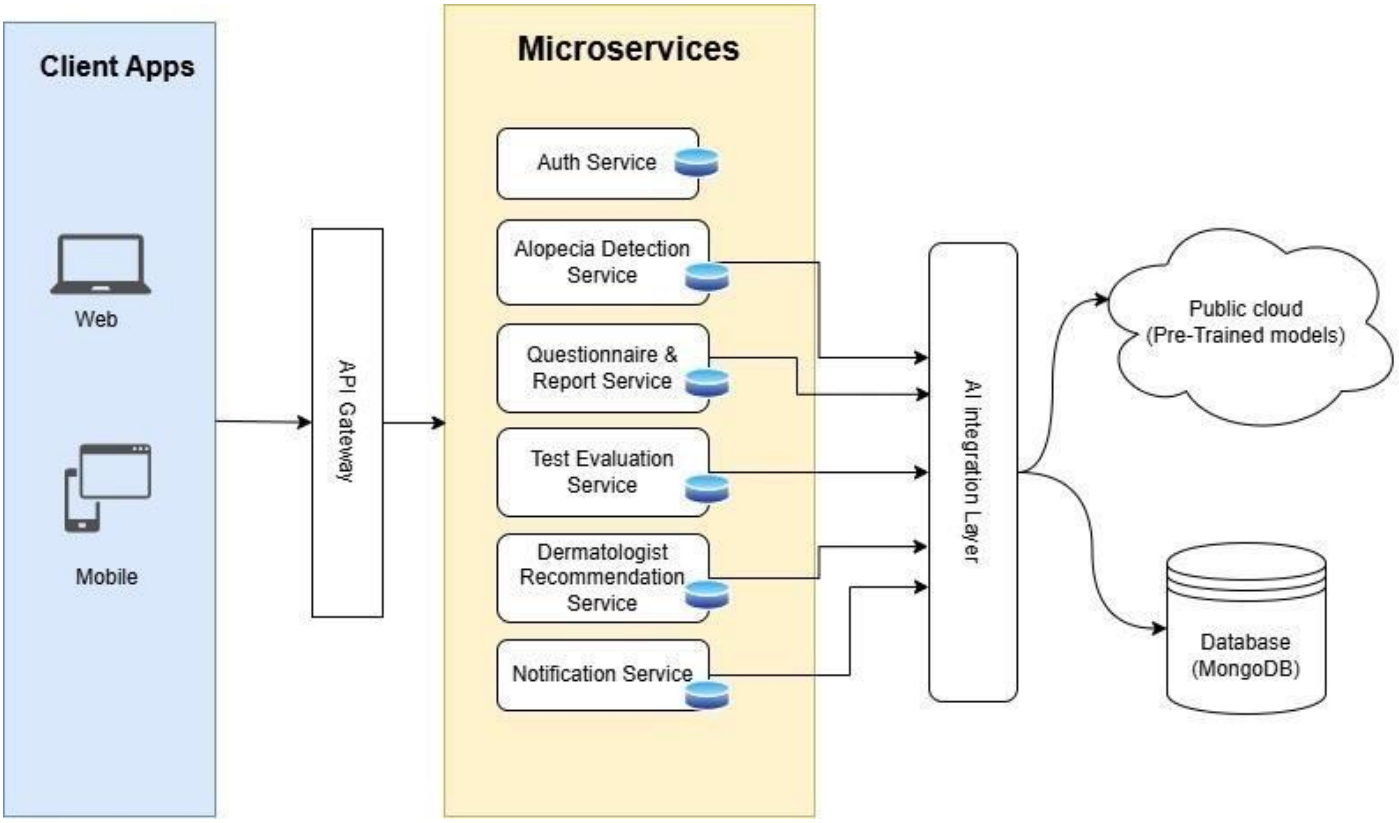
Chapter 4

System Design

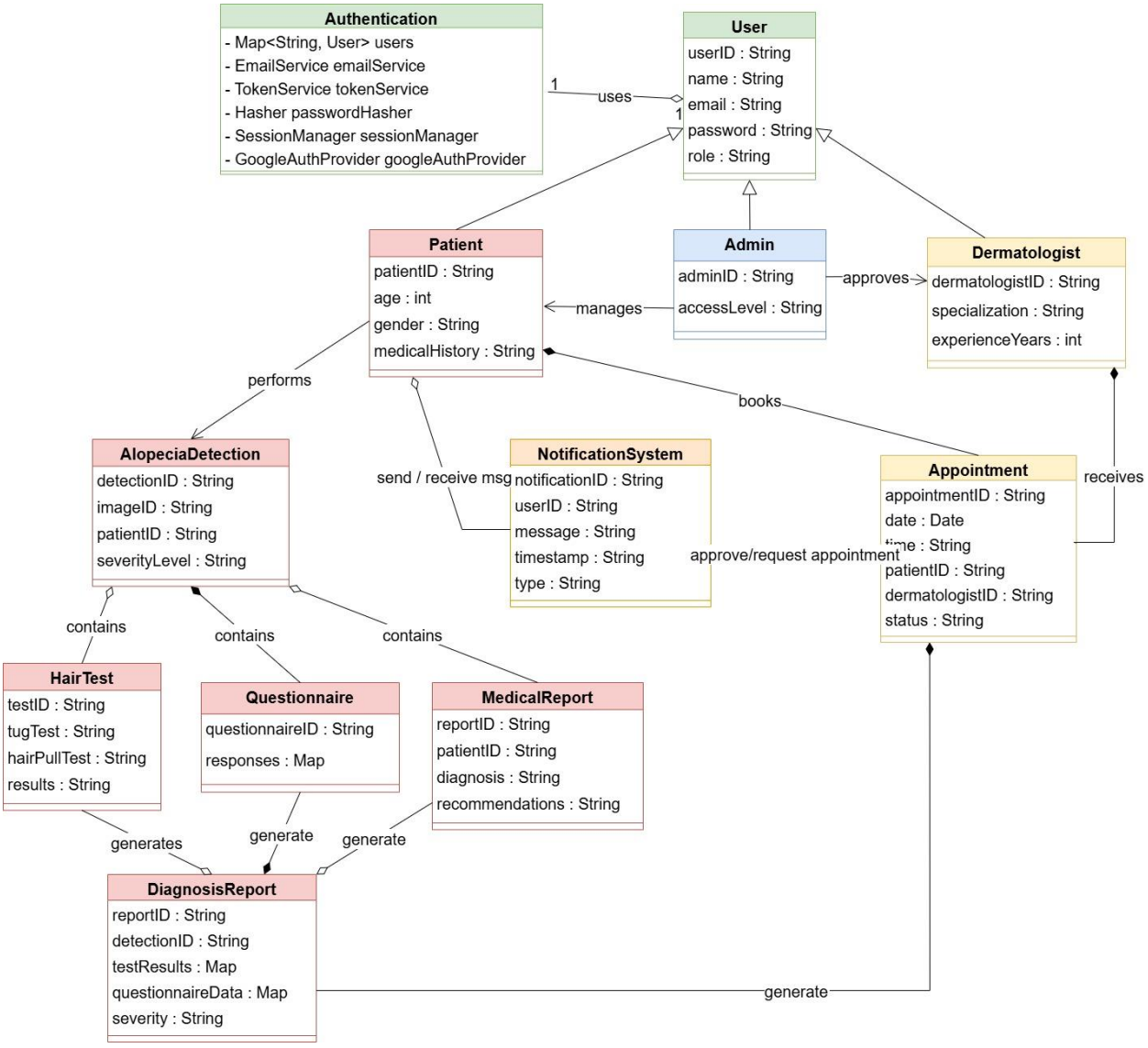
Chapter 4: System Design

Chapter 4 focuses on how the HairLytic system is structured and how its components work together. This chapter includes architectural diagrams, Class Diagram, workflow diagrams, and more. It translates the system requirements from the analysis phase into a blueprint that guides the development process. The goal of this chapter is to show how each module is organized, how data flows between them, and how the system will operate efficiently. This design lays the foundation for secure, flexible, and scalable implementation.

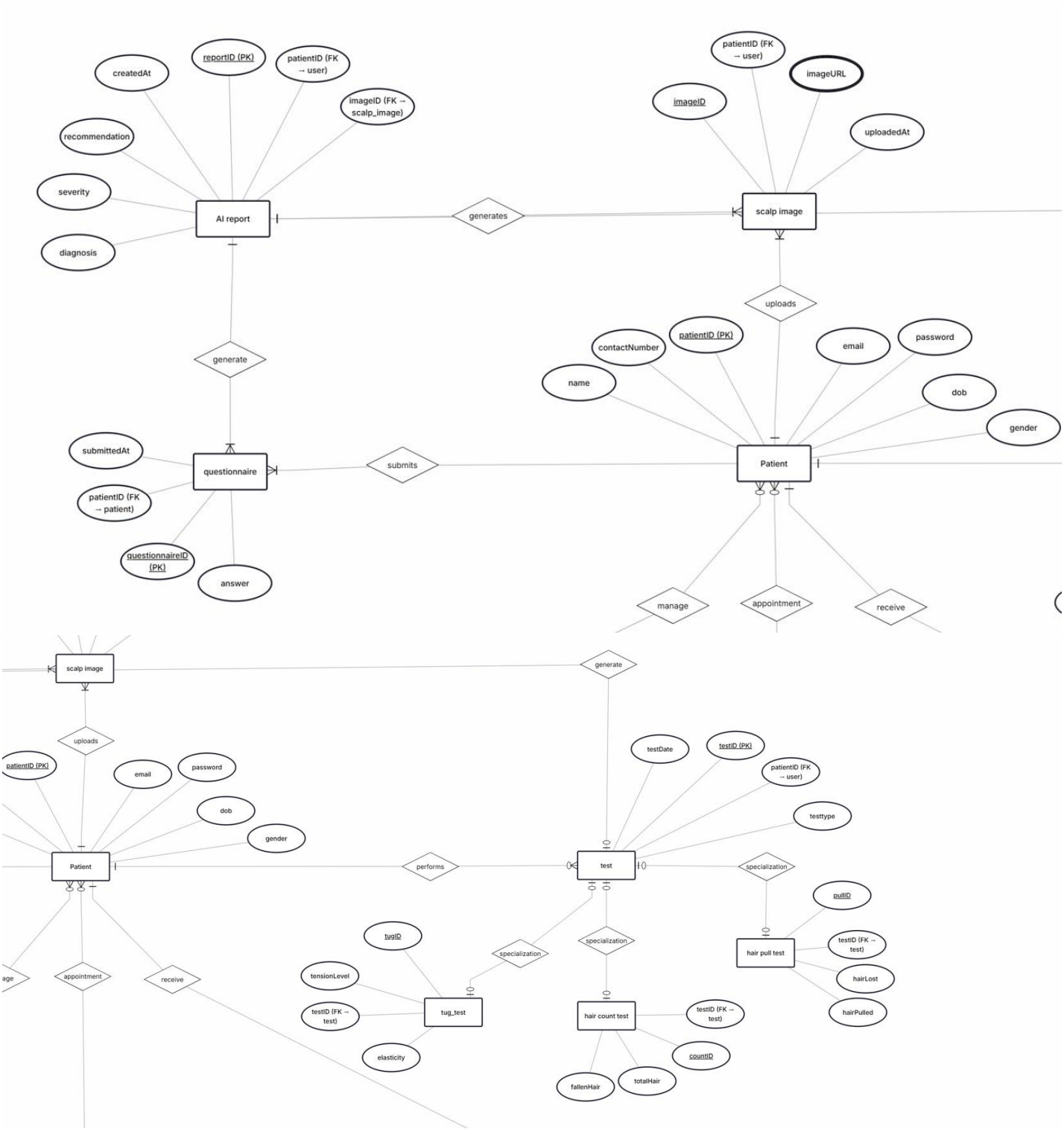
4.1. Architecture Diagram

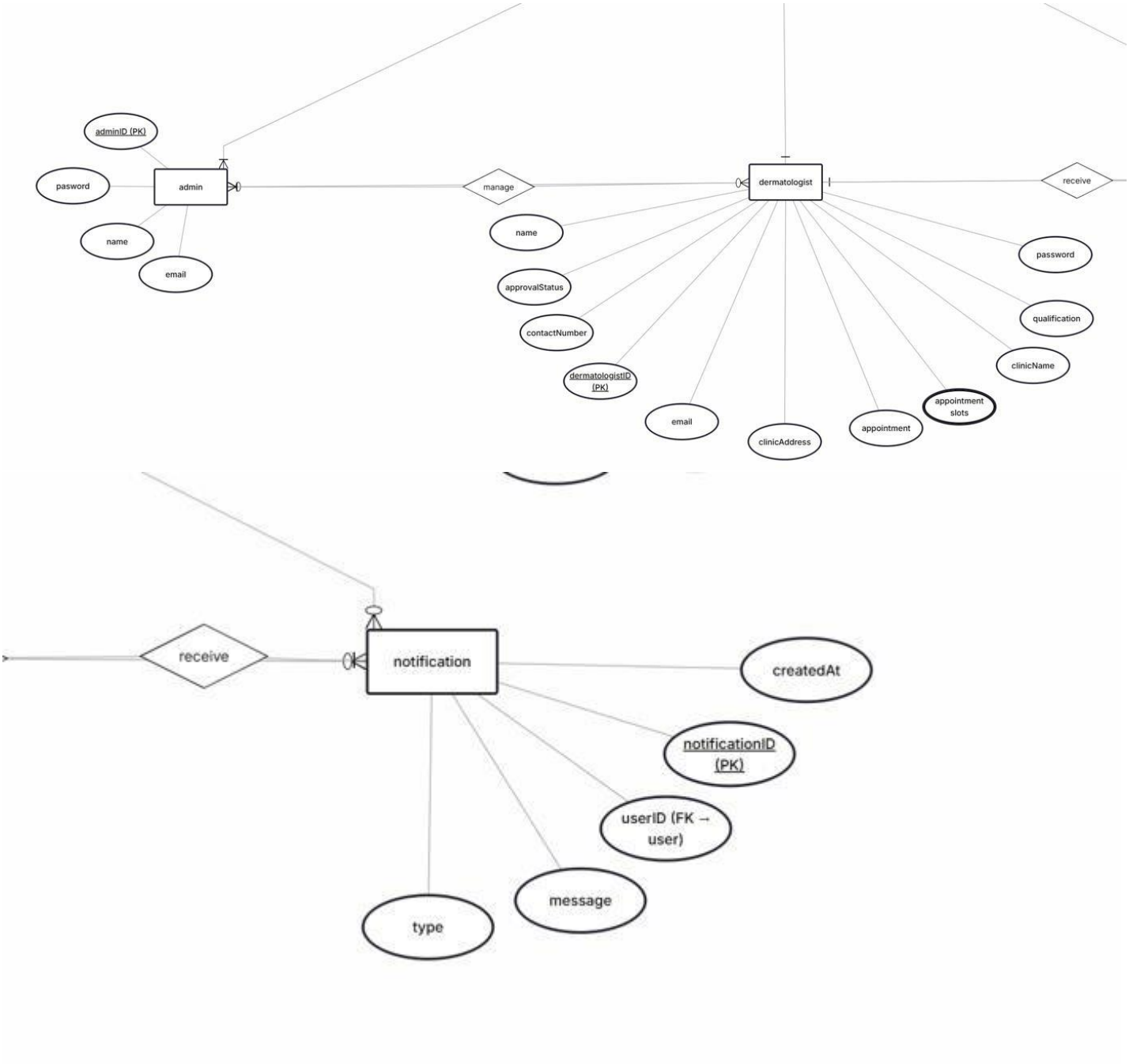


4.2. Domain Model

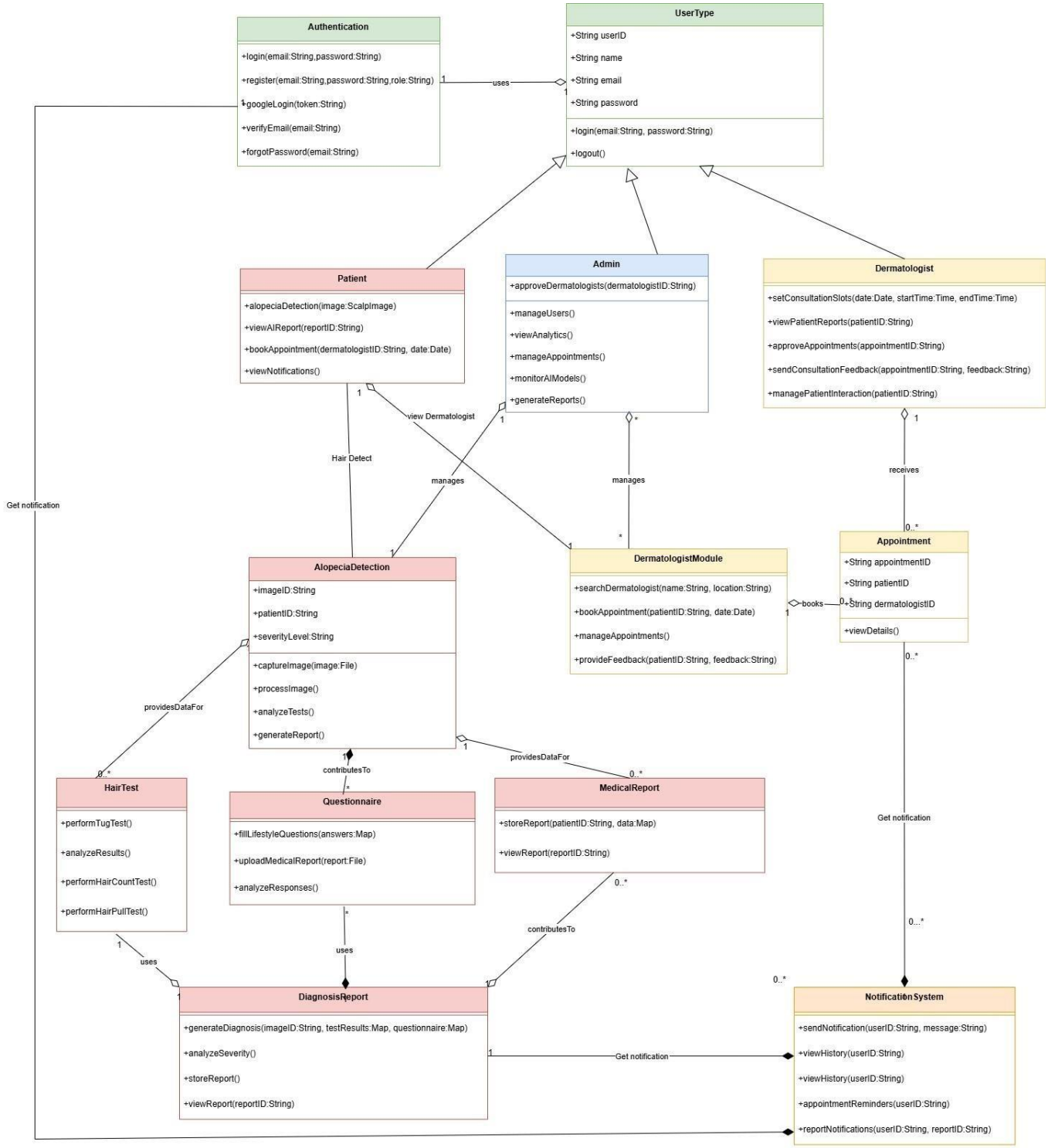


4.3. Entity Relationship Diagram with data dictionary



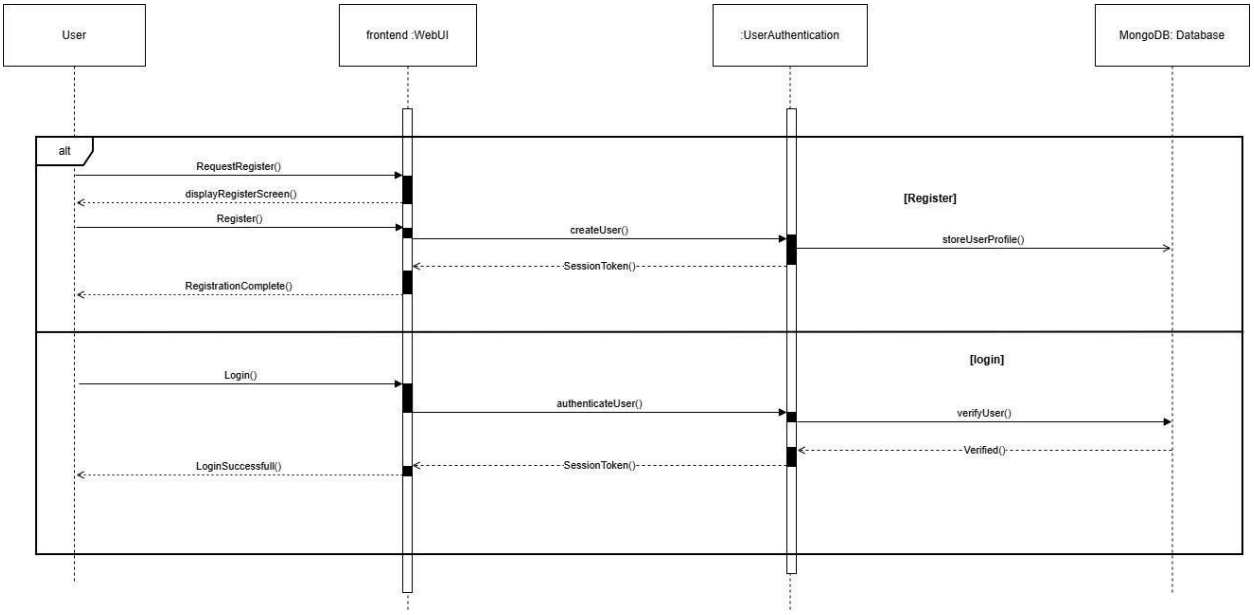

adminID (PK)
pasword
admin
name
email
manage
dermatologist
receive
name
approvalStatus
contactNumber
dermatologistID (PK)
email
clinicAddress
appointment
appointment slots
password
qualification
clinicName
receive
notification
createdAt
notificationID (PK)
userID (FK → user)
message
type

4.4. Class Diagram

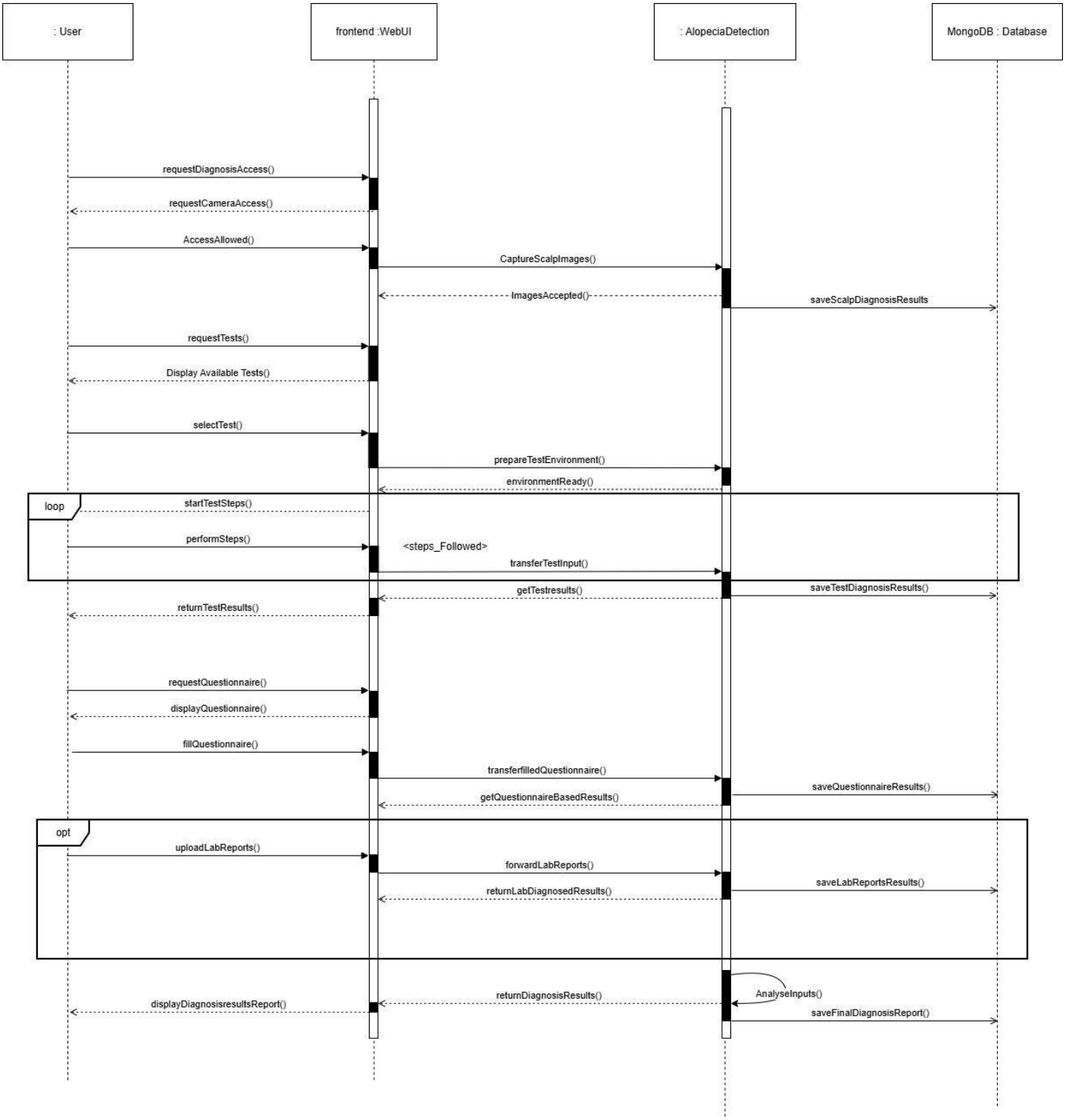


4.5. Sequence / Collaboration Diagram

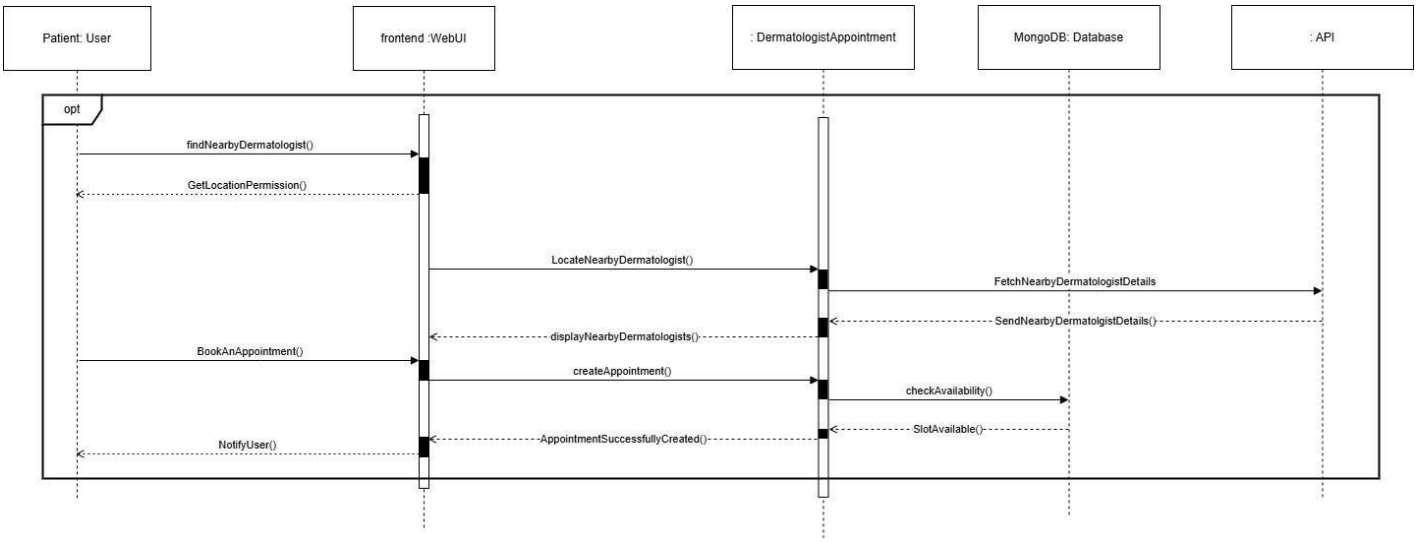
Login/Register



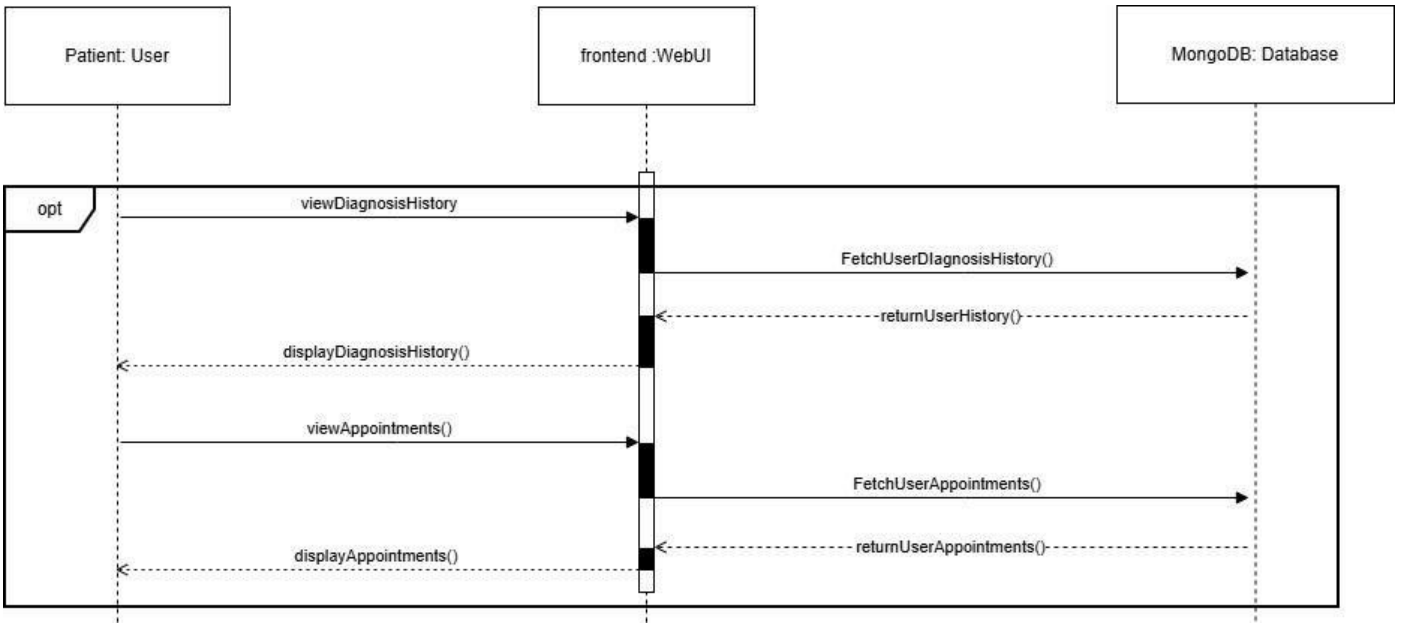
Alopecia Detection



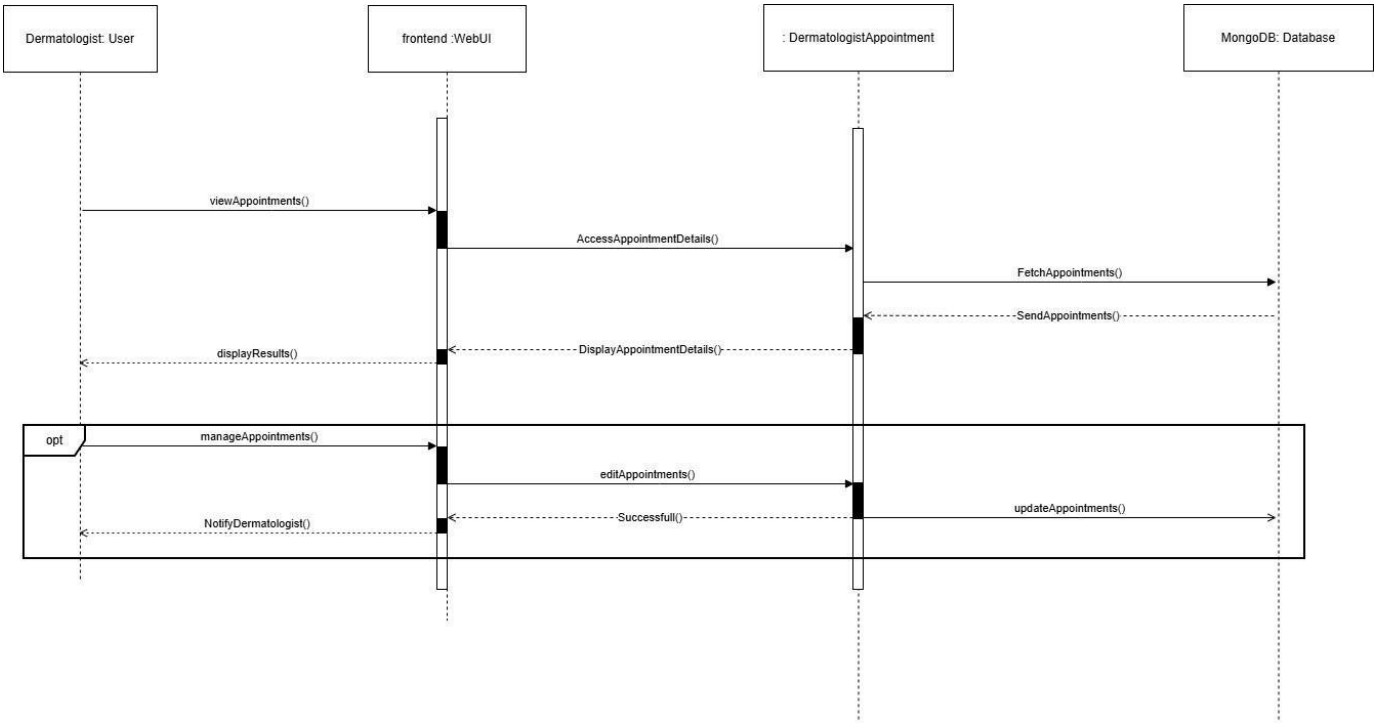
Dermatologist Appointment Booking



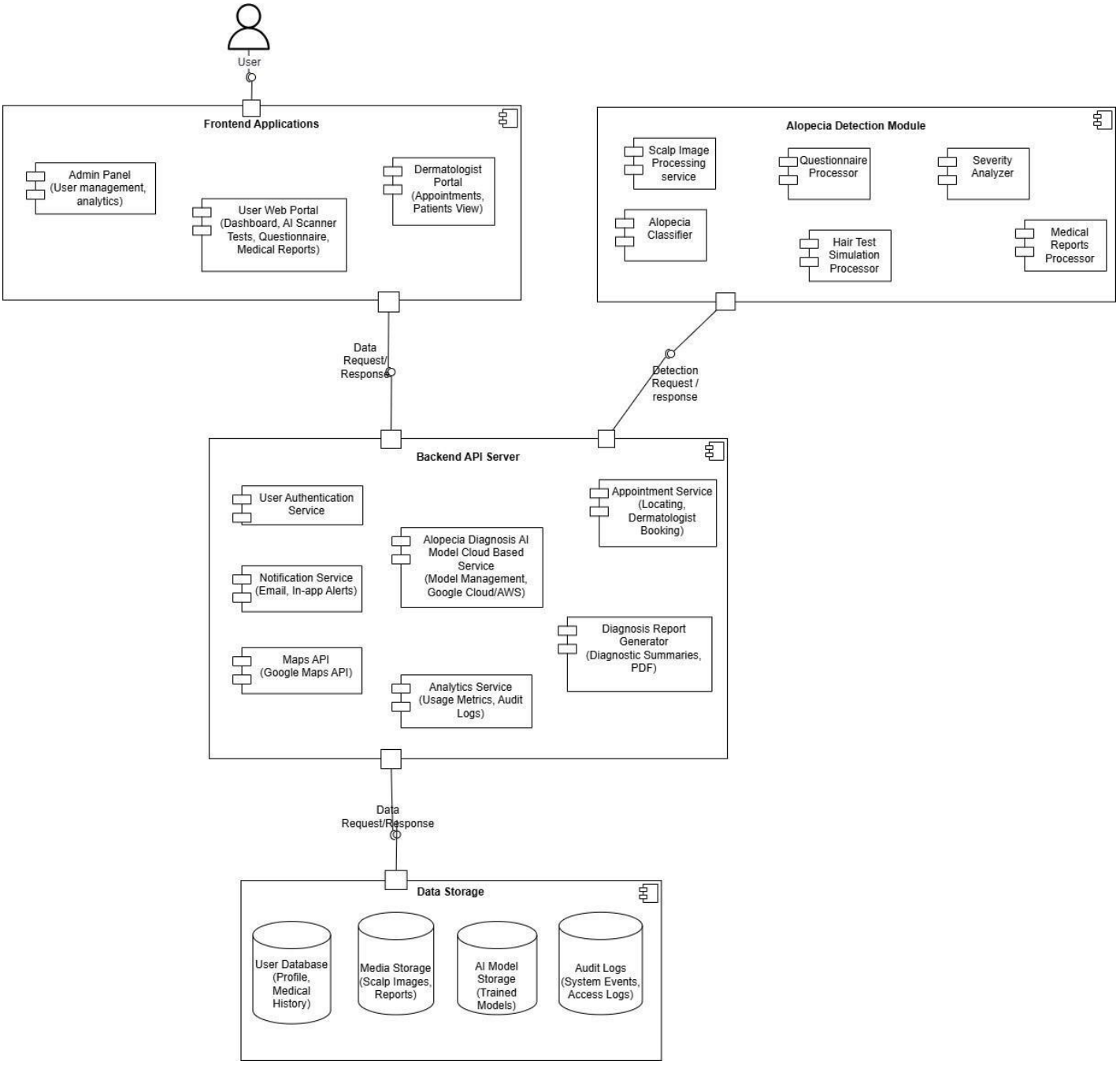
User History



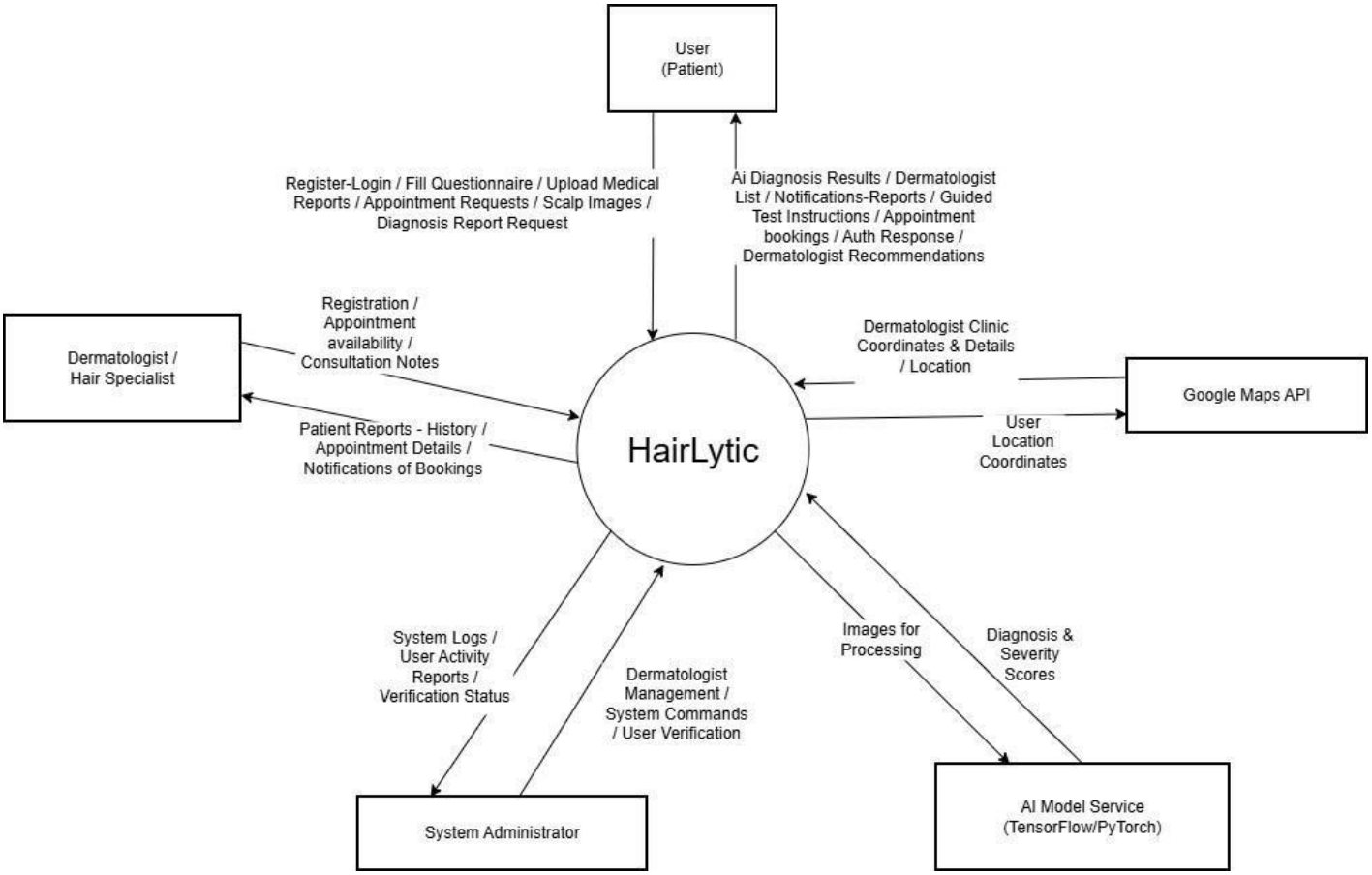
Dermatologist Appointment Management



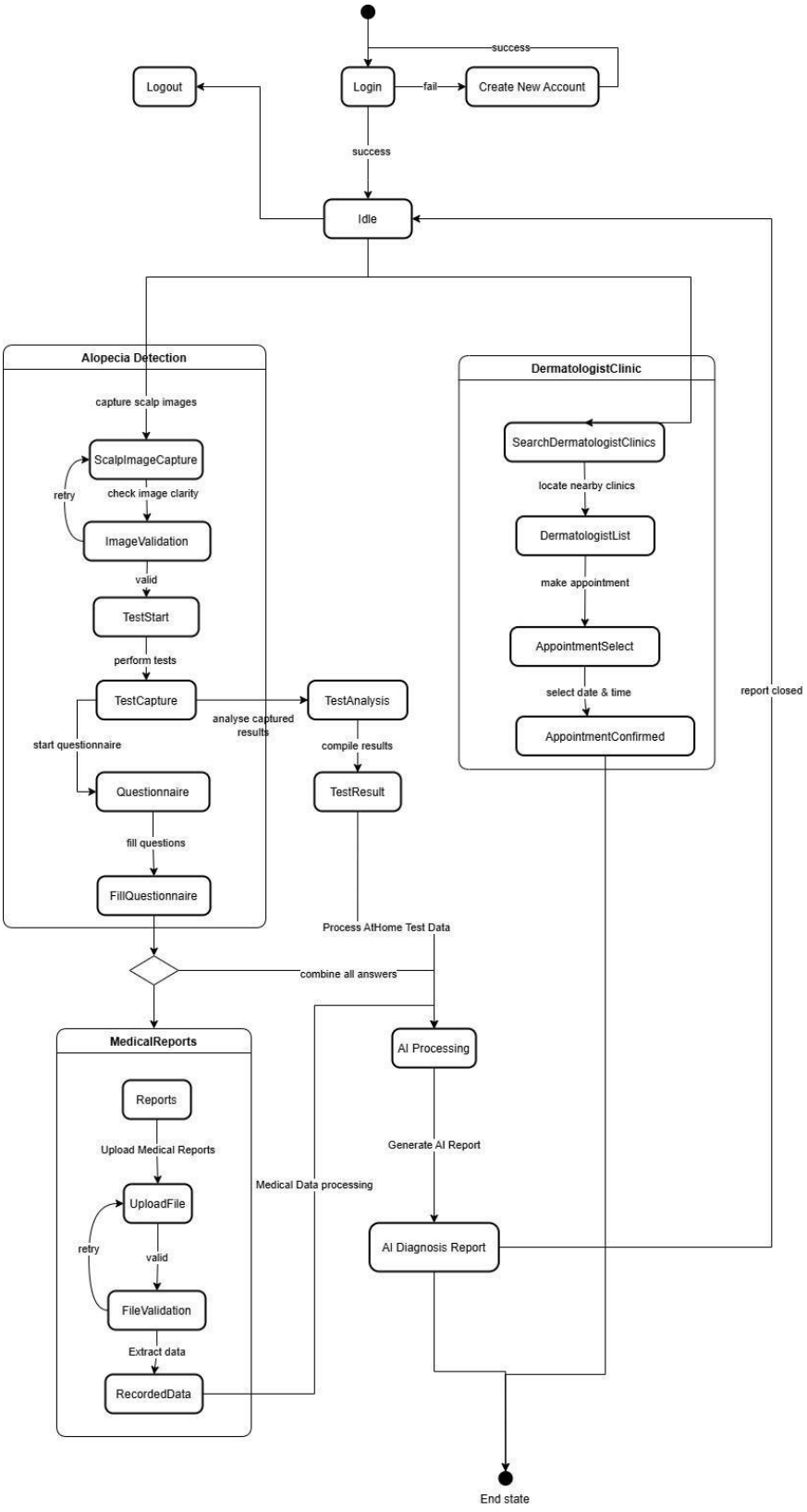
4.6. Component Diagram



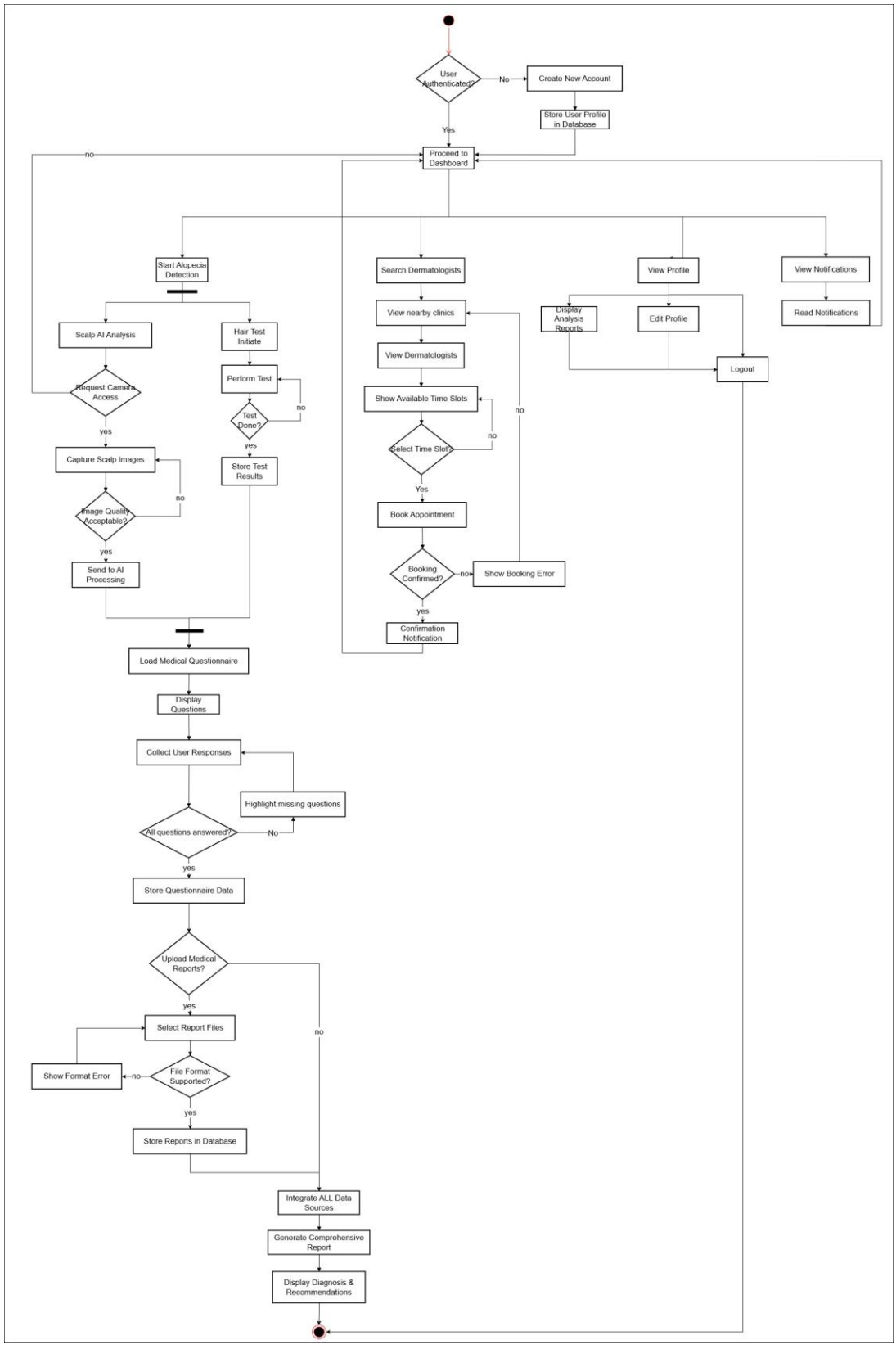
4.7. Data Flow Diagram



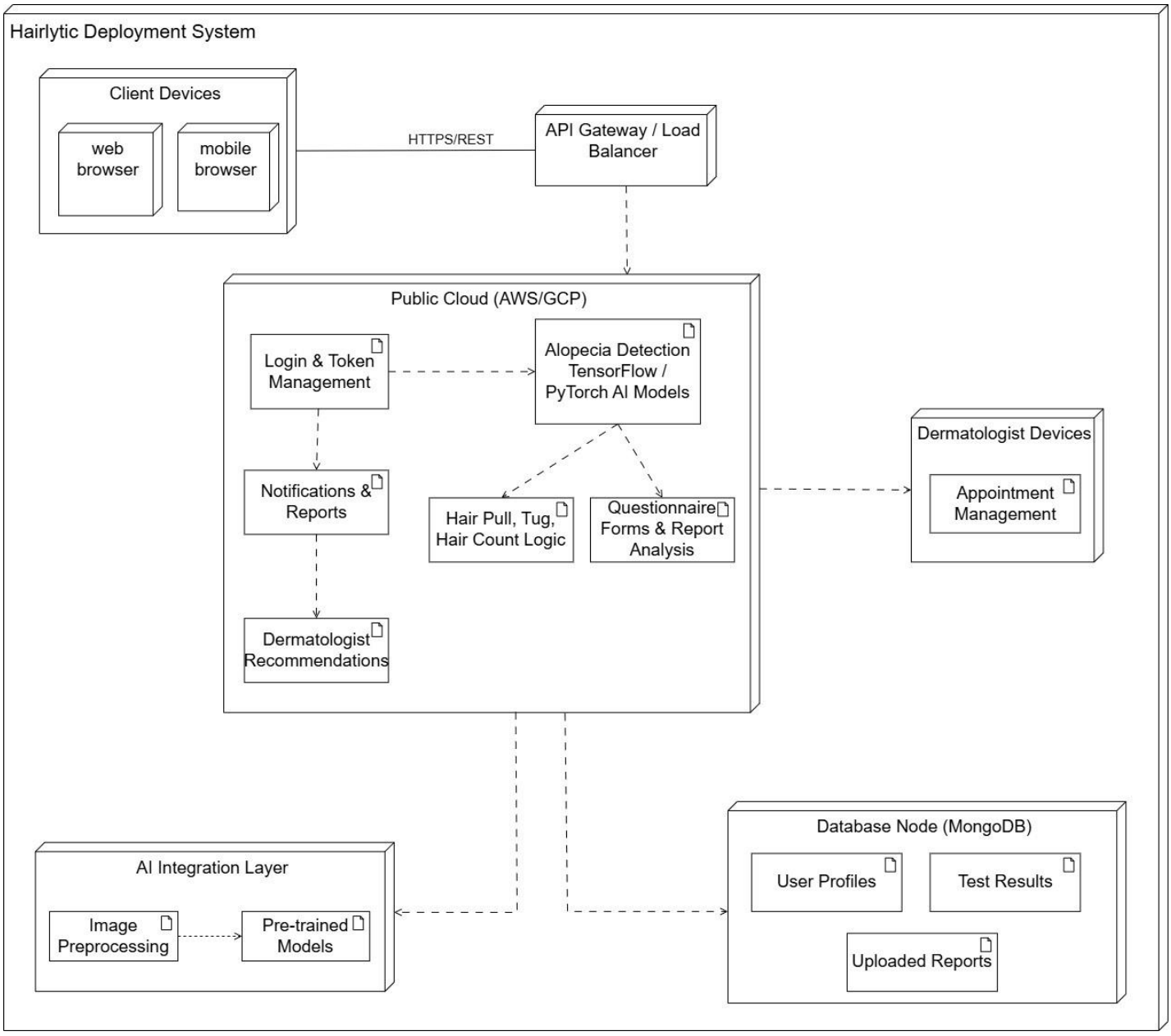
4.8. State Diagram



4.9. Activity Diagram



4.9. Deployment Diagram



Chapter 5
Implementation

Chapter 5: Implementation

This chapter explains how the Hairlytic system is built and how its main features work in practice. It explains how the system's modules, including AI-based alopecia detection, questionnaire handling, medical report uploads, and at-home hair test simulations, are implemented. The chapter also describes the technologies, tools, frameworks, and development methodologies used during construction of the platform. Overall, this chapter gives a clear view of how the system operates during development and how each part is implemented step by step.

5.1. Important Flow Control/Pseudo codes

5.1.1. User Login Process

```
START
USER enters email and password
SYSTEM checks if email exists
IF email does not exist
      SHOW "Account not found"
ELSE
      SYSTEM checks password
      IF password is correct
            LOGIN user
            OPEN dashboard
      ELSE
            SHOW "Wrong password"
END
```

5.1.2. User Registration Process

```
START
USER enters name, email, and password
SYSTEM checks if email already exists
IF email exists
      SHOW "Email already registered"
ELSE
      SAVE user details
      SEND verification email
      SHOW "Verify your email to continue"
END
```

5.1.3. Alopecia Detection

```
START
Patient opens Alopecia Diagnosis Module

SYSTEM asks for camera permission
IF camera allowed
      Patient show movements
      SYSTEM captures image and check quality
      IF image is clear
            SEND image to AI model
            RECEIVE alopecia type severity
            STORE AI results
      ELSE
            ASK patient to retake image
ELSE
      SHOW "Camera access needed"
      STOP camera
SYSTEM shows lifestyle questionnaire
Patient fills answers
SYSTEM checks if all required fields are filled
IF questionnaire complete
      SAVE questionnaire data
ELSE
      ASK patient to complete missing fields

SYSTEM explains how to perform Hair Pull Test
SYSTEM opens camera
```

```
IF camera opens successfully
      PATIENT performs hair pull test on camera
      SYSTEM analyzes captured frames
      COUNT number of shed hairs
      CLASSIFY shedding level (Normal / Mild / Severe)
      SAVE test results
ELSE
      SHOW "Camera access needed for Hair Pull Test"
SYSTEM asks if patient wants to upload medical reports
IF patient uploads file
      SYSTEM checks file format and size
      IF valid file
             SAVE report
      ELSE
             SHOW "Invalid file. Upload PDF/JPG/PNG"
ENDIF
SYSTEM gathers AI results + questionnaire + tests
COMBINE all data into report format
GENERATE final diagnostic summary
SHOW report to patient
ALLOW patient to download report

END
```

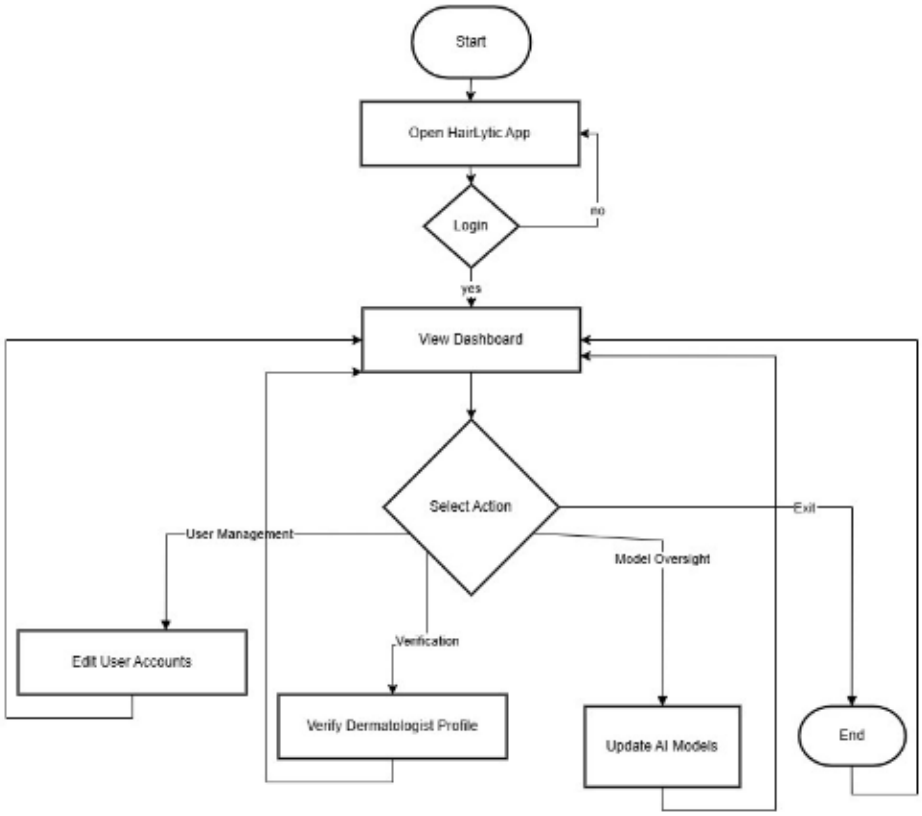
5.1.4. Alopecia Analysis & Dermatologist Flow

```
START
Patient opens Dermatologist Profile
SYSTEM asks for location permission
IF location allowed
      SYSTEM fetches nearby dermatologists from database / API
      DISPLAY list of dermatologists with distance and details
      Patient selects a dermatologist
      SYSTEM loads available time slots
      Patient chooses a time slot
      SYSTEM checks if slot is available
      IF slot is available
            Dermatologist confirms appointment
            SEND notification to patient and dermatologist
            SHOW "Appointment booked successfully"
      ELSE
            SHOW "Selected slot is not available"
ELSE
      SHOW "Location permission is required to search dermatologists"
END
```

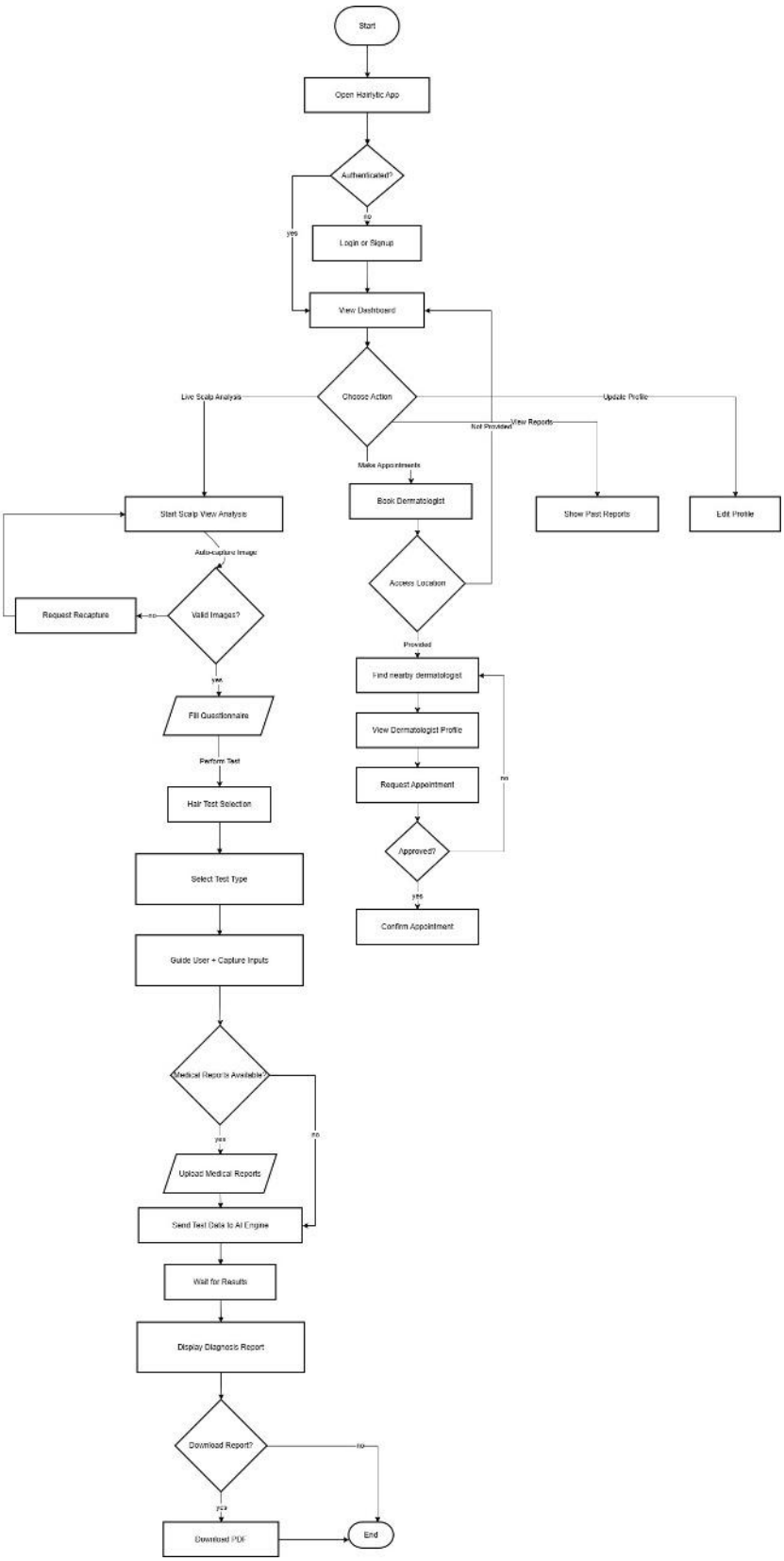
5.1.5. Notification Sending Flow

```
START
      SYSTEM detects new event (User Login, AI result, appointment
      update)
      CREATE notification message
      SEND notification to User
      IF delivery fails
            RETRY sending
      SHOW "Notification sent"
END
```

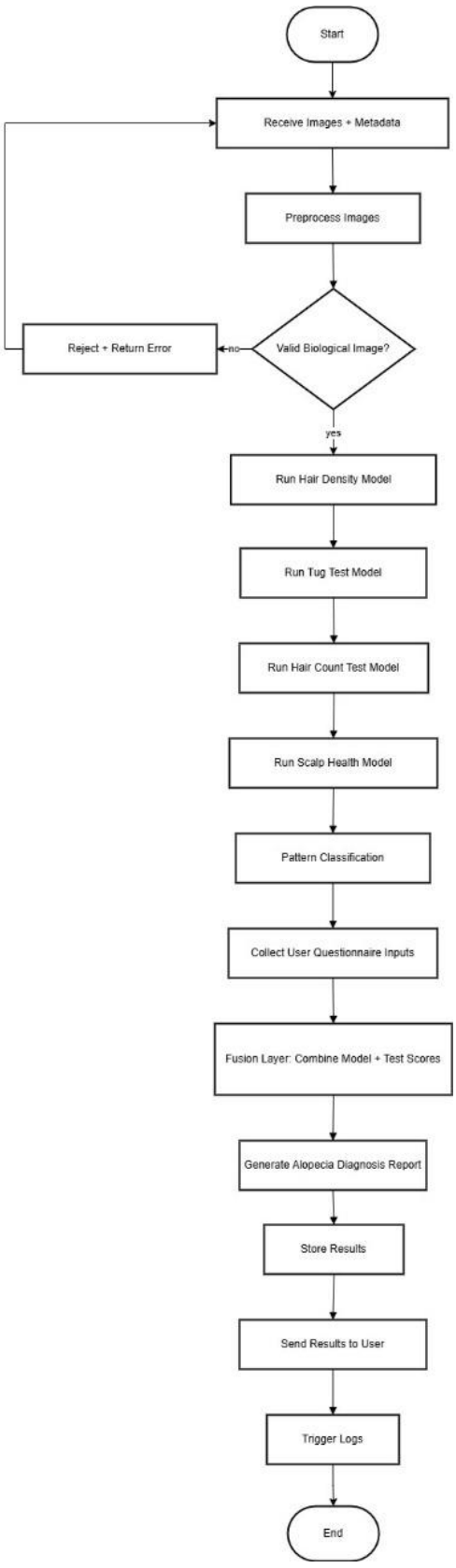
Role: Admin


Start
Open HairLytic App
Login
no
yes
View Dashboard
Select Action
Exit
User Management
Model Oversight
Verification
Edit User Accounts
Verify Dermatologist Profile
Update AI Models
End

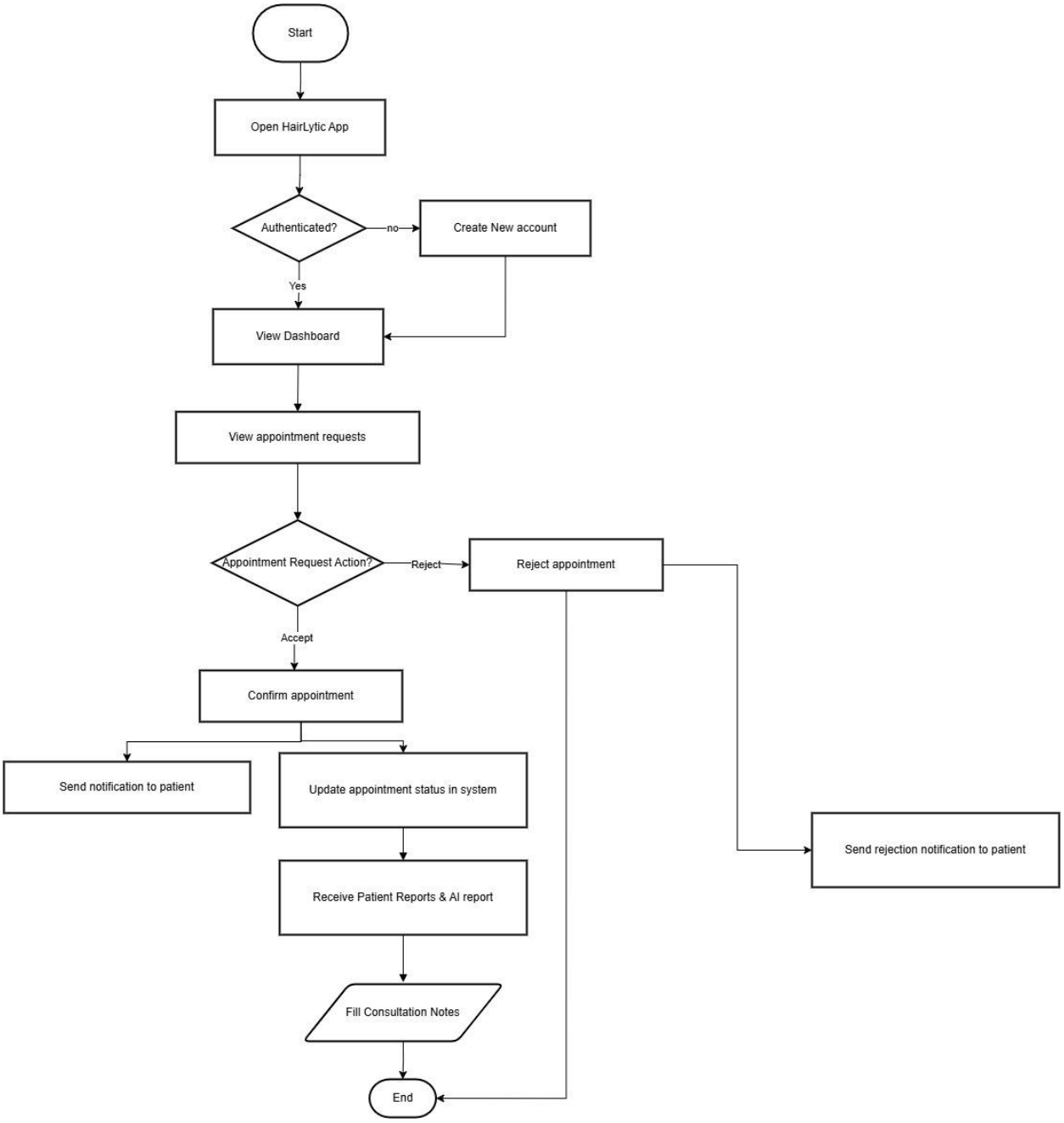
Role: Patient



Role: AI


Start
Receive Images + Metadata
Preprocess Images
Valid Biological Image?
no
Reject + Return Error
yes
Run Hair Density Model
Run Tug Test Model
Run Hair Count Test Model
Run Scalp Health Model
Pattern Classification
Collect User Questionnaire Inputs
Fusion Layer: Combine Model + Test Scores
Generate Alopecia Diagnosis Report
Store Results
Send Results to User
Trigger Logs
End

Role: Dermatologist



5.2. Components, Libraries, Web Services and stubs

5.2.1. System Components

- **Frontend Component (React.js Web App)**

  This component handles the user interface. It displays screens for login, AI detection, questionnaire, reports, tests, and clinic search.
- **Backend Component (Flask/FastAPI Server)**

  This component receives requests from the frontend, processes data, runs business logic, and sends back responses.
- **AI Processing Component**

  This module performs image analysis using trained machine-learning models. It detects alopecia type and severity.
- **Database Component (MongoDB)**

  Stores user profiles, AI results, medical reports, questionnaires, test data, clinic information, and appointments.
- **Notification Component**

  Sends email or in-app notifications for AI results and appointment confirmations.

5.2.2. Software Libraries

- **Frontend Libraries**
  - React.js — builds the user interface
  - Axios — sends API requests
  - Tailwind CSS — handles styling
  - React Router — manages navigation between pages
- **Backend Libraries**
  - Flask / FastAPI — creates API routes
  - PyMongo — connects backend to MongoDB
  - Auth / Auth 2.0 — handles secure login
  - OpenCV — checks image quality
  - TensorFlow / PyTorch — loads and runs the AI model
- **AI Libraries**
  - NumPy — data handling
  - Scikit-learn — preprocessing support
  - TensorFlow/PyTorch — model training and inference

5.2.3. Web Services and External APIs

- **Google Maps API**

  Used for finding nearby dermatologists and showing clinics on map view.
- **Email/Notification Service (Firebase)**

  Sends verification emails, appointment confirmations, and diagnostic alerts.
- **Cloud Storage (Google Cloud Storage)**

  Stores scalp images, medical reports, and other uploaded files.

5.2.4. Stubs (For Testing Unfinished Modules)

- **AI Model Stub**

  Returns a fake diagnosis like “Alopecia Areata, Mild” without running a real model.

  Used until the AI engine is fully integrated.
- **Appointment Booking Stub**

  Pretends to book an appointment and returns “Confirmed” even if backend logic is not ready.
- **Notification Stub**

  Prints “Notification Sent” instead of sending a real email.
- **Clinic Search Stub**

  Returns a list of sample dermatologists when the Google Maps API is not connected.

5.3. Deployment Environment

This project runs as a web-based application that works on common browsers like Chrome. The frontend is hosted on a cloud server where users can access the system from laptops and mobile devices with a stable internet connection. The backend runs on a cloud platform such as Google Cloud, which provides API hosting, database services, and GPU support for AI processing. The system uses MongoDB as the main database, stored securely on the server. The environment is designed to be scalable so it can support more users and higher AI processing loads in the future.

5.4. Tools and Techniques

This project uses a combination of development tools, programming techniques, and AI technologies to build the HairLytic web system. The frontend is developed using React.js and Tailwind CSS, which help create a clean, responsive, and user-friendly interface. The backend is built using Python with Flask/FastAPI, as these frameworks support fast API development and smooth communication with the AI models. For the AI part, TensorFlow and PyTorch are used to train and run alopecia detection models, while OpenCV helps with image processing tasks

such as checking image clarity. MongoDB is used as the main database because it stores images, reports, and user data efficiently. The system also uses Google Maps API to show nearby clinics. Throughout development, Git and GitHub are used for version control, making teamwork easier. These tools and techniques together support the smooth development, testing, and deployment of the complete AI-based dermatology platform.

5.5. Best Practices / Coding Standards

The project follows clean coding standards to keep the system easy to read, maintain, and extend. Code is written using meaningful variable names, proper indentation, and consistent formatting. Each module is kept separate to avoid complexity and to support reusability. Comments are added where necessary to explain logic, and error handling is implemented to manage invalid inputs or failures. The team uses Git-based version control to track changes and prevent conflicts during development.

5.6. Version Control

This project uses version control to keep track of all changes made during development. Git is used as the main version control system, and GitHub stores the online repository. Version control helps the team work together, upload updated files, and prevent loss of work. Each new feature or fix is committed with a clear message so changes are easy to understand. Branches are created to separate different modules, and they are merged only after testing. This process keeps the code organized, avoids conflicts, and ensures that the project history is stored safely for future improvements.