USF MUMA COLLEGE OF BUSINESS UNIVERSITY OF SOUTH FLORIDA

Final Project Report

Skinfinity – An AI-Powered Skin Care Consultant

Advanced System Analysis & Design

Abdallah Musmar, Ph.D.

Team Member

Ankita Tripathy

Aayushi Desai

Priyanka Jammu

TABLE OF CONTENT

1. Project Proposal and Requirements

Introduction

The skincare industry is filled with millions of products, routines, and conflicting advice, confusing customers and leaving them uncertain about what really works for their own skin. Most people waste money and time on ineffective or even damaging products. Getting the right skincare solution is like playing a guessing game.

Skinfinity does this by using a smart, AI-driven skincare guide. The ability to scan high-resolution facial photos and collect user-specific data, including age, lifestyle, and skin issues, enables Skinfinity to provide accurate, easy-to-understand customized skincare recommendations. The goal is to remove the confusion and trial-and-error associated with skincare and offer a focused, data-driven approach. Customers can receive personalized product and regimen suggestions from a simple, mobile-friendly interface, without the expense of a professional visit. With Skinfinity, individuals gain transparency, confidence, and better results on their skincare journey, leading to healthier skin and a better-informed practice of self-care.

Significance of the Project

- Minimize trial-and-error in skincare through serving accurate, AI-aided recommendations
- Support personalized skincare planning with minimal effort required.
- Prevent dermatology consultation overload through simple advice before a professional consultation.

Objectives of Quick Check

- Deliver AI-driven facial analysis with accurate detection of skin issues such as acne, dryness, and pigmentation.
- Support flexible, scalable deployment using a microservices architecture.
- Deliver secure, privacy-focused experiences using encrypted data storage and role-based access control.
- Offer an intuitive, mobile-optimized UI to ensure universal accessibility.

Problem Statement

Choosing the right skincare products and routines is generally a challenge for man people. This is because they may not be aware of what would suit their skin type, they lack professional guidance, or they spend time and money trying out different products without success. Therefore, most people resort to relying on random advice or trying out products based on advertisements, which can lead to negative results or even harm their skin. Without proper equipment to scan their skin, people may miss early warning signs of problems or treat them inappropriately. This can turn small issues into big ones, eventually requiring professional help.

Skinfinity is one solution to this problem. It uses artificial intelligence to analyze face photos and user data like skin problems, age, and lifestyle, to give personalized skincare advice. It is not a replacement for a dermatologist, but it gives users a helpful starting point to better understand and care for their skin. In an overcrowded marketplace with too many unclear messages, Skinfinity stands out with simple, personalized, and science-driven advice. It keeps customers clear of trial-and-error stress and helps them achieve better skin health with confidence.

Techniques Used for Requirements Gathering

To ensure that Skinfinity meets both user needs and business goals, we used several methods to gather crucial information before starting the project:

User-Centered Research

We talked to over 200 people of different ages and skin types through surveys and interviews. This helped us learn about their skincare routines, problems, and what features they would find helpful like getting personalized product suggestions and being able to filter by ingredients.

Expert and Clinical Input

We collaborated with dermatologists and skincare professionals to make sure our app gives safe and accurate advice. Their input helped us avoid offering risky suggestions and improved our ability to detect different skin issues.

Competitor and Market Analysis

We looked at other popular skincare apps to understand their strengths and weaknesses. This knowledge helped us to design Skinfinity to be more user-friendly, personalized, and powered by AI.

Prototype Testing and Feedback

We created early versions (both simple and detailed) of the app and tested them with users. Their feedback helped us improve the app’s design, features, and ease of use. We made changes based on what users liked or found confusing.

By combining user feedback, expert guidance, and market research, we built a system that is both useful and reliable for anyone looking for better skincare.

Stakeholder Analysis

Stakeholder	Role and Contribution
Users	Use the app to receive personalized skincare analysis and product recommendations.
Dermatologists	Provide expert feedback to ensure the AI recommendations are safe and medically sound.
Developers	Build and maintain the microservices, AI models, and cloud infrastructure.
Designers	Create intuitive, user-friendly interfaces based on user feedback and usability testing.
Business Analysts	Ensure features meet both user expectations and business goals by analyzing trends and needs.

Functional and Non-Functional Requirements

Functional Requirements

From the problem domain and stakeholder requirements, the team defined a comprehensive set of functional requirements for Skinfinity. The system functionality was broken down into separate pieces (eventually deployed as microservices) that address each major user story or feature:

User Registration & Profile Management:

Users will be able to register, log in, and have a personal profile. This entails storing personal data, preferences, and skincare history. (Managed by the User Profile Service, which manages authentication, profile editing, and user settings.)

Skin Questionnaire Intake:

The system will collect detailed user inputs through dynamic questionnaires. Users can input information about their lifestyle, skin problems, allergies, and current routine. (Handled by the Questionnaire Service, which shows forms to users to input these details.)

Image Upload and Analysis:

The platform will upload user-uploaded images and scan them for skin conditions (acne, dryness, pigmentation, etc.). The images will be processed through sophisticated AI/ML algorithms that identify features and issues. (Processed by the Image Analysis Service, where AI models are applied to the images to obtain skin condition measurements.)

Personalized Recommendation Generation:

The system generates personalized skincare advice (products and regimen) according to the user profile, answers in the questionnaire, and results from image analysis. This consists of suggesting single products (whose ingredients match the needs and intolerances of the user) and step-by-step routines. (Done by the Recommendation Engine Service that takes inferences from analysis services and determines personal suggestions.)

Product Catalog Browsing:

The system must maintain a product catalog of skincare products with data like ingredients, skin type suitability, and inventory status. It must allow the recommendation engine to request product information and possibly allow browsing of suggested product information. (Handled by the Product Catalog Service, which stores product metadata and provides lookup operations.)

Online Consultation Booking:

Users should be able to book virtual consultations with dermatologists or skincare experts through the website. Users can choose time slots, and the system manages appointment details and history. (Managed by the Online Consultation Service, supporting booking, managing session data, and storing dermatologist recommendations from sessions.)

Payment Processing:

If some of the features (e.g., consultations or premium services) are fee-based, the system will handle secure payment transactions. Payment processing, transaction history storage, and handling refunds or billing problems are all included. (Handled by the Payment Service, which integrates with payment gateways and stores billing records.)

Progress Tracking:

The system will allow users to track their skin's progress over time. Users can upload images periodically or update the survey, and the system shows changes (e.g., before/after photos, trends in skin condition measurements). (Handled by the Progress Tracking Service, which maintains time-series data and image history to show progress dashboards.)

Notifications & Feedback:

Users will be alerted by the system with timely reminders (for product application, routine compliance, or consultations) and gather user feedback on product/recommendation recommendations. (Handled by the Notification & Feedback Service, sending reminder apps/emails and saving feedback responses.)

All these functional needs were elicited and mapped to the elements of the system architecture so that traceability from requirements to code is maintained. Together, they offer a full complement of features to meet the end-to-end user experience: from onboarding, through analysis and recommendation, to ongoing engagement and expert support.

Non-Functional Requirements

- **Usability and Accessibility**

Skinfinity is simple to use, even for those who aren't very tech minded. We designed it with minimalism, intuitive navigation, and clear text and colors. Everyone can use the app, including those with visual impairments or little skincare knowledge. Our top priority was creating a fun, easy-to-use experience for everyone.

- **Scalability**

The more people who use the app, the faster it should be. That's why we used an elastic system architecture (microservices), where each part of the app can grow independently. For example, if more users are using the image analysis feature, it can handle extra load without making the rest of the app slow. We also use cloud services like AWS, which allow us to store a lot of data and maintain speed.

- **Performance**

Skinfinity is built to respond quickly. Scanning a face or suggesting products should take no longer than a few seconds. We use the right kind of databases for quick access and implement caching to speed up the process. Real-time actions (like logging in) are executed immediately, while background tasks (like sending messages) are done in the background so that the user experience is not interrupted.

- **Security**

As we handle personal data (e.g., face images and health records), strong security is a priority. Everything received or sent gets encrypted. Secure tokens are used for user authentication, and only the right people (e.g., dermatologists) are allowed to view special functions. Passwords, payment information, and other data of sensitive nature are also secured by keeping them safely stored. Our system is resistant to standard cyber attacks and is tested on a regular basis to keep it secure.

- **Reliability and Availability**

Skinfinity is designed to be reliable with minimal downtime. If one part of the app fails, the others will still work. We use health checks, backups, and auto-restart to quickly recover from failure. If something fails (such as sending a message), the system will attempt it again without data loss. This allows the user to have faith that the app will always be available when they need it.

- **Extensibility and Maintainability**

When the app grows, it must be simple to modify or add features. As we developed a modular configuration, developers can modify or fix one component without affecting the whole app. We wrote according to the best practices and documented everything we did, and future updates will be seamless. Let's say we need to include a skin care forum or a new AI functionality. We can include it without doing much of a change.

Hence, Skinfinity isn't just feature rich it's built to be fast, secure, simple to use, and scalable. These traits underpin our ambition to deliver a robust, contemporary, and secure skincare solution.

2. User-Centric Design Document

UX/UI Design Principles

Skinfinity is simple, non-cluttered, and easy to navigate. Users can upload face pictures and answer skincare questions easily. The interface boasts big buttons, readable text, and step-by-step instructions to make it accessible to users of every age and ability.

•AI-Powered Skin Analysis

The app uses AI to detect skin problems such as acne, redness, dullness, dryness, and pigmentation from uploaded images. Results are displayed with labels labeled clearly and can also be saved as a PDF to use on one's own or share with a dermatologist.

• Personalized Recommendations

According to user feedback and skin assessment, the app suggests products that match the user's issues and skin type. Suggestions are selected from a verified database approved by dermatologists.

• Dermatologist Consultations

Dermatologist consultations can be scheduled by users virtually within the application itself. There is a transparent view of available slots and past consultations stored for retrieval.

• Secure Payments

Payment for consultations and premium features is made through a secure interface. Users are guaranteed confirmations and a visible payment history in the app.

• Cross-Platform Compatibility

Skinfinity is supported by desktop computers, tablets, and phones. It is a consistent experience across all platforms.

• User Convenience Features

The app supports dark mode, theming, and multi-language capabilities. Progress bars guide users through each process, and offline capabilities allow data entry without an internet connection.

• Security and Privacy

Only necessary information is collected, and all information is encrypted and safely stored.

HCI Best Practices

To make Skinfinity enjoyable to use, we followed key Human-Computer Interaction (HCI) best practices. HCI guidelines help promote usability, accessibility, and general user satisfaction, and reliable skincare guidance.

• **Intuitive Navigation**

The application follows a step-by-step method loading pictures, analysis of the skin, suggestions, and dermatologist visits. Simple buttons, simple design, and progress bars smoothly guide users through the application without confusion.

• **Visual Signals for Skin Issues**

Skinfinity applies color highlights and highlighted markers to signal various skin issues such as redness, acne, or dryness. This makes it easy for users to understand skin health in an instant and identify areas that need attention.

• Input Validation and Error Handling

To avoid erroneous inputs, the app has real-time validation and smart suggestions. When users omit a step or enter something inaccurately, motivational messages encourage them to rectify it immediately.

• Accessibility and Inclusive Features

Skinfinity provides voice input, font size choice, and high contrast mode in an effort to meet users with diverse needs. It is also screen-reader compatible and provides gesture navigation for improved accessibility.

• Interactive and Personalized Experience

AI-derived outputs are tailored to every user's unique skin history and skin type. Interactive feedback, such as animations and friendly notifications, keeps the user involved. A personal dashboard allows users to track skin over time and stick to their routine.

- **Prototyping and User Testing**

A methodical, iterative design process was utilized to refine the Skinfinity interface to capture ease, responsiveness, and accessibility by continually making changes.

• **Wireframing & Prototyping**

1. Low-Fidelity Wireframes – Created to outline the general layout, user flow, and feature hierarchy.

2. High-Fidelity Prototypes – Included high-detailed images, brand features, and smoothed UI to provide an actual app-like look.

3. Interactivity Testing – Included interactive buttons, hover effects, and smooth transitions to simulate actual usage.

• Full Usability Testing

1. User engagement – Over 50 users tested the usability of features like image upload, questionnaire flow, and product suggestions.

2. Scenario-based testing – Users performed significant actions such as receiving AI skin analysis and booking consultations.

3. Accessibility Testing – Verified support for screen readers, voice input, high-contrast themes, and resizable fonts.

• **Continuous Iterative Refinements**

1. Feedback Loops – Collected feedback to identify confusing elements and areas for additional guidance.

2. Incremental Enhancements – Improved screen layout, button locations, and input validation.

3. Performance Optimization – Reduced loading time, optimized image processing, and provided smooth transitions.

Through the synergy of intensive prototyping, user-centered testing, and continuous design iteration, Skinfinity offers customers a smooth, accessible, and interactive experience in seeking tailored skincare guidance.

3. Cloud-Based System Architecture

Skinfinity is based on a microservices architecture for the cloud via AWS. There is one independent service with an independent database and API Gateway for communication from each service. This setup provides features like skin analysis, product recommendations, and dermatologist consults securely, scalably, and with top performance.

• Cloud Service Utilization

The Skinfinity platform integrates with AWS and cloud-native services to help with scalable, secure, and effective deployment in every microservice.

• **Compute & Containerization**

The application is executed in Docker containers and managed by AWS ECS and Kubernetes. This offers auto scaling and high availability for each microservice.

• **Database Management**

- RDS (PostgreSQL) manages structured data such as user profiles and recommendation logs.

- Flexible questionnaire inputs and skin records are stored in MongoDB Atlas.

- DynamoDB provides fast access to product catalogs.

- Encrypted SQL databases store product details securely.

- Time-series progress tracking data is managed by Influx DB.

- Relational databases are used for storing appointments and feedback.

• **Cloud Storage**

AWS S3 stores user-uploaded images, AI reports, and logs. Storage is HIPAA and privacy policy compliant.

• API Gateway & Load Balancing

AWS API Gateway routes user requests effectively between microservices. NGINX is used for load balancing for effective traffic distribution with minimal latency.

• **Messaging & Event Processing**

RabbitMQ or Apache Kafka enables asynchronous service-to-service communication, specifically image processing and feedback events.

• Notifications & Alerts

AWS SNS provides real-time notifications, including appointment reminders, skin updates, and skin care advice.

• Security & Identity Management

AWS IAM employs role-based access control (RBAC) for safe access to services. AWS WAF protects the system from attacks such as SQL injection and DDoS.

System monitoring, log monitoring, and performance tuning are achieved by utilizing AWS CloudWatch and the ELK Stack (Elasticsearch, Logstash, Kibana).

Cloud Architecture Patterns

The Skinfinity system follows a scalable, modular cloud architecture that supports flexibility, performance, and fault isolation.

- **Microservices-Based Architecture** – Core services like User Profile, Image Analysis, Recommendation Engine, and Appointment Booking run independently. This improves fault isolation and allows each service to scale as needed.
- **Event-Driven Architecture** – Asynchronous communication is enabled using RabbitMQ or Kafka. This allows services to exchange data in real time without creating system delays or bottlenecks.
- **Hybrid Cloud Approach** – Critical components such as image storage and notifications are hosted on AWS, while MongoDB Atlas handles flexible, unstructured data like questionnaire responses.
- **RESTful & GraphQL APIs** – Services interact through lightweight RESTful and GraphQL APIs, making data exchange efficient and improving overall system responsiveness.
- **Serverless Computing (AWS Lambda)** – Background tasks like AI image processing and skincare reminders are handled using AWS Lambda, reducing infrastructure costs and improving system performance.

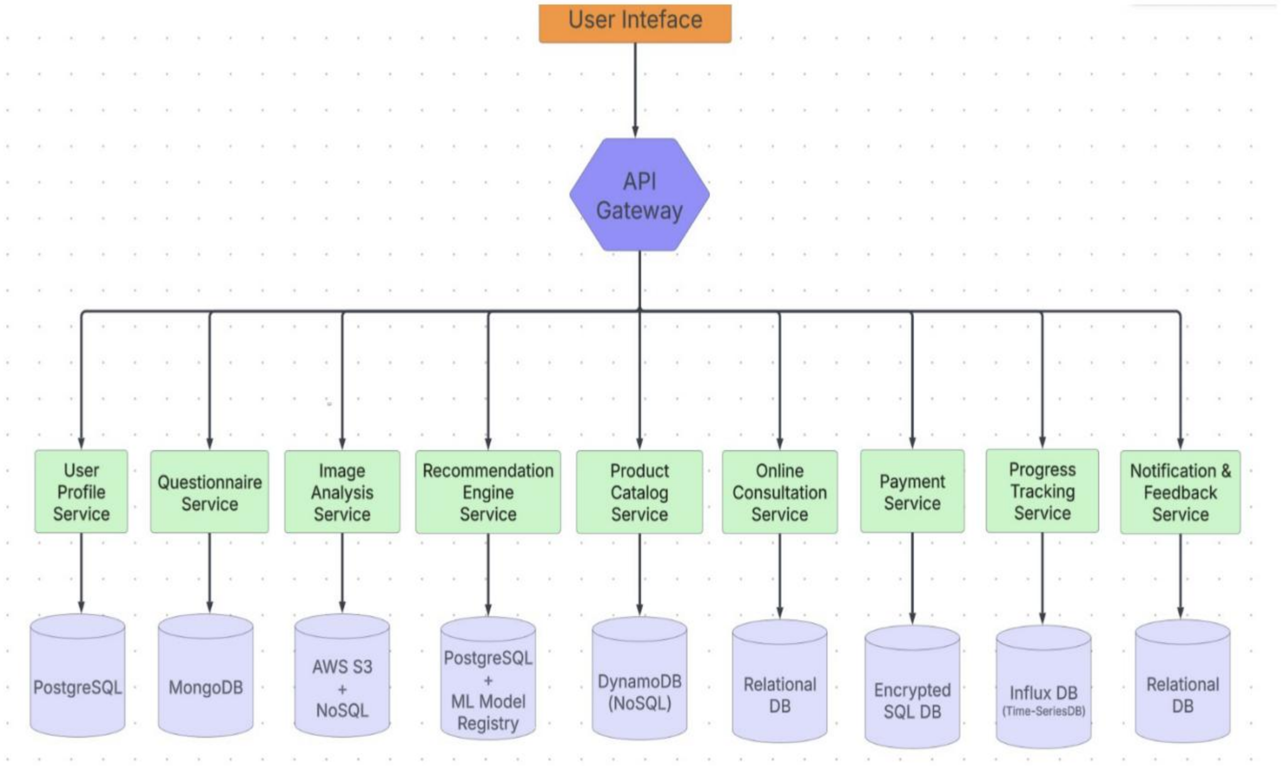

Fig 3.1 : System Architecture Diagram

4. Agile Project Delivery and Management Strategy

The Skinfinity project was developed using the Agile Scrum methodology to ensure smooth project implementation, quick adaptability, and constant evolution. This methodology allowed the team to produce an extensible, stable, and easy-to-use skincare system.

Methodology Application

The team followed a Scrum-based Agile model, shipping in 2-week sprints to provide steady momentum, build quick prototypes, and incorporate feedback. Within every sprint, the main features of AI analysis, product recommendations, and dermatologist consultations were polished.

- **Scrum Framework –** The Scrum Master coordinated sprint planning, daily standups, sprint reviews, and retrospectives. Regular meetings kept the team aligned, resolved blockers rapidly, and ensured new features were being delivered and tested in cycles.

- **Kanban Elements –** A JIRA Kanban board represented tasks, monitored progress, and prioritized backlog items. This enabled work to be monitored openly, reduced delays, and optimized task flow within the sprint cycle.

- **Effectiveness & Adaptability –** Agile's flexibility allowed the team to rapidly react to feedback and changing user demands. The ongoing feedback cycle assisted in improving the usability, functionality, and design of the app. This helped to avoid risks and made every release finer and more useful for users.

With Scrum's structured process combined with Kanban's transparent flow, Skinfinity had a focused but agile development process that led to a refined and optimized final product.

Iterative Development

Each sprint in the Skinfinity project focused on building, testing, and refining specific system features using a rapid prototyping and continuous improvement approach.

Sprint	**Deliverables**	**Key Focus Areas**
Sprint 1	User Registration & Profile Setup	Secure login, profile management, data storage
Sprint 2	AI-Based Skin Analysis Engine	Image processing, ML model integration, accuracy

Sprint 3	Product Catalog & Appointment Booking	API integration, real-time availability checks
Sprint 4	Security and Performance Optimization	Load testing, data encryption, response tuning

- **Prototyping & Testing** – Each feature was tested continuously through unit tests, integration tests, and user acceptance testing (UAT) to ensure functionality and reliability.
- **Refinement Process** – Feedback collected from early users and testers was carefully reviewed and applied in following sprints. This ensured that the system remained user-friendly, responsive, and aligned with real skincare needs.

Team Collaboration

Effective teamwork and clear communication were key to the success of the Skinfinity project. The team used collaboration tools to stay organized, aligned, and responsive throughout development.

• **JIRA** – Used to plan sprints, manage the product backlog, assign tasks, and track progress. This helped ensure transparency and focus across the team.

• **Zoom** – Used for daily standups, sprint planning sessions, and quick resolution of blockers or development issues.

• **GitHub** – Provided version control and allowed for seamless collaboration through code reviews, issue tracking, and pull requests.

By combining Agile practices with strong communication and collaboration tools, the Skinfinity team was able to work efficiently, adapt quickly to feedback, and deliver a high-quality system through continuous refinement.

5. Technology Stack, Security, and Scalability Strategy

The Skinfinity system is built using a modern, cloud-native technology stack with strong security and scalability practices to support performance, data protection, and future growth.

Technology Selection

The chosen technology stack aligns with system requirements, performance needs, and scalability objectives, ensuring an efficient, secure, and responsive user experience.

Component	**Technology Used**	**Justification**
Backend	Java (Spring Boot)	Supports scalable microservices and RESTful API development.
AI Model	Python (Scikit-Learn, TensorFlow)	Powers the skin analysis engine and personalized product recommendations.
Frontend	Angular (TypeScript)	Delivers a responsive, dynamic UI optimized for mobile and desktop.
Database	PostgreSQL (AWS RDS)	Stores structured data like user profiles and appointment details.
NoSQL Storage	MongoDB Atlas	Manages unstructured questionnaire and AI output data.
Messaging & Events	RabbitMQ/Kafka	Enables asynchronous communication between services for smoother performance.
API Gateway & Load Balancing	NGINX & AWS API Gateway	Routes and balances API requests across services efficiently.
Cloud Storage	AWS S3	Securely stores image uploads, reports, and logs.

Security Framework

To safeguard user data, prevent unauthorized access, and reduce cyber threats, the system employs several layers of security:

- **Data Encryption** – AES-256 encryption is used to protect all data at rest and in transit.
- **Authentication & Authorization** – OAuth 2.0 and JWT tokens ensure secure user access and session

control.

- **Access Control** – AWS IAM enforces role-based access to limit user privileges.
- **Web Security** – AWS WAF protects against threats like SQL injection, DDoS, and XSS.
- **API Security** – Includes rate limiting, input validation, and HTTPS enforcement.
- **Compliance & Privacy** – The system adheres to HIPAA and GDPR standards for data protection.

→ Scalability Measures

To handle high traffic loads, optimize database performance, and ensure seamless scaling, the system integrates multiple scalability strategies:

- **Load Balancing** – AWS ELB and NGINX balance traffic and prevent server overload.
- **Database Optimization** – Indexing and sharding are used to enhance database performance.
- **Containerization & Orchestration** – Docker and Kubernetes allow horizontal scaling of services.
- **Caching** – Redis or AWS ElastiCache is used to speed up frequently accessed data.
- **Auto-Scaling** – AWS Auto Scaling automatically adjusts system resources based on demand.

6. Conclusion

Skinfinity was successfully implemented through the combination of advanced AI technologies, a user-friendly interface, and a highly scalable cloud-based microservices architecture. Machine learning is utilized in the system for detailed skin analysis, while AWS services and scalable infrastructure enable real-time performance, data security, and seamless integration of all the features.

By following the Scrum Agile methodology, the team was able to develop, test, and refine the application in iterative cycles. Continual feedback from users allowed for design, functionality, and overall user experience. JIRA and GitHub facilitated easy project management, sprint tracking, and collaboration throughout the development process.

System goals and performance requirements drove technology choices. Angular and Spring Boot facilitated scalable and modular application layers, and Python-based AI models enhanced skin condition detection accuracy. PostgreSQL on AWS RDS and MongoDB Atlas were chosen for structured and unstructured data handling, and RabbitMQ/Kafka provided event-driven communication between microservices.

Security remained a top priority. The system ensures data security through AES-256 encryption, OAuth 2.0 authorization, HTTPS enforcement, and AWS IAM for role-based access control.

With its intelligent capabilities, strong security framework, and flexible design, Skinfinity offers a sound, personalized skincare assistant that adapts to user needs and will scale with future innovation.

Our application links:

Low Fidelity:

https://www.figma.com/design/jttruvRwnI6KY20s2IuT0U/Skinfinity--User-Centric-Design?node-id=0-1&p=f&t=jTspkFt1l8Bdq50e-0

High Fidelity:

https://www.figma.com/design/jttruvRwnI6KY20s2IuT0U/Skinfinity--User-Centric-Design?node-id=1-2&p=f&t=jTspkFt1l8Bdq50e-0

High Fidelity Prototype Link:

https://www.figma.com/proto/jttruvRwnI6KY20s2IuT0U/Skinfinity--User-Centric-Design?node-id=122-2&p=f&t=HDdB0IGMZAVH3FnF-1&scaling=scale-down&content-scaling=fixed&page-id=1%3A2&starting-point-node-id=122%3A2