Skin Cancer Detection

DESIGN DOCUMENT

07
Dr. Gaffar Ashraf
Evan Hanson, Mishar Alhyarbi, Ziyad Alqahtani, Wonjun Choi
Anirudh Ambore, Abdelrahman Mohamed
sdmay24-07@iastate.edu
https://sdmay24-07.sd.ece.iastate.edu/

09/05/2023

Executive Summary

Development Standards & Practices Used

List all standard circuit, hardware, and software practices used in this project. List all the Engineering standards that apply to this project that were considered.

- Agile processes (Standards)
- Jira (Workstation)
- Git (Version Control)

Summary of Requirements

List all requirements as bullet points in brief.

- Learn basic principles and tools of Artificial Intelligence (AI)
- Develop simple AI models to recognize skin cancer
- Train the AI model on the Premises and on the Cloud
- Expand the model to train using real images from different medical institutes around the world

Applicable Courses from Iowa State University Curriculum

List all Iowa State University courses whose contents were applicable to your project.

- SE 422 (Cloud Computing)
- COM S 309 (Software Development Practices)
- COM S 317 (Introduction to Software Testing)
- ENGL 314 (Technical Communication)
- COM S 127 (Introduction to Computer Programming)
- SE 329 (Software Project Management)

New Skills/Knowledge acquired that was not taught in courses

- AWS/GCP (Cloud computing)
- Artificial Intelligence
- Model building
- Python programming
- Traditional/AI approaches to Image Recognition

Table of Contents

List of figures/tables/symbols/definitions

- AI: Utilizing software to adapt to presented problems.
- Cloud computing: Computing programs on the cloud which has more resources and computing power.
- Python: A high level programming language utilized in this project
- Skin cancer: The abnormal growth of skin cells — most often develops on skin exposed to the sun. But this common form of cancer can also occur on areas of your skin not ordinarily exposed to sunlight.

*Figures and Tables displayed in following chapters - 2,3,4,6

1 Team, Problem Statement, Requirements, and Engineering Standards

1.1 Team Members

Evan Hanson

Ziyad Alqahtani

Wonjun Choi

Anirudh Ambore

Abdelrahman Mohamed

Mishari Alharbi

1.2 Required Skill Sets for Your Project

AWS/GCP (Cloud computing)

Artificial Intelligence

Model building and training

Python Programming

Soft Skills

Traditional vs. AI approaches to Image Recognition

1.3 Skill Sets Covered by the Team

Anirudh Ambore - Python, UML diagrams

Abdelrahman Mohamed - Cloud computing

Evan Hanson - Python, Docker, Soft Skills

Mishari Alharbi - Python

Wonjun Choi - UML diagrams, soft skills

Ziyad Alqahtani - Python

1.4 Project Management Style Adopted by the team

Agile/Scrum project management

1.5 Initial Project Management Roles

Evan Hanson - Project Manager/Programmer

Ziyad Alqahatani - Researcher/Programmer

Wonjun Choi - Researcher/Programmer

Anirudh Ambore - Scheduler/Programmer

Abdelrahman Mohamed - Researcher/Programmer

Mishari Alharbi - Researcher/Programmer

1.6 Problem Statement

The project aims to address the problem of improving skin cancer detection using artificial intelligence (AI) and cloud computing technologies. The primary problem is the need for more accurate and efficient methods to detect skin cancer. Currently, skin cancer diagnosis relies heavily on manual examination by medical professionals, which can be time-consuming and subject to human error. By leveraging AI, the project intends to enhance the accuracy and speed of skin cancer detection, ultimately benefiting patients and healthcare providers.

1.7 Requirements & Constraints

Functional requirements:

- Develop an AI model capable of accurately detecting skin cancer from medical images.
- Continuously improve the accuracy and performance of the AI model over time.
- Port the AI model to a cloud-based infrastructure like GCP for scalability and remote access.
- Incorporate real medical images from different medical institutes to enhance the model's training data.
- Ability to upload new images to the database

UI Requirements: Buttons to upload images to the site, display cells, and information on if the user has skin cancer.

Performance Requirements: Users should be able to expect results on the same day, Computations should be able to take a 5-megapixel jpeg and convolute the pixels into smaller 500-pixel portions to increase runtime performance

Legal Requirements: WHO calls for a set of legal requirements when using AI in health care 1) protect autonomy, promote human well-being, human safety, and the public interest, ensure transparency, explainability, and intelligibility, foster responsibility and accountability, ensure inclusiveness and equity, finally to promote AI that is responsive and sustainable.

Testing Requirements: The model should have a 90%+ hit rate for accuracy in deciding if an image contains cancerous cells.

Resource Requirements: Utilize AWS cloud to allow for computations to run for building the model running computations require 100's of cores in parallel so utilizing third-party hardware (AWS) to re-teach the model.

Maintainability Requirements: Ensure that the AI model can be updated and maintained to accommodate changes in skin cancer detection practices.

Qualitative Aesthetics Requirements: For patient education, users must have a basic understanding of what to look out for and how to know if symptoms begin to appear. Texture and elevation of the skin are also important to take into account, Finally, color consistency if a user discovers a patch of color then they should be able to utilize our service

Resource Requirements: We would need a high-power GPU computer for model development and training. We would also need access to cloud computing resources like AWS or GCP for model scalability and cloud-based training.

Economic Requirements: While the project itself may not require financial resources from the client, there may be economic considerations, such as the potential cost savings for healthcare institutions through more efficient skin cancer diagnosis. Additionally, if the AI model is successful, it could create market opportunities for AI-driven healthcare solutions.

Constraints:

- The model should achieve a minimum specified level of accuracy, which will be determined based on medical standards and expert feedback.
- The improvement process should be based on available data and expert input, and the model's performance should be consistently monitored.
- The cloud infrastructure must comply with relevant security and privacy standards, especially when handling medical data
- Ensure the ethical and legal use of medical images, complying with data privacy regulations such as HIPAA, and obtaining appropriate permissions for data access and usage.

1.8 Engineering Standards

Software Development Practices: Kanban/agile and scrum methodologies for project management and development
Data Privacy Regulations: Compliance with data privacy laws (e.g., GDPR, HIPAA) when handling medical data.

Cloud Computing Standards: Following best practices for cloud-based infrastructure and data security.

Machine Learning and AI Ethics: Adhering to ethical guidelines and standards for AI model development and deployment.

Educational Standards: Ensuring that the educational program meets relevant teaching and curriculum standards.

1.9 Intended Users and Uses

Our project helps in recognizing skin cancer which benefits people who want to identify potential issues, which helps in promoting self-awareness. They can use it whenever they are suspicious about having skin cancer.
More specifically:

Students: They will benefit from training in AI and cloud computing, gaining skills that can be applied in various fields.

Mayo Clinic: The clinic could potentially adopt and benefit from the AI model for skin cancer detection, enhancing its diagnostic capabilities.

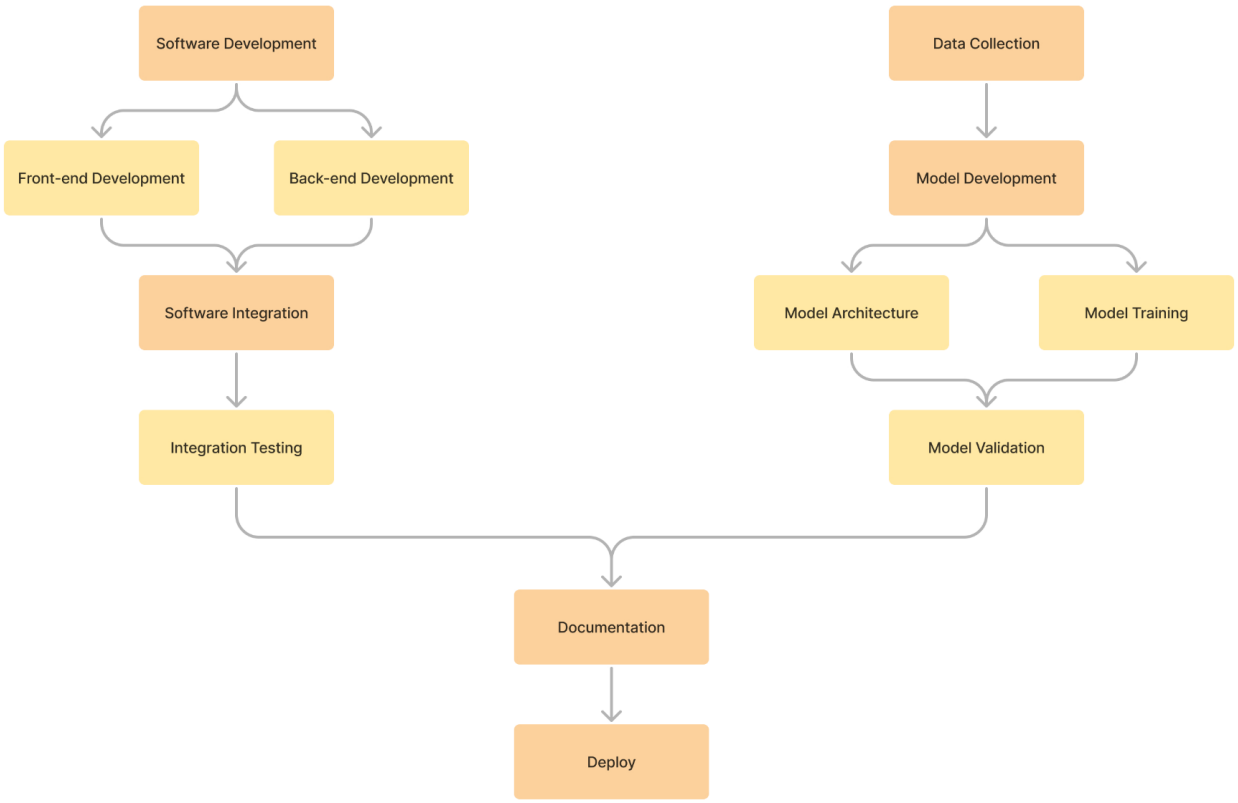
Medical Professionals: The AI model can assist healthcare professionals in skin cancer diagnosis, potentially reducing misdiagnosis rates and improving patient outcomes.

Patients: Faster and more accurate skin cancer detection can lead to early diagnosis and timely treatment, improving patient health.

Other Medical Institutes: The project can provide a blueprint for implementing AI-based skin cancer detection systems in other medical institutions.

2 Project Plan

2.1 Task Decomposition



- **Software Development**
    - Software Development: The core foundation of the project, ensuring a solid software infrastructure is essential to meet the project's requirements.
    - Frontend Development: Creating an intuitive and responsive user interface to ensure an optimal user experience.
    - Backend Development: Developing a backend to handle data processing and integration with the machine learning model.
- **Software Integration**
    - Software Integration: Merging the frontend and backend components to work seamlessly together.
    - Integration Testing: Ensuring that integrated components work as expected and identifying any bugs or issues.
- **Data Collection**
    - Data Collection: Gathering the necessary data to train and validate the machine learning model.
- **Model development**
    - Model Development: Developing the machine learning model to fulfill the project's objectives based on collected data.
    - Model Architecture: Designing the architecture of the machine learning model.
    - Model Training: Training the model with the collected data to achieve desired accuracy.
    - Model Validation: Validating the model's performance against a set of criteria to ensure it meets the project's requirements.
- **Documentation**
    - Documentation: Creating comprehensive documentation to explain the system's functionality and provide guidelines for its use and maintenance.
- **Deploy**
    - Deploy: Deploying the completed system to ensure that it is ready for use by end users.

2.2 Project Management/Tracking Procedures

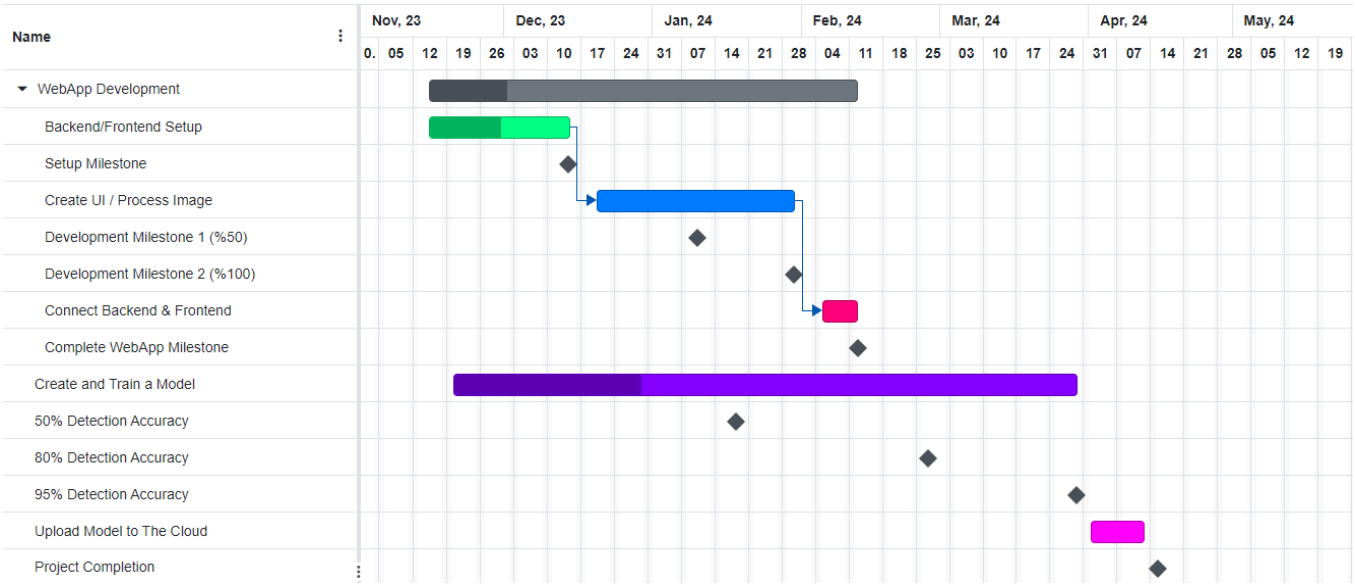
Our management style will combine waterfall and agile since we will be working in a waterfall-style progression from design to implementation to testing but will be holding agile-styled goals/ tracking methods. Progress tracking will be utilized through the usage of the GitLab issue board (Kanban), Creating a main trunk in GitHub, which will further be split into a main backend branch and a main frontend branch for version control. Our communication tools will be through GitLab QA board and Discord for informal communication.

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

- Fronted Development:
    - Milestone 1 (50% completion): Developing a basic user interface mockup. Measured by the completion of the mockup.
    - Milestone 2 (100% completion): Implementing frontend functionality for user input. Measured by having a functional user input.
- Backend Development:

    - Milestone 1 (50% completion): Implementation of the core backend logic. Measured by how much of the logic is done.
    - Milestone 2 (100% completion): Implementation of the database integration. Measured by completing the database integration.
- Integration Testing:
    - Milestone 1 (50% completion): Initial integration of frontend and backend. Measured by completion of the basic integration.
    - Milestone 2 (100% completion): Comprehensive integration testing with user scenarios. Measured by how many tests have been completed.
- Data Collection:
    - Milestone 1 (50% completion): Data collection identification and setup. Measured by identifying the data source.
    - Milestone 2 (100% completion): Data quality assurance. Measured by having qualified data and when it is ready to be used.
- Model Architecture:
    - Milestone 1 (50% completion): Implementation of the model architecture. Measured by the completion of the architecture.
    - Milestone 2 (100% completion): Model architecture optimization. Measured by having an optimized architecture.
- Model Training:
    - Milestone 1 (50% completion): Initial model training on a subset of data. Measured by the completion of the training.
    - Milestone 2 (100% completion): Calibrating the model with 80% of the dataset and achieving target accuracy. Measured by achieving the target accuracy.
- Model Validation:
    - Milestone 1 (50% completion): Validating the model with the last 20% of the dataset. Measured by the completion of the validation.
    - Milestone 2 (100% completion): Final model validation and testing. Measured by passing the final validation.
- Documentation:
    - Milestone 1 (50% completion): Creating a project documentation. Measured by the completion of the document.
    - Milestone 2 (100% completion): Reviewing and finalizing the project documentation. Measured by having a finalized version of the document.
- Deployment:
    - Milestone 1 (50% completion): Preparing the deployment environment. Measured by having a ready environment.
    - Milestone 2 (100% completion): Deploying the software to the desired environment. Measured by a successful operational deployment.

2.4 Project Timeline/Schedule



2.5 Risks And Risk Management/Mitigation

Task 1) Creating the Login/Homepage and Database:

Issues that can be created are formatting of the page not being set as desired or the database failing to store information in correct locations; Risk 0.1

Task 2) Create UI for uploading Images/ Backend process and label image

Issues: Images do not upload as desired, and the backend is not able to correctly label an image. The risk is low as multiple examples of how to do this online exist. Risk 0.1

Task 3) Connect the Backend and front end

Loss of package information while sending .jpeg files from the front end to DB. Another risk can be HTTP requests not being received by the back end or front end; the risk is low as the team's experience covers all of these criteria. Risk 0.4

Task 4) Training Model for Skin Cancer

Model accuracy is lower than 90 percent. The risk is high for failure, as the team is new in the AI field. To mitigate this problem, we plan to reach out to our advisor in the event that we can not accurately train the AI model to detect skin cancer with an accuracy of 90 percent. Risk 0.4

Task 5) Upload result to Front end and display/Process .jpeg using AI

Risk is low as once we have reached this point in the project, simply uploading results will be a simple task that will have a low chance of failure. Risk 0.2

2.6 Personnel Effort Requirements

Member's Name	Backend Frontend Setup	Create UI/ Process images	Connect backend and frontend	Train the model	Update the results to the web page	Person-hours
Abdelrahman Mohamed	20 hrs.	15 hrs.	5 hrs	5 hrs	0 hrs	40 hrs +
Anirudh Ambore	20 hrs.	15 hrs.	5 hrs	10 hrs	0 hrs	40 hrs +
Evan Hanson	20 hrs.	15 hrs.	5 hrs	10 hrs	0 hrs	40 hrs +
Jun Choi	20 hrs	15 hrs.	5 hrs	5 hrs	5 hrs	40 hrs +
Mishari Alharbi	20 hrs.	15 hrs.	5 hrs	5 hrs	5 hrs	40 hrs +
Ziyad Alqahtani	20 hrs.	15 hrs.	5 hrs	5 hrs	0 hrs	40 hrs +

2.7 Other Resource Requirement

- Keras which is an open-source library that provides a Python interface for artificial neural networks.
    - We will be using it to learn more about AI and Machine Learning as none of our group
- AWS which is an on-demand cloud computing platform
    - We will be using it for deploying our software
- GitLab, which is a DevOps software package that can develop, secure, and operate software.
    - We will be using it for developing and managing our software.

3 Design

3.1 Design Content

The design content in our project can be described by the following:

1. Data Collection and Preprocessing: Designing a strategy to collect and preprocess skin cancer images for training and testing the AI model.
2. Training Strategy: Designing the training process, which involves specifying how the model will learn from the data, how data will be fed to the model, and how the model's performance will be evaluated.
3. Accuracy Improvement Techniques: Planning for techniques to enhance the model's accuracy over time.
4. Cloud Deployment Architecture: Designing the architecture for deploying the AI model on the cloud.
5. Security and Compliance: Ensuring that the design incorporates security measures to protect patient data and complies with healthcare data privacy regulations, such as HIPAA
6. Scalability: Considering how the design can scale to handle larger datasets and increased demand for skin cancer diagnosis.
7. User Interface Design:The project involves a user interface for medical professionals or patients to upload a skin image for diagnosis.
8. Documentation and Reporting: Preparing documentation that outlines the project's design, methods, and findings.

3.2 Design Complexity

1. The design consists of multiple components/subsystems that each utilize distinct scientific, mathematical, or engineering principles
    a. Learn Image Recognition (Mathematical/Scientific)
        i. Learning math functions utilized while building a model
        ii. Training an AI model on a dataset
        iii. Pre-processing Data
        iv. Running an AI model
    b. Transfer Processing to cloud computing (Engineering)
        i. AWS
        ii. Backend (TS/Java)
    c. Modern Frameworks (Engineering)
        i. Flutter
        ii. JavaScript
        iii. C++
2. The problem scope contains multiple challenging requirements that match or exceed current solutions or industry standards.

We will be learning modern techniques utilized in machine learning to distinguish cancerous lesions from non-cancerous lesions. Another problem we will be attempting to tackle is the issue of keeping high accuracy among all skin colors, as studies have shown patients with darker skin complections return false positives at higher rates, so we will work at reducing these false positives. We will also be moving our computations to be fully run through the cloud to reduce stress on user devices and make our service more accessible to the broader population. These two challenges are exceeding current industry standards as we are taking one step further than what has been accomplished in the past.

3.3 Modern Engineering Tools

- **Keras** - Keras is an open-source deep learning framework that played a central role in building the AI model for skin cancer detection. Its primary role was to facilitate the development, training, and evaluation of the machine learning model. Keras provides a user-friendly interface for designing and configuring neural networks, making it an essential tool for the project.
- **Amazon Web Services (AWS)** - AWS is a cloud computing platform that was chosen for the deployment of the skin cancer detection system. Its role is to provide the infrastructure and resources needed to host and run the AI model in the cloud. AWS ensures scalability, reliability, and accessibility, making it a crucial tool for deploying the system.
- **Flutter** - Flutter is a modern UI toolkit that was used to design the user interface for the skin cancer detection system. Its role is to create an intuitive and user-friendly interface that allows users, including healthcare professionals and patients, to upload skin images and view diagnostic results. Flutter's cross-platform capabilities ensure that the system is accessible on various devices.
- **Jupyter Notebook** - Jupyter Notebook, as an interactive computing environment, will play a pivotal role in the project, serving as the platform for model development, data preprocessing, and documentation. Its interactive nature enables the team to experiment with machine learning algorithms, visualize data, and document the development process efficiently.

3.4 Design Context

Area	Description	Examples
Public health, safety, and welfare	This project will benefit health and safety since this will directly influence how users can determine whether or not. they may have skin cancer. Health institutions such as the Mayo Clinic can also have shorter wait periods for patients as users can immediately get	Increasing/reducing exposure to Skin Cancer deaths through spreading awareness to better assess if a user has skin cancer or not.

	a result of whether a lesion is cancerous or not.	
Global, cultural, and social	Populations with little access to medical facilities will now be more likely to seek treatment if they suspect they may have cancer, leading to fewer deaths globally and more people getting the medical attention they need.	Users may be hours from the nearest hospital, but they have noticed a strange skin lesion has shown up now, instead of having to take a day off of work or being forced to choose between health and work, they can get an answer quickly
Environmental	A software project that uses machine learning to analyze skin images for cancer detection may indirectly impact the environment through energy consumption, data storage, hardware requirements, data privacy, and medical equipment usage. Developers should consider energy efficiency, data security, and responsible e-waste practices to mitigate these environmental effects. Additionally, promoting telemedicine can reduce the environmental footprint related to patient travel.	A skin cancer detection software that employs machine learning may indirectly impact the environment by consuming energy, necessitating data storage, and requiring hardware. It also raises considerations regarding data privacy, medical equipment usage, and potential reductions in transportation-related emissions through telemedicine. Developers should focus on energy efficiency, data security, and promoting remote medical services to minimize these environmental effects.
Economic	The project's economic impact includes cost savings, labor market effects, business opportunities, consumer costs, data and privacy considerations, and contributions to research and development. It influences both microeconomic and macroeconomic dynamics within companies, communities, and nations.	The development of skin cancer detection software utilizing machine learning technology carries significant economic implications. It introduces new revenue streams for healthcare providers and dermatologists, potentially reshapes the labor market, reduces costs for patients, encourages the growth of the AI health-tech sector, and fosters investment in data security. Additionally, ongoing research and development in this field contribute to the expansion of the AI and healthcare industries.

3.5 Prior Work/Solutions

In this expanding industry, there have been several attempts to produce a product that used AI to diagnose skin cancer. Some of these attempts were successful and had a huge impact towards this issue. Some of these successful products are:

1. SkinVision:
    - Utilized AI and machine learning algorithms to analyze images of moles or skin lesions captured by users.
    - Provided a risk assessment and recommended whether users should consult a dermatologist.

2. DermEngine:
    - Offered a suite of dermatology tools for healthcare professionals, including AI-powered image analysis.
    - Utilized computer vision and machine learning to assist in the evaluation of skin lesions, providing insights to dermatologists.
3. First Derm:
    - Allowed users to submit photos of various skin conditions, including potential skin cancer concerns, for evaluation by dermatologists.
    - While it had a dermatology focus, it did not rely solely on AI but rather connected users with healthcare professionals.

3.6 Design Decisions

1. Choice of Machine Learning Framework (Keras):
    - A critical design decision revolves around the selection of the machine learning framework for developing the skin cancer detection model. The team has chosen Keras, an open-source deep learning library, for its ease of use and comprehensive capabilities. Keras provides a user-friendly interface making it an excellent choice for building and training the machine learning model. This decision significantly influences the development and training of the model.
2. Deployment Infrastructure (AWS):
    - Another key decision relates to the choice of deployment infrastructure. The team has decided to leverage Amazon Web Services (AWS) as the cloud computing platform for deploying the skin cancer detection system. AWS offers scalable and reliable cloud resources, enabling the project to achieve optimal performance and accessibility. This choice ensures that the system can handle computations, accommodate user demand, and provide a robust and efficient deployment environment. Selecting AWS as the deployment infrastructure is pivotal to the success of the project.
3. User Interface Design (Flutter):
    - Designing an intuitive and user-friendly interface for uploading images and displaying results is a crucial design decision. The team has decided to use Flutter, a versatile UI toolkit, to design the user interface. Flutter allows for the development of applications that can run on Android, iOS, and desktop platforms, ensuring cross-device compatibility. This design decision ensures that the skin cancer detection system can be accessed and used seamlessly on a wide range of devices, enhancing user accessibility and the project's educational objectives.

3.7 Proposed Design

To automatically diagnose skin cancer from photos, our team made a specific design. Our main goal was to design a system that was both accurate and easily accessible to a wide range of users, including those in medically underserved areas. Image Collection and Preprocessing: An effective AI model begins with the right data. Our strategy defined the methods of collecting skin cancer

images and the necessary preprocessing steps to ensure that these images are ready for model training and testing.

Training Strategy: This encompasses how the AI model learns from skin cancer images. It includes feeding the model with preprocessed data and employing specific metrics to evaluate its performance. The emphasis was on iterative learning, where the model's accuracy could be improved with each cycle.

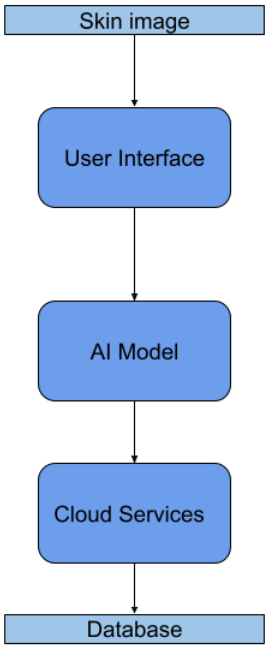
Cloud Deployment on AWS: Recognizing the need for a scalable and robust system, we decided on Amazon Web Services for deploying our AI model. This would allow healthcare professionals and patients to access our system without geographical constraints or performance issues.

User-Friendly Interface with Flutter: The interface is the bridge between our AI model and the end-users. Using Flutter, we designed an intuitive platform that would simplify the process of uploading images and viewing diagnostic results.

By this stage, we had crafted the blueprint for our AI-based skin cancer detection system. Each component was intricately designed, keeping in line with the overarching project objectives and specific requirements.

3.7.1 Design 0 (Initial Design)

Design Visual and Description



Users start by uploading an image of their skin through the platform, which is then processed by an AI model that leverages deep learning to assess the probability that the user has skin cancer. During this process, the model can search for similar case data in a cloud database to improve the accuracy

of its decision-making. Once the AI model completes its assessment, the user is presented with a diagnosis.

Data collection and preprocessing: Strategize how to collect and preprocess skin cancer images to make them suitable for AI model training and testing.

Cloud database and architecture: Provide a large database of past cases to support and enhance real-time diagnosis, and an architecture for deploying the AI model on AWS that ensures scalability and robustness as users grow.

User interface: Create a preliminary design using Flutter, focusing on ease of use for both healthcare professionals and patients.

Functionality

The design is intended to provide a platform for all users - medical professionals and patients alike - to upload skin images and receive a quick and accurate diagnosis. Once trained, the AI model scrutinizes these images and communicates the results through a user interface. The cloud-based approach does not require any heavy installation on the user's device, and the system can be accessed and used anytime, anywhere.

Each design component is precisely crafted to meet the requirements of the project, ensuring not only functionality but also aspects such as accuracy, scalability, security, and user intuitiveness.

Functional Requirements Satisfaction

1. Image Upload and Processing

    Requirement: Users should be able to upload images of skin lesions for analysis.

    Satisfaction: The system allows users to upload images through a user-friendly interface. The AI model then processes these images, ensuring this primary functionality is met.

2. Skin Cancer Diagnosis

    Requirement: The system should provide an accurate assessment of whether the uploaded image is indicative of skin cancer.

    Satisfaction: The AI model, trained on a comprehensive dataset of skin cancer images, assesses the uploaded images. While the initial design ensures this functionality, continuous improvement in model training will enhance accuracy over time.

3. Accessibility of Previous Cases

    Requirement: Users should have access to a database of previous cases for reference and comparison.

Satisfaction: The cloud-based architecture incorporates a large database of past skin cancer cases, fulfilling this requirement and aiding in more accurate diagnoses.

Non-Functional Requirements Satisfaction

1. Scalability

Requirement: The system should be able to handle a growing number of users without performance degradation.

Satisfaction: By choosing AWS for cloud deployment, the system is inherently scalable, ensuring that it can accommodate an increasing user base while maintaining performance.

2. Accessibility

Requirement: The system should be accessible from anywhere, without geographical constraints.

Satisfaction: The cloud-based approach ensures that users can access the system from anywhere, satisfying this requirement.

3. Usability

Requirement: The system should be easy to use, catering to both healthcare professionals and patients.

Satisfaction: The user interface designed with Flutter is intuitive, simplifying the process of uploading images and viewing results, which enhances the system's usability.

4. Security

Requirement: User data, especially medical images, should be protected, and privacy should be maintained.

Satisfaction: While the initial design incorporates security measures, continuous attention and updates are necessary to ensure the highest security standards, complying with legal and ethical requirements.

5. Performance

Requirement: The system should provide quick response times, especially when generating skin cancer diagnoses.

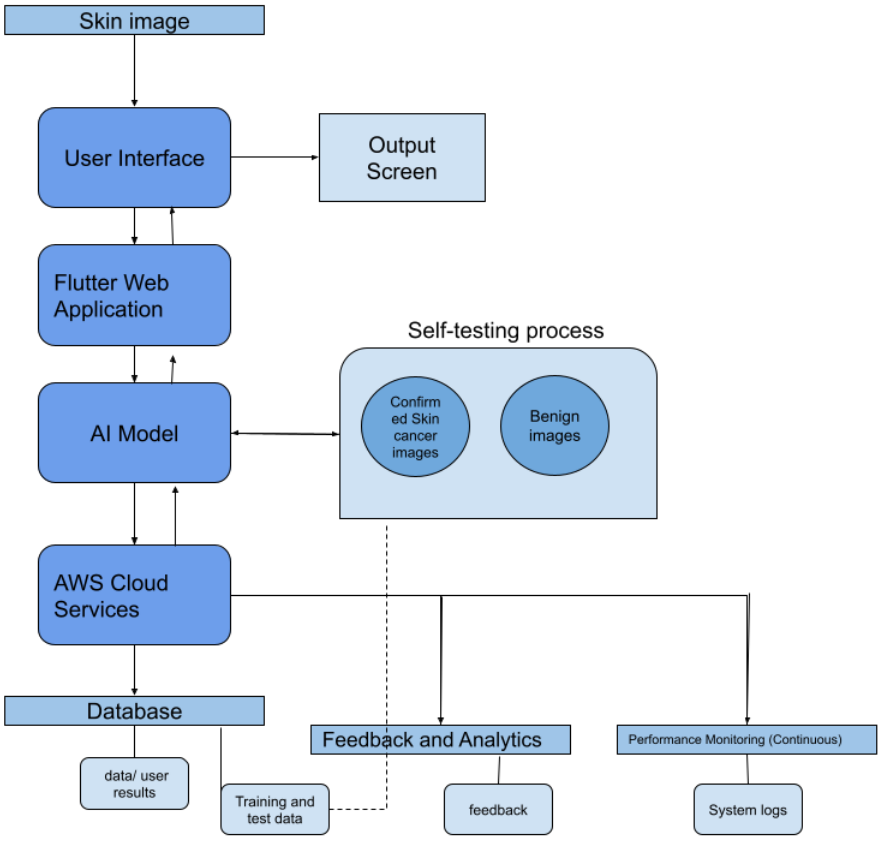
Satisfaction: The cloud architecture is designed for robust performance, and the AI model's efficiency ensures quick diagnoses, meeting the performance requirements.

6. Reliability

Requirement: The system should provide accurate and reliable diagnoses.

Satisfaction: The AI model's accuracy is central to the system's reliability. The initial design lays the groundwork for this, but ongoing model training and updates are vital to maintain and improve reliability over time.

3.7.2 Design 1 (Design Iteration)



After implementing Design 0, we conduct a comprehensive evaluation to identify areas for improvement and iterate on the initial design. These iterations are critical to refining the system, addressing shortcomings, and improving overall performance and user experience. Design 1 represents a continuous improvement and response to feedback gathered from the initial deployment of the skin cancer detection system, with the details below.

Design Iteration 1 is a " validation" phase. Post the initial deployment of our AI skin cancer detection system, a self-testing process is undertaken by using both confirmed skin cancer images and those that resemble skin cancer but are not.

Iterative Learning from AI Model Feedback:

Change: We now know how to run an iterative learning method where the AI model collects feedback from real-world diagnoses, including confirmed skin cancer cases and resembling cases.

Justification: Aligns with the requirement to continuously enhance model accuracy.

Database Improvements:

Change: Enhances the cloud database by segmenting it into categories for better searchability, including confirmed skin cancer, non-cancerous, and ambiguous cases.

Justification: Meets the requirement to incorporate diverse medical images for model training.

User Interface Improvements:

Change: Refines specific sections of the Flutter-based interface based on user feedback.

Justification: Ensures a user-friendly design, critical for the project's success.

Scalability Improvements:

Change: Optimizes the AWS backend for anticipated user base growth.

Justification: Ensures efficient performance as user demand increases.

Security Enhancements:

Change: Proactively addresses security and privacy to meet legal requirements.

Justification: Protects user data and ensures legal compliance.

3.8 Technology Considerations

Strengths

The design employs a combination of a user-friendly interface Flutter web application, a sophisticated AI model, and AWS Cloud Services. This integration leverages the strength of each component to create a seamless user experience and efficient data processing system.

- Flutter UI

  Cross-platform compatibility enables wide usage and simplifies the complex underlying technology to make it accessible to non-technical users.

- AI models

  Deliver highly accurate results through precise image analysis, which is essential for accurate skin cancer detection.

- AWS cloud services

Strong security measures ensure scalability, reliability, and efficient processing of large amounts of data.

Weaknesses

- AI models

  AI models require extensive training data to ensure sophistication and accuracy. However, this data is not always readily available, which can lead to problems with data collection and model accuracy.

- AWS cloud services

  Due to its cloud-based architecture, it relies on a consistent internet connection, which can limit accessibility in remote areas. Therefore, it can potentially limit access in areas with low connectivity.

- Complexity of AI and cloud integration can lead to maintenance challenges.

Trade-offs

Some potential features are designed as tradeoffs to balance performance and cost.

- AWS cloud services

  Provide service reliability and global reach but can be more expensive than other providers.

- Flutter

  Using web-based applications ensures faster development time and broad compatibility, but native apps can provide higher performance.

Possible Solutions and Design Alternatives

- Hybrid mobile applications

  To overcome the weakness of potentially limited access in low-connectivity areas, alternative solutions such as using hybrid mobile applications can improve the accessibility of the app by providing offline capabilities.

- Cloud services

  It can consider other cloud providers, such as Google Cloud or Azure, which offer different pricing structures or services.

3.9 Design Analysis

The proposed design integrates various technologies to facilitate the process of self-testing for skin cancer detection. Users can upload skin images through the interface, which are then analyzed by an AI model hosted on AWS cloud services. The results are stored in a database, and a feedback and analysis system provides valuable insights for continuous improvement.

Did your proposed design from 3.7 work?

This design effectively demonstrates a proof of concept for a skin cancer detection system and is successful. The feedback loop helps improve the AI model, and the scalable cloud infrastructure is key to the system's functionality.

Observations and Thoughts

While the design meets functional requirements, ongoing user feedback and system analysis can reveal areas for improvement, and it's important to identify these. For example, the user interface should be intuitive so that users can correctly upload images for analysis. The accuracy of the AI model is paramount, so continuous monitoring is required to maintain and improve performance.

Ideas for modifications or additional iterations

- Improve AI model accuracy

  Iterations could focus on increasing the diagnostic accuracy of the AI model using more diverse datasets.

- Interactive User Interface

  Streamlining the image upload process with a guided approach can enhance user engagement and ease of use.

- Offline Capabilities

  Developing offline functionalities can extend the system's reach, allowing it to serve a wider customer base, including those in areas with limited internet access.

In conclusion, there is room for further improvement and development in terms of accuracy, user interface design, and broader accessibility.

4 Testing

4.1 Unit Testing

Since the majority of our software is focused on AI and model learning, unit testing will not be very attainable. Instead, we will focus our unit test efforts on our webpage, testing the different functions, calls, and requests using tools such as Postman for the request portion, and the webpage portion, we will utilize testing software such as Apache Bench to measure our performance and response times.

Building on this foundation, we'll apply continuous integration to our development process: we'll automatically run unit tests whenever new code is committed to help us quickly identify and fix issues early in the development cycle.

4.2 Interface Testing:

We will have a webpage that can be accessed by users which will present two interfaces, one a page allowing users to upload images to our database for testing, and secondly, a results page to display the results of running their uploaded images through our model for skin cancer detection. Tools such as Screenster can be utilized for testing purposes since it is an AI-driven cloud-based system that takes in user inputs and builds off of those inputs to create a robust test set. We will focus on testing our server's ability to display results to the user along with interactions with various buttons on our page to perform as expected.

4.3 Integration Testing

The critical integration paths in our design include the interaction between the user interface, the image upload functionality, and the backend AI model. Integration testing is important to ensure proper communication between these components. For example, testing the flow from uploading an image, its reception by the model, correct processing, and accurate output generation. This critical integration path will be tested using tools like Selenium for automated testing, ensuring that the different components work together cohesively.

4.4 System Testing

For system testing, we will test the end-to-end functionality of our system. The system tests will comprise multiple lower-level tests like unit and interface tests. For example, one system test would be to upload an image of a non-cancerous skin lesion through the user interface and get the desired output. The AI model would process it and determine whether the lesion is cancerous or not and then print an output message. This would involve multiple unit and integration tests like: uploading an image, checking if the image is received by the model, checking if the model processes the image correctly, verifying if the model produces output and whether it is right. There would also be interface tests as the image would be uploaded through a UI and the output message as well. Another system test would be uploading a cancerous lesion and getting a positive result for cancer. Since system tests are composed of smaller tests, it would utilize tools used by those tests like Postman, Apache Bench, and Screenster. We can do automated testing using Selenium.

4.5 Regression Testing

In our testing approach, we will make sure that any changes or new code we add will not create unintended consequences. We will focus on important aspects, like ensuring users can still upload

images, the AI correctly processes them, and the results show up properly on the output screen. We can use test automation frameworks like Selenium as well as run the software used for our unit, integration, and interface tests with each update. By doing this regularly and focusing on how things go, we can catch and fix issues quickly. This ensures that the skin cancer detection system stays reliable and functions as it should.

4.6 Acceptance Testing

Our largest factor in assuring our requirements are met will be centered around the accuracy of our model. So, to assure proper results we will require our model's accuracy to be at least 95% before releasing it to users. We will allow our users to send in data to run analysis on and present our model's accuracy along with the results of their submissions to help the client understand what our system is capable of. Our non-functional requirements will be met by the above mentioned tactics of a mixture of unit testing, and the inclusion of various software tools such as Apache Bench, Screenster.

To ensure that the system meets our strict performance, accuracy, and overall usability requirements, we will use JMeter. JMeter is an essential tool for performance testing, allowing us to simulate different user behaviors and measure the system's response under different load conditions. This can validate that the system can handle the expected user traffic while maintaining the high level of accuracy required.

4.7 Security Testing

Security testing is important given the nature of projects that deal with sensitive medical data. The purpose of security testing is to implement fundamental and effective measures to protect the system from common vulnerabilities and safeguard sensitive data, with a particular focus on patient medical images and diagnostic results. The tests below and additional tests (password testing, data encryption, etc.) are used to build services that users can trust.

Vulnerability scanning: Perform automated scans using tools like OWASP ZAP to identify common security issues.

Access Control Checks: Manually assess the system to ensure that users can only access data appropriate to their access levels, simulating different user roles to verify proper access restrictions.

Compliance Review: Perform a basic checklist review to ensure adherence to data protection regulations such as HIPAA and GDPR, focusing on key privacy requirements.

4.8 Results

The testing results collectively demonstrate that our skin cancer detection system aligns closely with the specified requirements and design objectives. The unit testing phase focused on the webpage, ensuring its functionality and responsiveness. Interface testing confirmed smooth user interactions and the proper display of results. Integration testing verified effective communication between the user interface, image upload, and the AI model. System testing, encompassing multiple lower-level tests, showcased the robustness of our end-to-end functionalities. Regression testing, by preventing unintended consequences, ensured the stability of the codebase during updates. The

acceptance testing phase, marked by an accuracy rate of over 95%, validated the system's capability to deliver on its primary objective which is accurate skin cancer detection. User involvement in the testing process further reinforced the system's reliability and its alignment with user expectations. In conclusion, the testing results collectively affirm that our design has successfully translated into a functional and reliable skin cancer detection system. The combination of unit, interface, integration, system, and acceptance testing ensures that our system not only meets but exceeds the specified requirements, providing users with a dependable tool for early skin cancer detection. Regular updates and continuous testing will be maintained to uphold the system's performance and accuracy over time.

Unit Testing	
Test Type	Tools
Webpage performance	Apache Bench
Webpage functions	Postman

Interface Testing	
Test Type	Tools
Image upload	Screenster
Servers	

Integration Testing	
Test Type	Tools
Communication between Image upload/ reception, correct processing, and accurate output generation	Selenium

System Testing	
Test Type	Tools
End-to-end functionality of our system	Postman

	Apache Bench
	Screenster

Regression Testing	
Test Type	Tools
New/Edited code does not create unintended consequences.	Selenium
	Apache Bench
	Screenster

Acceptance Testing	
Test Type	Tools
Result accuracy	JMeter

Security Testing	
Test Type	Tools
Vulnerability scanning	OWASP ZAP
Access Control Checks	Manual checks
Compliance Review (HIPAA and GDPR)	Manual review

5 Implementation

Next Semester Plans:

Develop a user-friendly Android/IOS app so that users can upload images of skin lesions to it and have the app output whether it is cancerous or not using the AI model.

Host the application on the Cloud.

Further improve and train the model:

- Develop the model to identify multiple skin cancer types
- Improve the model to identify skin cancer in different skin colors
- Train the model further to have an accuracy of 90% or more and reduce the model's loss

6 Professionalism

6.1 Areas of Responsibility

Area of Responsibility	**Definition**	**NSPE Canon**	**SE Code of Ethics**	**Difference from NSPE**
Work Competence	Perform work of high quality, integrity, timeliness,and professional competence.	Perform services only in areas of their competence; Avoid deceptive acts.	Ensure they are qualified, by an appropriate combination of education and experience, for any project on which they work or propose to work. Be accurate in stating the characteristics of software on which they work, avoiding not only false claims but claims that might reasonably be supposed to be deceptive, misleading, or doubtful.	The SE Code of Ethics provides specific guidance related to software development, emphasizing qualifications for projects and accurate representation of software characteristics. The NSPE Canon shares the general principle of competence and avoidance of deception but applies broadly across all engineering disciplines without the specific focus on SE nuances.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs	Act for each employer or client as faithful agents or trustees	Ensure realistic estimates of cost, scheduling, personnel, and outcome on any project on which they work or propose to work and provide a risk assessment of these estimates	The SE Code of Ethics specifically addresses realistic cost estimates, scheduling, and risk assessment in software projects. The NSPE Canon outlines a general principle for engineers to act faithfully for their employers or clients without focusing on project cost estimation specific to software engineering.

Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	Be fair and truthful in all statements, particularly public ones, concerning software or related documents.	The SE Code focuses on being fair and truthful in statements concerning software or related documents, reflecting the unique context and considerations within software engineering. In contrast, the NSPE Canon applies to public statements across engineering disciplines without the specific focus on software-related communications.
Health, Safety, Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	Approve software only if they have a well-founded belief that it is safe, meets specifications, has passed appropriate tests, and does not diminish quality of life or harm the environment	The SE Code of Ethics aligns with the NSPE Canon in prioritizing safety and well-being but specifically addresses software approval. It outlines criteria for approving software, requiring a well-founded belief in its safety, meeting specifications, passing appropriate tests, and not impacting quality of life or the environment. This distinctively focuses on software's impact, ensuring it meets safety and quality standards unique to software engineering, unlike the broader

				emphasis of the NSPE Canon on public safety across engineering domains.
Property Ownership	Respect property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	Develop a fair agreement concerning ownership of any software, processes, research, writing, or other intellectual property to which an employee has contributed.	The SE Code of Ethics specifically addresses intellectual property. It emphasizes the development of fair agreements concerning ownership of software, processes, research, or other intellectual property contributed by employees. This focuses on the fair distribution of ownership rights related to software and intellectual property specific to the software engineering field, while the NSPE Canon has a broader scope across engineering disciplines without specific focus on intellectual property issues in software development.
Sustainability	Protect environment and natural resources locally and globally.		Disclose to appropriate persons or authorities any actual or potential danger to the user, a third party, or the environment, they reasonably believe to be associated with the software or related documents for which they	N/A since there is no NSPE canon for this area of responsibility.

			are responsible, or merely know about.	
Social Responsibility	Produce products and services that benefit society and communities.	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.	Work to identify, define, and address ethical, economic, cultural, legal, and environmental issues related to any work project	The SE Code of Ethics specifically emphasizes addressing ethical, economic, cultural, legal, and environmental issues related to work projects. It focuses on identifying and addressing a broader spectrum of ethical and societal considerations directly linked to software projects, reflecting the complexities inherent in software engineering's impact on various facets of society. The NSPE Canon, while promoting honorable and responsible conduct, does not explicitly address the diverse ethical and societal concerns in the same detailed manner as the SE Code within the context of specific projects.

6.2 Project Specific Professional Responsibility Areas

Area Responsibility	**Does it apply to the project's professional context? Why?**	**How our team is performing**
Work Competency	We are learning to produce models declaring whether or	High

	not someone has skin cancer. Understanding models and how they work is of high importance for our team and the patients to reduce risk of false positives or worse missed hits.	
Financial Responsibility	Since a majority of our service will be maintained through a Cloud provider such as AWS, ensuring the clients understand the financial costs of maintaining a database will be of utmost importance.	Medium
Communication Honesty	Disclose information regarding the current accuracy of our models, ensuring clients understand our system is not 100% accurate and results are not final.	Medium
Health, Safety, Well-Being	Our product's focus is determining whether user images contain cancerous tissue or not. With this in mind, accuracy and performance in these classifications are top priority, in turn improving overall health of patients.	High
Property Ownership	Since we are using pre-built models and datasets, giving proper credit to those who created the base of our models will be integral to the launch of our applications.	Medium
Sustainability	Our product is fully online, so carbon footprints will be minimal and thus is not a main concern of our design.	Low
Social Responsibility	Our application will cause an increase in overall testing of patients. This will assist in reducing deaths caused by skin cancer.	High

6.3 Most applicable Professional Responsibility Area

In the context of a skin cancer detection system, the area of "Health, Safety, Well-Being" is paramount for several reasons:

1. Patient Well-Being: The system deals directly with individuals' health by detecting potential skin cancer. Ensuring accuracy in diagnosis directly impacts the well-being of the patients relying on this technology and reducing cancer fatalities through early and reliable detection.

2. Safety and Trust: Reliable detection is critical to prevent misdiagnoses that could lead to delayed treatment or unnecessary distress for patients. Ensuring the safety of patients' health data is also crucial, especially since it involves sensitive medical information.

3. Ethical Responsibility: Accuracy and reliability in diagnosing skin conditions are ethical imperatives in the medical field. Any errors or oversights in this area could have serious consequences for patients' health and well-being.

4. Legal and Regulatory Compliance: Compliance with health data regulations (such as HIPAA, GDPR) is crucial to protect patient privacy and ensure the lawful handling of medical information.

Given these factors, the "Health, Safety, Well-Being" area is highly relevant to your project, as it directly impacts patients' lives, their trust in the system, and the ethical and legal responsibilities associated with handling medical data.

7 Closing Material

7.1 Discussion

Main results of the project:

1. Image upload and processing:
    - Users should be able to upload images of skin lesions for analysis. The system allows users to successfully upload images through the designed user interface. The AI model processes these images for analysis.
2. Skin cancer diagnosis:
    - The system should accurately assess whether the uploaded image indicates skin cancer. The AI model, trained on a comprehensive dataset of skin cancer images provided by ISIC, assesses the uploaded images. The initial design ensures accurate diagnoses, but ongoing model training is crucial for continuous improvement.
3. Scalability:
    - The cloud-based architecture, particularly the use of AWS, ensures scalability. The system can efficiently handle an increasing user base without compromising performance.
4. Usability:
    - The user interface designed with Flutter is intuitive and user-friendly, meeting the requirement of usability for a diverse user base.
5. Performance:
    - The cloud architecture is designed for robust performance, and the AI model's efficiency ensures quick diagnoses, meeting the performance requirements.

7.2 Conclusion

The project focuses on developing a cloud-based skin cancer detection system using artificial intelligence.

Goals:

1. Develop a user-friendly AI model for skin cancer detection.
2. Implement a cloud-based architecture for scalability and accessibility.
3. Integrate real images from various medical institutes to enhance model training.
4. Create a user-friendly app using Flutter for seamless interaction.

Best plan of action:

1. AI Model Development using Keras:
    - Utilize Keras to develop an AI model for skin cancer detection. Emphasize the use of diverse datasets, including real images from various medical institutes, for comprehensive model training.
2. Cloud Deployment with AWS:
    - Leverage AWS for cloud deployment, ensuring scalability, reliability, and accessibility of the skin cancer detection system.

3. User-Friendly App Development using Flutter:
    - Utilize Flutter to create a user-friendly app for seamless interaction with the skin cancer detection system. Prioritize an intuitive user interface that enhances the user experience for both healthcare professionals and patients.

Constraints:

1. Learning Curve:
    - We may face a learning curve, especially when using new tools like Keras, AWS, and Flutter.
2. Model Interpretability:
    - The complexity of some AI models, including those built with Keras, might pose challenges in terms of interpretability. Understanding and explaining the decisions made by the model may be crucial, especially in a medical context.

Future Iteration Considerations:

1. Enhanced Training Materials:
    - Comprehensive training materials to address the learning curve, ensuring that we are well-prepared to use the tools effectively.
2. Mobile App Optimization:
    - Continuously optimize the Flutter-based mobile app for different devices and operating systems to ensure a seamless and consistent user experience across various platforms.
3. Continuous Learning Modules:
    - Develop and integrate continuous learning modules for us, allowing us to stay updated on new developments in AI, cloud computing, and related technologies.

7.3 References

[1] A. Mercier, "Amplify," Amazon, https://docs.aws.amazon.com/amplify/latest/userguide/getting-started.html (accessed Dec. 3, 2023).

[2] D. Rothman *et al.*, *Python: Beginner's Guide to Artificial Intelligence: Build Applications to Intelligently Interact with the World around You Using Python*. Birmingham: Pack, 2018.

[3] "International Skin Imaging Collaboration," ISIC, https://www.isic-archive.com/ (accessed Dec. 3, 2023).

[4] K. Team, "Simple. flexible. powerful.," Keras, https://keras.io/ (accessed Dec. 3, 2023).

[5] S. Alessandria, *Flutter Cookbook 100+ Step-by-Step Recipes for Building Cross-Platform, Professional-Grade Apps with Flutter 3.10.x and Dart 3.X*. Birmingham: Packt, 2023.

7.4 Appendices

Any additional information that would be helpful to the evaluation of your design document. If you have any large graphs, tables, or similar data that does not directly pertain to the problem but helps support it, include it here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc,. PCB testing issues etc., Software bugs etc.

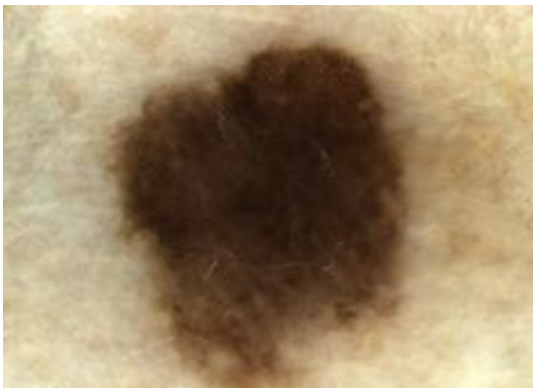
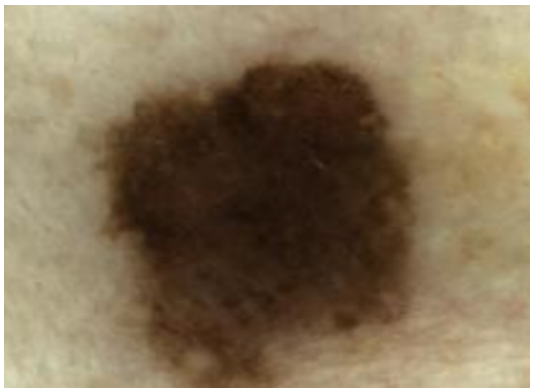
Dataset used for training, validating, and testing:

Source: The International Skin Imaging Collaboration (ISIC)

Link: [ISIC Challenge (isic-archive.com)]

Sample of images:

Malignant**:**

Benign:


Legal and Regulatory Compliance:

HIPAA Privacy Rule stated by U.S. Department of Health and Human Services:

“The HIPAA Privacy Rule establishes national standards to protect individuals' medical records and other individually identifiable health information (collectively defined as “protected health

information") and applies to health plans, health care clearinghouses, and those health care providers that conduct certain health care transactions electronically. The Rule requires appropriate safeguards to protect the privacy of protected health information and sets limits and conditions on the uses and disclosures that may be made of such information without an individual's authorization. The Rule also gives individuals rights over their protected health information, including rights to examine and obtain a copy of their health records, to direct a covered entity to transmit to a third party an electronic copy of their protected health information in an electronic health record, and to request corrections."

URL: https://www.hhs.gov/hipaa/for-professionals/privacy/index.html

General Data Protection Regulation (GDPR):

Below is a link to the official PDF of the regulation:

Link: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679

7.5 Team Contract

Team Members:

1) Anirudh Ambore 2) Abdelrahman Mohamed

3) Mishari Alharbi 4) Evan Hanson

5) Ziyad Alqahtani 6) Wonjun Choi

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:

Meetings with client (Professor Ashraf Gaffar)

- We will meet weekly on Tuesday at 2:15 PM to receive basic training on AI and cloud computing from Professor Gaffar.
- We will meet either in-person at Durham 353 or virtually through Webex.

Regular team meetings

- We are planning to meet 3 times a week
- Online/In person optional

2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):

- Face-to-face meetings are preferred but online meetings through Webex are acceptable.
- Updates will be shared with Professor Gaffar and our TA through email or face-to-face.
- We will work on the project together both in person and online.

- We will use Discord as our channel of communication among the team members regarding scheduling, issues, reminders, and other updates.

3. Decision-making policy (e.g., consensus, majority vote):

- Majority vote.
- Split vote delegated by team manager.

4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):

- Excel spreadsheet, basic stopwatch.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

- Team members are expected to attend meetings on time and be active participants.
- Members should let the team know in advance if they cannot attend.
- Team members are expected to be present for majority of the meetings and should follow up with other members if they are absent with information on what they have completed.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

- Each team member will try their best to fulfill their part when it comes to team assignments, timelines, and deadlines.

3. Expected level of communication with other team members:

- Team members are expected to respond to messages/emails by the next morning unless they are unable to do so due to external issues.

4. Expected level of commitment to team decisions and tasks:

- Team members are expected to fully commit to the team decisions and tasks.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):

Abdelrahman: Testing

Ziyad: Client Interaction

Evan: Team Organization

Wonjun: Individual component design

Mishari: Tasks Coordinator

Anirudh: Schedule and Planning Coordinator

2. Strategies for supporting and guiding the work of all team members:

- Track progress with Agile/Kanban board, if issues @ person in discord with a link to the ticket you have issues with

3. Strategies for recognizing the contributions of all team members:

- Using Git/Agile, we will know which team member has finished their task.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.

   Abdelrahman: Industry and Tutoring experience. Debugging/Testing

   Anirudh: Industry and research experience

   Evan: Industry Experience, Testing, Debugging

   Mishari: TAing experience, problem solving skills

   Wonjun: Industry Experience, Testing, Project management

   Ziyad: Experienced with different programming languages, good at verbal communication,

2. Strategies for encouraging and supporting contributions and ideas from all team members:

- Snacks, code pass testing, achieving goals.

3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)

- Anonymous polls to check on team members.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

- Learning the basics of models, Learning AI, Understanding Cloud computing

2. Strategies for planning and assigning individual and teamwork:

- Task coordinator job is to distribute tasks depending on team members' different skills.

3. Strategies for keeping on task:

- Weekly check-ups held on Discord to make sure the team is on track.

Consequences for Not Adhering to Team Contract

1.How will you handle infractions of any of the obligations of this team contract?

- Reassigning tasks to other team members with the same/similar skills.

2.What will your team do if the infractions continue?

- Hold a team meeting and if the infractions continue, reach out to the professor.

****

a)I participated in formulating the standards, roles, and procedures as stated in this contract.

b)I understand that I am obligated to abide by these terms and conditions.

c)I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.

1)	Anirudh Ambore	DATE 09/05/2023
2)	Evan Hanson	DATE 09/05/2023
3)	Wonjun Choi	DATE 09/05/2023
4)	Mishari Alharbi	DATE 09/05/2023
5)	Ziyad Alqahtani	DATE 09/05/2023
6)	Abdelrahman Mohamed	DATE 09/05/2023