Feasibility Study
HealthBot+

Date
2024/07/10

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

Mentor - DR. THANUJA D. AMBEGODA

Table of Contents

1.Introduction

1.1 Overview of the Project

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+ utilises 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 user-friendly interface. HealthBot+ aims to benefit both users and healthcare professionals by offering accessible, accurate, and transparent diagnostic tools.

1.2 Objectives of the Project

i) Design and Implement an AI-Powered Diagnostic Tool - Develop an application that uses pre-trained models and transfer learning to diagnose skin diseases from images and text inputs.

ii) Provide Patient Support through a Chatbot - Integrate an oncology-trained chatbot to offer medical advice and information to users, enhancing their understanding and management of skin conditions.

iii) Ensure Data Privacy and Secure Management - Implement robust data privacy measures and secure storage systems to manage patient data and history effectively.

iv) Enhance Transparency through Explainable AI - Use explainable AI techniques to make the diagnostic process transparent and understandable for users and healthcare professionals.

v) Offer Doctor Integration for Comprehensive Care - Allow for doctor consultations where healthcare professionals can review and validate diagnostic results, providing a comprehensive care solution.

1.3 The Need for the Project

The development of HealthBot+ addresses critical needs by providing an accessible tool for early detection and management of skin diseases, particularly in underserved regions with limited dermatological care. By facilitating early detection of skin cancer through accurate image analysis, HealthBot+ significantly improves treatment outcomes. Additionally, its integration of image analysis and text-based inputs offers a comprehensive approach to diagnosing various skin conditions. The oncology-trained chatbot further empowers users by providing detailed information and medical advice, enhancing patient education and support.

1.4 Overview of Existing Systems and Technologies

Existing Systems:

i) SkinVision - SkinVision is a similar product to our project, designed to assist in early skin cancer detection through image analysis using advanced machine learning algorithms. It provides a convenient method for users to monitor skin changes over time, encouraging proactive skin health management.

ii) UMSkinCheck - UMSkinCheck is a IOS mobile app developed by the University of Michigan Health System, designed to assist in early skin cancer detection by guiding users through self-examinations and tracking skin changes over time. It offers step-by-step instructions and a comprehensive mole-tracking feature to promote proactive skin health management.

Technologies:

i) Machine Learning and Deep Learning - The core technology behind skin disease diagnostic apps is machine learning, particularly deep learning. CNNs are commonly used for image recognition tasks, making them ideal for analysing skin images.

ii) Cloud Computing - Cloud platforms such as AWS, Google Cloud, and Microsoft Azure provide the necessary infrastructure for deploying and scaling web-based diagnostic applications. They offer services for data storage, machine learning model hosting, and real-time image processing.

iii) Web Development - Modern web development frameworks like React, Angular, and Django are used to build responsive and user-friendly web interfaces. These frameworks enable seamless integration with backend services and machine learning models.

1.5 Scope of the Project

The project aims to develop a web application that enables patients to identify common skin diseases, including those that may lead to cancer. After diagnosing the disease, the app features a chatbot that provides detailed information and precautions. Utilising explainable AI, the application generates comprehensive reports on the diagnosis, assisting healthcare professionals with essential information. Patients can then consult with a doctor for further evaluation and treatment.

1.6 Deliverables

The project will result in a fully functional web application for skin disease diagnosis, along with associated documentation and support materials.

i) Web Application - A responsive and interactive web application that allows users to upload user data and skin images, receive diagnostic results, and use a chatbot for providing detailed information.

ii) User Documentation - Comprehensive user guides and tutorials to help users navigate and utilise the application effectively.

2. Feasibility Study

2.1 Financial Feasibility

The financial feasibility of HealthBot+, an AI-powered web application for early detection and management of skin diseases, involves considering the costs associated with various technical resources needed for its implementation. Below are the key financial projections:

i) Azure Blob Storage - Storing medical images and patient data securely is crucial. Azure Blob Storage is a reliable and scalable solution. The cost for storage is estimated based on the volume of data from the SIIM-ISIC Melanoma Classification dataset and other datasets used. Azure Blob Storage prices are approximately $0.0184 per GB for hot storage.
ii) Website Hosting - Hosting the HealthBot+ web application requires a robust and scalable service. Azure App Service or similar cloud hosting services can be used. The estimated monthly cost for a standard hosting plan on Azure is around $73.28.
iii) Database (MongoDB) - MongoDB Atlas is a recommended managed database service. For a small cluster suitable for initial stages, the estimated monthly cost is around $57.

2.2 Technical Feasibility

i) Frontend: React.js - React.js is chosen for its efficiency in building interactive user interfaces. It enables the creation of dynamic and responsive user interfaces, which is essential for an application like HealthBot+ that requires seamless user interaction.
ii) Backend: Python Flask - Flask is a lightweight and flexible web framework in Python, making it suitable for building the backend of HealthBot+. It supports rapid development and integration with machine learning models.
iii) Database:MongoDB - MongoDB is a NoSQL database known for its scalability and flexibility in handling unstructured data. It is ideal for storing diverse data types, including patient records and medical images.
iv) Storage: Google Firebase / Azure Blob Storage - Google Firebase and Azure Blob Storage are both considered for storing medical images. These cloud storage solutions offer high availability, security, and easy integration with the web application.
v) Machine Learning Models:TensorFlow, Keras, Hugging Face - Pre-trained models from TensorFlow and Keras, along with language models from Hugging Face, will be utilised for image analysis and natural language processing. These frameworks are well-supported and provide robust tools for model fine-tuning and deployment.
vi) API Testing: Postman - Postman will be used for API testing to ensure that the backend services are functioning correctly and efficiently.
vii) Security: JWT Encryption, Google OAuth - JSON Web Token (JWT) encryption will be implemented for secure user authentication and authorization. Google OAuth will be used to streamline the login process and enhance security.
viii) State Management: Redux - Redux will be employed for state management in the frontend application, ensuring consistent and predictable application behaviour.

These technologies collectively ensure that the HealthBot+ project is technically feasible and capable of delivering accurate and reliable diagnostic results.

2.3 Resource and Time Feasibility

Resource Feasibility

i) Software Resources - All the technologies mentioned are either open-source or have free tiers available, reducing software costs. This includes React.js, Flask, MongoDB (community edition), TensorFlow, Keras, and Postman.
ii) Hardware Resources - Personal laptops and mobile phones are sufficient for the development and initial testing phases. Cloud infrastructure for hosting and storage can be utilised from services like Azure and Google Cloud.
iii) Human Resources - The project team consists of three undergraduate students specialising in computer science and engineering. Their combined skills in software development, machine learning, and data science are adequate to carry out the project successfully.
iv)Services - Hosting services are required to deploy HealthBot+. Given that the project is intended for the healthcare sector, government hosting services may be utilised to minimise costs.

Time Feasibility

i) Project Duration - The development and implementation of HealthBot+ are planned to be completed within three months. Given the scope of the project and the team's capabilities, this timeframe is feasible but ambitious.
ii) Development Timeline - The project will follow a predefined high-level timeline to ensure timely completion. This includes setting milestones for frontend development, backend integration, model training, and system testing.
iii) Potential Challenges - Factors such as the complexity of the system, personal issues among developers, and unforeseen technical challenges could cause delays. However, by adhering to a structured development plan and maintaining clear communication, these risks can be mitigated.

2.4 Risk Feasibility

i) Data Privacy and Security Risks - Ensuring patient data is protected through robust encryption methods and secure data handling practices to prevent unauthorised access and breaches.
ii) Model Accuracy Risks - Addressing the potential risks associated with false negatives (failing to diagnose a condition) and false positives (incorrectly diagnosing a condition), which could have serious health implications for users.
iii) Operational Risks - Evaluating the potential risks in the deployment and maintenance of the application, such as technical failures, downtime, and integration issues with existing healthcare systems.
iv) Compliance Risks - Ensuring that the application complies with relevant medical standards and regulations to avoid legal issues and potential penalties.
v) User Acceptance Risks - Assessing the likelihood of user acceptance and trust in the application, and planning strategies to improve user engagement and satisfaction.

2.5 Social/Legal Feasibility

i) Ethical Concerns - Addressing ethical issues related to the use of AI in medical diagnosis, including fairness, transparency, and the potential for bias in the model. Ensuring that the application does not discriminate against any group of users.
ii) Regulatory Compliance - Ensuring that the application adheres to relevant health regulations, such as HIPAA in the United States, GDPR in Europe, and other regional data protection and medical device regulations.
iii) Collaboration with Healthcare Providers - Ensuring that the application integrates smoothly with existing healthcare practices and that there is clear communication and collaboration with healthcare providers and institutions.

3. Considerations

Performance:
i) Accuracy and Reliability - Ensure the model provides accurate and consistent diagnoses.
ii) Speed and Efficiency - Deliver quick response times for user queries and diagnoses.
iii) Scalability - Support increasing numbers of users and data inputs without performance degradation.

Security:
i) Data Protection - Safeguard sensitive patient information with strong encryption and secure storage.
ii) Authentication and Authorization - Restrict access to authorised users only.
Compliance: Adhere to relevant data protection regulations like HIPAA and GDPR.

Usability:
i) User Interface Design - Create an intuitive and easy-to-navigate interface.
ii) Accessibility - Ensure the system is usable by people with varying technical skills and abilities.
iii) Feedback Mechanism - Provide clear and actionable feedback to users.

4. References

i) 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.