HealthBot+

Software Requirements Specification

For

AI-Powered Web App for Skin Disease Diagnosis and Management

Version 1.0

210329E - LAKSARA K.Y.
210333K - LAKSHAN P.D.
210343P - LIYANAGE I.V.S.

Revision History

Date	Version	Description	Author
17/7/2024	1.0	Initial Commit	210329E, 210333K, 210343P

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this SRS is to fully describe the external behavior of the HealthBot+ application. It also describes nonfunctional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the requirements for the software.

1.2 Scope

HealthBot+ is an AI-powered web application designed to enhance the early detection and management of skin diseases. The system integrates advanced image analysis and text-based inputs to accurately diagnose various skin conditions, including skin cancer. HealthBot+ utilizes pre-trained models and transfer learning techniques to provide reliable diagnostic support. Additionally, the platform includes an oncology-trained chatbot that offers detailed medical information and advice to users. The application ensures data privacy, secure management of patient history, and a user-friendly interface. HealthBot+ aims to benefit both users and healthcare professionals by offering accessible, accurate, and transparent diagnostic tools.

1.3 Definitions, Acronyms, and Abbreviations

AI: Artificial Intelligence

SRS: Software Requirements Specification

HealthBot+: The name of the AI-powered web application for skin disease detection and management.

Oncology: A branch of medicine that deals with the prevention, diagnosis, and treatment of cancer.

Chatbot: An AI-powered program that simulates human conversation.

Pre-trained Models: Machine learning models that have been previously trained on a large dataset.

Transfer Learning: A machine learning technique where a pre-trained model is adapted to a new, but related, task.

1.4 References

1. SkinVision. (2024, May 30). SkinVision | Skin Cancer Melanoma Detection App | SkinVision. SkinVision - https://www.skinvision.com

2. U. K. Lilhore, S. Simaiya, Y. K. Sharma, K. S. Kaswan, K. B. V. B. Rao, V. V. R. M. Rao, A. Baliyan, A. Bijalwan, and R. Alroobaea, "A precise model for skin cancer diagnosis using hybrid U-Net and improved MobileNet-V3 with hyperparameters optimization," Scientific Reports, vol. 14, no. 4299, 2024. doi: 10.1038/s41598-024-54212-8.

3. J. J. Bolognia, J. L. Jorizzo, and J. V. Schaffer, Dermatology, 4th ed. Philadelphia, PA: Elsevier Saunders, 2017.

1.5 Overview

The SRS will detail the specific requirements for the HealthBot+ application, including both functional and nonfunctional requirements. The document is organized to first introduce the purpose and scope of the system, followed by definitions and references. Subsequently, it will delve into detailed descriptions of system features, external interface requirements, system attributes, and design constraints. The organization ensures a logical flow from general to specific information, aiding in comprehensive understanding and ease of reference.

2. Overall Description

[This section of the SRS describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand. Include such items as:

2.1 Product Perspective

HealthBot+ is an AI-powered web application designed to enhance the early detection and management of skin diseases. It leverages advanced image analysis and text-based inputs to accurately diagnose various skin conditions, including skin cancer.

2.2 Product Functions

HealthBot+ offers the following key functions:

- **Image Analysis:** Analyzes uploaded images of skin conditions to diagnose potential diseases.
- **Text-Based Input:** Allows users to input symptoms and receive diagnostic suggestions.

- **Chatbot Support:** Provides detailed medical information and advice via an oncology-trained chatbot.
- **User Management:** Ensures secure handling of user data and patient history.
- **Report Generation:** Generates medical reports based on diagnostic results.

2.3 User Characteristics

The primary users of HealthBot+ include:

- **General Users:** Individuals seeking early detection and management of skin conditions.
- **Healthcare Professionals:** Doctors and dermatologists using the application for diagnostic support and patient management.
- **Administrators:** Personnel managing the application, ensuring data privacy and system security.

2.4 Constraints

- **Data Privacy:** HealthBot+ must comply with data protection regulations to ensure user data confidentiality and integrity.
- **Accuracy:** The diagnostic models must maintain a high level of accuracy to provide reliable medical advice.
- **Performance:** The application must process and analyze images quickly to provide timely results.

2.5 Assumptions and Dependencies

- **Internet Access:** Users must have a stable internet connection to access HealthBot+.
- **Device Compatibility:** The application is assumed to be compatible with modern web browsers and devices.
- **Model Updates:** Regular updates to the AI models are assumed to maintain diagnostic accuracy and incorporate new medical knowledge.

2.6 Requirements Subsets

The requirements for HealthBot+ can be grouped into the following subsets:

- **Functional Requirements**: Describes the core functionalities such as image analysis, text input processing, and chatbot interactions.
- **Nonfunctional Requirements**: Includes performance, usability, reliability, and security requirements.
- **Interface Requirements**: Details the user interface and interaction requirements for different types of users.
- **Design Constraints**: Specifies constraints related to data privacy, system performance, and compatibility.

3. Specific Requirements

3.1 Functionality

This section describes the functional requirements of the HealthBot+ system. Each function of the system/user activity is detailed to ensure designers and testers can understand and implement the requirements effectively. The organization is by feature to facilitate clarity and coherence.

3.1.1 Image Upload and Analysis

Users need to upload images of their skin conditions for analysis. The system should process these images and provide diagnostic results.

Functional Requirements:

1. Image Upload:

- Users should be able to upload images in JPEG, PNG, or DICOM format.
- Multiple images can be uploaded in one session.

2. Image Storage:

- Uploaded images should be securely stored in Azure Blob Storage or Google Firebase or AWS buckets.

3. Image Processing:

- The system should preprocess the uploaded images to ensure they meet the required specifications for analysis.

4. Image Analysis:

- The system should use pre-trained TensorFlow/Keras models to analyze the images and identify potential skin conditions.

5. Result Display:

- Diagnostic results should be displayed to the user, including the probability of different skin conditions.

3.1.2 User Authentication and Authorization

HealthBot+ requires a secure user authentication and authorization system to ensure that only registered users can access the application and its functionalities.

Functional Requirements:

1. User Registration:

- Users must be able to register using their email address and password or OAuth.
- The system should send a verification email to the user for account activation.

2. User Login:

- Registered users should be able to log in using their email and password.
- Google OAuth should be supported to allow users to log in using their Google accounts.

3. Account Management:

- Users should be able to reset their passwords via a password reset link sent to their registered email.
- Users should be able to update their profile details.

3.1.3 Medical Reports and Data Management

The system should generate and manage medical reports based on the analysis results and user inputs.

1.Report Generation:

- After analysis, the system should generate a comprehensive medical report that includes images, diagnostic results, and explanations which can be reviewed by the doctor.

2. Data Storage:

- Generated reports should be securely stored and accessible to the doctor at any time.

3. Data Privacy:

- The system must ensure that all user data, including images and reports, are encrypted and comply with relevant data privacy regulations.

4. User History:

- Users should be able to view their history of uploads and reports and medical history.

3.1.4 User Dashboard

A user-friendly dashboard for users to manage their profiles, view analysis results, and access their medical reports.

Functional Requirements:

1. Profile Management:

- Users should be able to update their profile information, including personal details and medical history.

2. Result Tracking:

- The dashboard should display a history of all uploaded images and their analysis results.

3. Report Access:

- Users should be able to download and share their medical reports from the dashboard.

3.1.5 Chatbot Assistance

The system includes an oncology-trained chatbot to assist users with information about their conditions and general dermatological advice.

Functional Requirements:

1. Chatbot Interaction:

- Users should be able to interact with the chatbot via a user-friendly chat interface.

2. Information Retrieval:

- The chatbot should provide detailed information about various skin conditions, including symptoms, treatments, and preventive measures.

3.1.6 Administrative Functions

Administrators need access to system management functions to maintain and oversee the HealthBot+ application.

Functional Requirements:

1. User Management:

- Administrators should be able to manage user accounts, including approving new registrations and handling user inquiries.

2. System Monitoring:

- The system should provide tools for monitoring system performance and usage statistics.

3.2 Usability

This section outlines the usability requirements for HealthBot+, ensuring that the system is intuitive, efficient, and user-friendly for both normal and power users.

3.2.1 Training and Task Efficiency

1. Training Time
   - **Normal Users:** Normal users should achieve productivity within 5-10 minutes of initial training.
   - **Power Users:** Power users should achieve productivity within 2-5 minutes of advanced training.

2. Task Times
   - **Image Upload and Analysis:** Users should upload images and receive initial analysis results within 2 minutes.
   - **Chatbot Interaction:** Users should obtain relevant information and support within 1 minute of initiating a query.
   - **Report Generation:** Automated medical reports should be accessible within 5 minutes of completing analysis.

Comparison to Existing Systems

- **Benchmarking:** Usability requirements will be benchmarked against leading healthcare applications known for their user-friendly interfaces, such as SkinVision and similar AI-powered diagnostic tools.
- **User Feedback:** Systematically collecting feedback on usability, interface design, and overall user experience.

3.2.2 Usability Standards

1. Common Usability Standards

- **GUI Standards**: HealthBot+ will conform to Microsoft's GUI standards, ensuring a familiar and intuitive interface for users.
- **Accessibility:** The application will meet WCAG 2.1 standards to ensure accessibility for users with disabilities.

2. Usability Metrics

- **User Satisfaction:** Measured through surveys and feedback forms, aiming for a satisfaction rate of at least 90%.
- **Error Rates:** The system will track and aim to minimize user errors, with a target error rate of less than 5% for typical tasks.
- **Task Completion Rates:** Aim for a task completion rate of 95% or higher for all critical functionalities.

3.2.3 User Interface Design

1. Intuitive Navigation

- **Consistent Layouts:** Ensure consistent layout and navigation elements across all pages.
- **Clear Instructions:** Provide clear, concise instructions and tooltips for all functionalities.

2. Feedback Mechanisms

- **Error Messages:** Display clear, actionable error messages to guide users in resolving issues.

3.2.4 User Testing

Continuous Improvement

- **Iterative Updates:** Implement iterative updates based on user feedback and usability testing results.
- **Post-Launch Support:** Provide ongoing support and updates to address usability concerns and enhance user experience.

3.2.5 Additional Core Features

1. Clarity

- **System Understanding:** Ensure the system's operation is easy to understand for all users.
- **Design Elements:** Use readable fonts, high-contrast colors, and clear visual hierarchy to enhance clarity.

2. Responsiveness

- **Smooth Navigation:** Ensure smooth navigation throughout the application with minimal technical issues.
- **Performance:** Optimize the application for speed and responsiveness to provide a seamless user experience.

3.3 Reliability

3.3.1 Availability

The system will be available 99.95% of the time, ensuring minimal downtime. Maintenance activities will be scheduled during non-peak hours, and any required downtime will be communicated to users in advance. The system should support degraded mode operations to provide basic functionality even during partial failures.

3.3.2 Mean Time Between Failures (MTBF)

The MTBF for the system is targeted to be 10,000 hours, indicating a high level of reliability and robustness in operation. This means the system is expected to run continuously for this duration without encountering critical failures.

3.3.3 Mean Time To Repair (MTTR)

In the event of a system failure, the MTTR should not exceed 2 hours. The system must be designed to facilitate quick identification and resolution of issues to minimize downtime and maintain service continuity.

3.3.4 Accuracy

The system's output must maintain an accuracy of at least 85% in diagnosing skin diseases, as measured against a known medical standard. The deep learning model must be trained and validated rigorously to achieve this level of precision.

3.4 Performance and Security

The system's performance characteristics are critical to ensuring a seamless and reliable user experience. This section outlines the specific performance metrics and security measures for the HealthBot+ system.

3.4.1 Performance Requirements

1. Response Time

- **Average Response Time:** The average response time for uploading an image and receiving the diagnostic result should not exceed 5 seconds.
- **Maximum Response Time:** The maximum response time, under peak load conditions, should not exceed 10 seconds.
- **Chatbot Response Time:** The chatbot should respond to user queries within 2 seconds on average.

2. Throughput

- **Image Analysis Throughput:** The system should be capable of processing at least 50 image analysis requests per minute.
- **Concurrent Users:** The system should support at least 500 concurrent users without significant performance degradation.

3. Capacity

- **User Capacity:** The system should accommodate up to 10,000 registered users.
- **Data Storage Capacity:** The system should be able to store up to 1 TB of image data and associated medical reports.

4. Resource Utilization

- **Memory Utilization:** The system should operate within the allocated memory limits, with an average memory utilization of no more than 70%.
- **Disk Utilization:** Disk usage should be monitored to ensure it does not exceed 80% of total capacity to allow room for peak loads and unexpected data spikes.
- **Network Utilization:** Network bandwidth should be optimized to handle up to 100 Mbps of data transfer during peak times without significant latency.

3.4.2 Security Requirements

1. Data Encryption

- **In-Transit Data Encryption:** All data transmitted between the client and server should be encrypted using HTTPS with TLS 1.2 or higher.
- **At-Rest Data Encryption:** All stored data, including images and medical reports, should be encrypted using AES-256 encryption.

2. Data Privacy and Compliance

- **Data Anonymization:** Personal identifiable information (PII) should be anonymized where possible to protect user privacy.
- **Regulatory Compliance:** The system should comply with relevant data protection regulations such as GDPR and HIPAA, ensuring secure handling of medical data.

3.5 Supportability

This section outlines requirements aimed at enhancing the supportability and maintainability of HealthBot+, ensuring efficient ongoing support and system maintenance.

3.5.1 Bug Tracking and Resolution

- **Issue Tracking:** HealthBot+ will implement a robust issue tracking system to manage bug reports and feature requests. Users will be able to report issues through the web application, and the system will provide timely updates on the status of reported issues.

3.5.2 Code Reusability and Dependencies

- **Modular Architecture:** HealthBot+ will prioritize code reusability through a modular architecture, allowing for efficient integration and reuse of code components across different modules and updates.
- **Documentation:** The codebase will be well-documented and organized to facilitate easy navigation and understanding, promoting code reuse and minimizing development time for new features and enhancements.
- **Coding Standards:** HealthBot+ will adhere to industry-standard coding practices such as consistent indentation, naming conventions (e.g., camelCase for variables, PascalCase for classes), and commenting guidelines. These practices ensure code readability and maintainability.
- **Class Libraries:** Modular design principles will be employed, utilizing reusable class libraries to facilitate code reusability and simplify updates or modifications without impacting the entire system.
- **Dependencies:** All external dependencies and libraries used in HealthBot+ will be documented and kept up-to-date to minimize conflicts and ensure system stability.

3.5.3 Maintenance Access and Utilities

- **Access Control:** Roles and access levels for maintenance personnel and administrators will be defined, implementing secure access protocols to safeguard sensitive data and maintain system integrity.
- **Maintenance Tools:** Diagnostic tools, logging mechanisms, and monitoring utilities will be integrated into HealthBot+ to enable proactive maintenance, troubleshooting, and real-time insights into system performance and usage patterns.

3.5.4 Documentation and Knowledge Transfer

- **Documentation:** Comprehensive documentation covering system architecture, APIs, data models, deployment procedures, and maintenance workflows will be maintained and regularly updated to reflect system changes and improvements.
- **Knowledge Transfer:** Regular knowledge transfer sessions will be conducted to ensure that new team members and support staff are proficient in HealthBot+'s functionalities, maintenance procedures, and troubleshooting techniques.

3.5.5 Version Control and Change Management

- **Version Control:** A robust version control system like Git will be used to manage code versions, track changes, and facilitate collaborative development and bug fixing.
- **Change Management:** A structured change management process will be implemented to evaluate, approve, and document system changes. This process will include rigorous testing procedures and rollback plans to mitigate risks associated with updates or modifications.

3.5.6 Updates and Releases

- **System Updates:** HealthBot+ will have a structured process for releasing updates and enhancements to the system. Updates will undergo rigorous testing to ensure stability and functionality before deployment.
- **Notification System:** Users will be notified of updates and changes through the web application to keep them informed about new features, improvements, and changes in functionality.

3.5.7 Compatibility with Operating Systems

- **OS Compatibility:** HealthBot+ will maintain compatibility with the latest operating systems to ensure seamless operation for users. Compatibility checks will be conducted regularly, and any issues will be promptly addressed and communicated to users.

3.5.8 Self-Service Support Resources

- **Knowledge Base:** HealthBot+ will provide self-service support resources such as FAQs, knowledge bases, and video tutorials accessible through the web application. These resources will be regularly updated to ensure accuracy and relevance.

3.5.9 User Feedback

- **Feedback Mechanisms:** HealthBot+ will implement mechanisms for gathering and addressing user feedback and feature requests. This may include a feedback form within the application or a dedicated forum where users can submit and vote on ideas for improvements.
- **Feedback Handling:** User feedback will be monitored and prioritized to align new features and enhancements with user needs, ensuring continuous improvement and user satisfaction.

3.6 Design Constraints

3.6.1 Standards Compliance

- **HIPAA (Health Insurance Portability and Accountability Act):** To ensure the privacy and security of user medical information, the system must implement necessary safeguards, such as encryption and secure access controls, to protect patient data.
- **GDPR (General Data Protection Regulation):** To comply with European data protection laws, the system must provide users with control over their personal data, including options for data access, correction, and deletion. Additionally, it must ensure data is processed lawfully, transparently, and for a specific purpose.

3.6.2 Hardware Limitations

- **Server Requirements:** The application must be capable of running on cloud-based infrastructure provided by AWS, specifically utilizing services such as Amazon EC2 for compute power and Amazon S3 for storage.
- **Client Devices:** The system must be optimized for performance on a range of client devices, including desktops, laptops, tablets, and smartphones, ensuring responsive design and efficient resource usage.
- **Edge Devices:** The image analysis feature must support integration with edge devices (e.g., smartphones) for initial image preprocessing to reduce server load and improve response times.

3.7 On-line User Documentation and Help System Requirements

3.7.1 Help About Notices

Requirement Description:

- **About Page:** The system should have an "About" page that provides information about HealthBot+, including its purpose, key features, and the team behind its development.
- **Version Information:** The "About" page should also include the current version of the software and any recent updates or changes made to the system.
- **Contact Information:** Users should have access to contact information for support, including an email address and a contact form for submitting queries or reporting issues.

3.7.2 User Feedback and Support

Requirement Description:

- **Feedback Form:** The system should provide a feedback form for users to submit suggestions, report bugs, or provide general feedback about their experience with HealthBot+.
- **Live Support:** Live chat support should be available during designated hours to assist users with urgent issues or questions that cannot be resolved through the documentation or help system.
- **Support Tickets:** Users should have the ability to submit support tickets for issues that require detailed attention, with a tracking system to monitor the status and resolution of their tickets.

3.8 Purchased Components

HealthBot+ will utilize the following purchased components to enhance its functionality and performance:

3.8.1 Cloud Services (Azure or AWS):

HealthBot+ will leverage cloud services from Azure or AWS for scalable computing resources, storage, and deployment capabilities. This includes hosting the application, storing image data, and managing user interactions securely in compliance with data protection regulations.

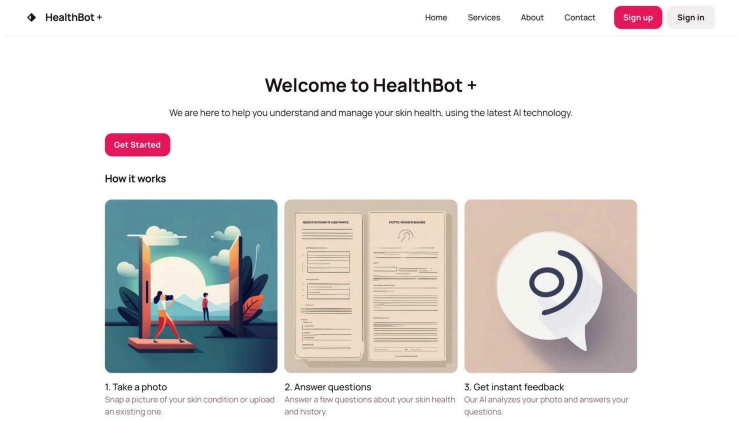
3.8.2 ChatGPT

HealthBot+ will integrate ChatGPT-3.5 Turbo, a natural language processing model provided by OpenAI, to enhance the chatbot's conversational abilities. ChatGPT will be used to provide detailed information, medical advice, and support to users regarding skin diseases.

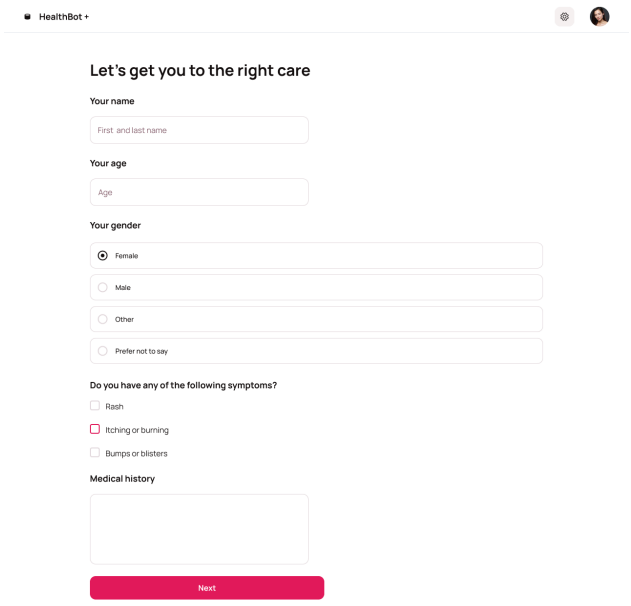
3.9 Interfaces

3.9.1 User Interfaces

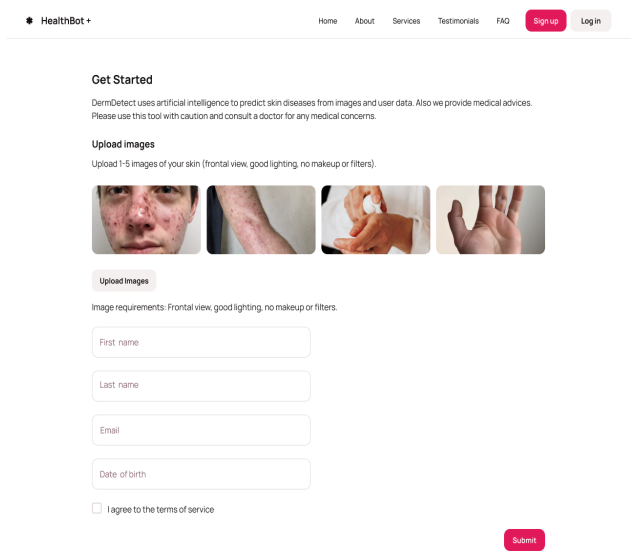
The interfaces described herein are prototype designs, serving as initial conceptual representations, and the actual implementation may vary in future development stages



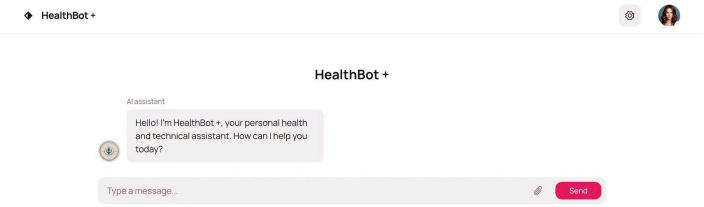
(Home Page)



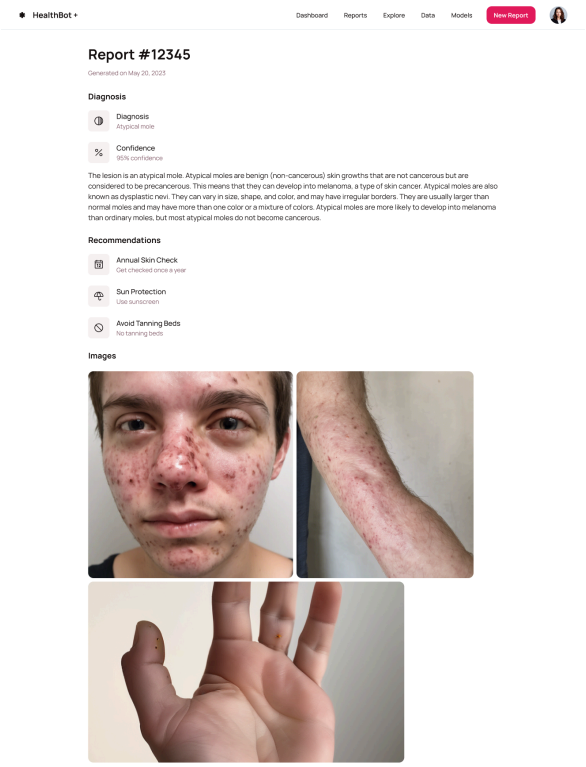
(SignUp Page)



(Image Uploading Page)



(ChatBot Page)



(Medical Report Page)

3.9.2 Software Interfaces

External APIs:

- **Machine Learning Model API:** The application communicates with a pre-trained machine learning model API that performs image analysis and returns the diagnosis results. This API is hosted on a separate server and is accessed via RESTful web services.
- **Chatbot API:** The chatbot functionality is provided by an external service like gpt-3.5-turbo. The web app interacts with this service to handle user queries and provide responses based on the context.
- **Email Service API:** For generating and sending medical reports, the application integrates with an external email service (e.g., SendGrid, Amazon SES) to handle email dispatches.

Reused Components:

- **UI Frameworks:** The web app may utilize existing UI frameworks (e.g., React, Angular) to streamline front-end development. These frameworks provide pre-built components and tools to enhance user experience.

3.9.3 Communications Interfaces

Local Area Network (LAN):

- **Internal Communication:** The web app components (e.g., front-end, back-end, database) communicate over a secure local network within the hosting environment. This ensures efficient data transfer and low latency.

Internet:

- **Cloud Storage:** Image files and other user data are stored and accessed from cloud storage solutions. Communication with these services is secured via HTTPS.
- **Email Service:** The web app sends emails (e.g., medical reports) through internet-based email services. The communication is encrypted and follows standard email protocols (SMTP, IMAP).

3.10 Database Requirements

This section defines the database requirements for the HealthBot+ system. The database is a critical component, responsible for storing and managing user data, medical images, diagnostic results, and other essential information.

3.10.1 Database Type

Requirement Description:

- **NoSQL Database:** HealthBot+ will use a NoSQL database, specifically MongoDB, to handle the storage of unstructured data such as user profiles, medical images, and diagnostic reports. MongoDB's flexible schema and scalability make it suitable for the application's needs.

3.10.2 Data Storage

Requirement Description:

- **User Data:** The database should store user information, including personal details (name, email, password), registration dates, and login history.
- **Medical Images:** All uploaded medical images must be stored in the database, linked to the corresponding user and diagnostic results.
- **Diagnostic Results:** The results of the image analysis, including probabilities of various skin conditions and detailed reports, should be stored and accessible.
- **Chatbot Interaction Logs:** Logs of interactions with the chatbot should be stored for future reference and improvement of chatbot responses.

3.11 Licensing, Legal, Copyright, and Other Notices

3.11.1 Licensing Enforcement and Usage Restrictions

- **Cloud Services:** Utilization of Azure or AWS services will comply with their respective terms of service and licensing agreements.
- **ChatGPT:** Usage of OpenAI's ChatGPT will follow OpenAI's usage policies and licensing terms.
- **Pre-trained Models and Datasets:** Usage of pre-trained models and datasets (e.g., SIIM-ISIC Melanoma Dataset) will adhere to their specific licensing agreements and usage restrictions.

3.11.2 Legal Disclaimers and Warranties

- **Disclaimer:** HealthBot+ provides diagnostic assistance and informational support but does not substitute professional medical advice. Users are advised to consult healthcare professionals for medical decisions.
- **Warranties:** The software is provided "as-is" without warranties of any kind, express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

3.11.3 Copyright, Patent, and Trademark Notices

- **Copyright:** All original content, including software code, documentation, and graphical assets, are copyright protected under applicable laws.
- **Third-Party Content:** Ensure compliance with copyright requirements for any third-party content or libraries used within the application.
- **Patent:** No specific patents are claimed for HealthBot+ unless explicitly mentioned in future legal notices.
- **Trademark:** Any trademarks used in HealthBot+ belong to their respective owners and are acknowledged. The HealthBot+ logo and name must be used in accordance with trademark laws.

3.11.4 Compliance Issues

- **Wordmark and Logo Compliance:** Any wordmarks, logos, or trademarks used in the software comply with applicable trademark laws and guidelines.
- **Data Privacy and Security Compliance:** HealthBot+ will adhere to data privacy regulations such as GDPR and HIPAA to ensure user data is protected and managed responsibly.

3.12 Applicable Standards

1. Legal and Regulatory Standards

- HIPAA (Health Insurance Portability and Accountability Act)
- GDPR (General Data Protection Regulation)

2. Quality and Security Standards

- ISO/IEC 27001 (Information Security Management)
- ISO 13485 (Medical Devices - Quality Management Systems)

3. Industry Standards

- HL7 (Health Level Seven International)

4. Usability and Accessibility Standards

- WCAG 2.1 (Web Content Accessibility Guidelines)
- ISO 9241-11 (Ergonomics of Human-System Interaction)\

4. Supporting Information

Definitions, Acronyms, and Abbreviations

Abbreviation	**Meaning**
AI	Artificial Intelligence
CNN	Convolutional Neural Network
DICOM	Digital Imaging and Communications in Medicine
XAI	Explainable Artificial Intelligence
AWS	Amazon Web Services
React	A JavaScript library for building user interfaces
Django	A high-level Python web framework
SIIM-ISIC	Society for Imaging Informatics in Medicine - International Skin Imaging Collaboration