

Team Biased



SkinLens

Your Personal Dermatologist

AGENDA

01

Roles and
Responsibilities
Improvements

02

Problem Statement
Project Description
Working Agreement

03

Personas
MVP
Project Design

04

Technologies
Algorithms
Diagrams

05

Project backlog

06

Sprint 1:

- Stories
- Test cases
- Completed
- Not completed

07

Sprint 1 Metrics:

- Velocity
- Burndown
Charts
- Committed Ratio

08

Sprint 1 Retrospective
Sprint 2 Planning
Project Demo

Roles and Responsibilities



MOHAMMAD ZAID

TEAM LEADER/DEVELOPER



ZHUOWEN YAN

SCRUM MASTER/DEVELOPER



NAGANJALI PUJITHA KALLI

DBA/DEVELOPER

Roles and Responsibilities



MOHAMMED TANZIL
DEVELOPER/TESTER



BANGLING YIN
DEVELOPER



ABHIKUMAR PATEL
DEVELOPER/TESTER

IMPROVEMENTS

An option to mail all the team members at once on wiki page.

In Sprint Retrospective instead of What went wrong, changed it to If we were to do it again, what would we do differently.

Added Details to project schedule.

WHAT PROBLEM ARE WE SOLVING?

Accurate and timely diagnosis of skin diseases is a challenge because:

- Limited availability of dermatologists in remote areas
- Traditional methods primarily involve only visual inspection of the skin which may be prone to human error or variability in interpretation.

What does it affect?

This results in delayed diagnosis, limited personal care and possibly worsening of skin diseases.

PROJECT DESCRIPTION

- A web-based platform to help patients diagnose skin conditions using deep learning.
- Intended for patients who wants to quickly access their skin conditions, skinLens integrates image analysis to provide accurate diagnosis of skin diseases and recommend treatments.
- Intended for dermatologists who wants to efficiently access patient's skin conditions.
- Unlike traditional visual inspections, SkinLens uses advanced image-based deep learning model and offers accurate predictions and recommend treatments along with real-time consultation with dermatologists.
- Benefits - Improved diagnose accuracy, faster and personalized care, increased accessibility, reduced costs and real-time access to consultations with dermatologists.

Team Working Agreement

Communication expectations

- All communications will be through Slack.
- Opinions and ideas of all team members will be respected.
- Changes, if any, should be discussed and agreed within the team.
- Conflicts should be focused on issues and not individuals. (Strictly avoid blame game)
- Unresolved conflicts among the team should be reported to the team leader.
- We encourage everyone to express their frustrations and discomfort early.
- Clear understanding and listening to others are the most important things to resolve conflicts.

Meeting structure and frequency

- Meetings will be held via zoom
- Daily scrum at 8pm for 15 minutes.
- A follow-up meeting should be planned after the daily scrum if there are any blockers that can't be addressed quickly.
- A meeting for Sprint Planning will be planned at the beginning of the Sprint.
- A meeting for Sprint Retrospective will be planned before the Sprint end date.
- Encourage everyone to actively participate in the meeting.
- We will adhere to agreed meeting time and will inform the team of any absences.
- Any changes in the timings will be mutually decided.
- Meeting minutes with objectives and decisions will be recorded in a Google Sheet.
- If you miss a meeting, support decisions made in your absence.
- Scrum Master should host the meeting and control time.

Team Working Agreement

Team norms and values

- Each team member understands their roles and responsibilities.
- We divide tasks transparently and help each other when necessary.
- Each task should have a clear due date and mutually accepted definition of “Done”.
- Each team member takes responsibility for their tasks and for achieving the team's goals.
- Team members should report obstacles immediately if they cannot complete their work on time, and actively contribute to the solution.
- Team members should update their task's process via Jira before the daily scrum meeting.
- Team members should upload their work to prescribed places like GitHub or Google Drive before the daily scrum meeting.

Continuous Improvement

- Encourage teams to research, learn, come up with new ideas and share with the team.
- We will reflect on areas of improvement during each Sprint Retrospective so that our process is better for the next sprint.

Definition of DONE

- Work has been fully reviewed by another team members.
- Code can run and no errors.
- All tests are successfully completed.
- No critical defects.
- Work meets the acceptance criteria.
- Code pushed to GitHub.
- Documentation has been updated.

PATIENT PERSONA

Nashley is a busy professional person, living in a city and has sensitive skin prone to rash and acne.

Challenges:

- She has skin issues and often experiences delays in scheduling appointments with dermatologists.
- Finds very difficult to describe disease accurately and fears misdiagnosis due to lack of knowledge.
- Concerned about long-term skin damage and wants fast treatment options.

Goals:

- She wants a reliable, easy-to-use platform to self-access skin issues quickly and get initial feedback without needing to wait for appointments.
- Hopes to receive personalized treatment recommendations based on her skin type, age, and medical history.



Nashley Pagioli

Age: 30

Gender: Female

Occupation: IT Employee

PARENT PERSONA

Gloria is a housewife balancing taking care of four children, household duties, and her husband, leaving her with little time to focus on individual healths.

Challenges:

- Frequently deals with minor skin issues like rashes or eczema among her kids.
- Finds to gets confused especially when it comes to skincare for different ages and skin types.
- Concerned about the cost and time required for multiple doctor visits, particularly for non-serious conditions.

Goals:

- Seeks personalized advice for each children skincare with their unique needs like skin type, allergies, or past treatments.
- Hopes to save time by using images and she uploads.
- Wants peace of mind by receiving accurate, clear advice and thinks homecare is sufficient.



Gloria Garson

Age : 40

Gender : Female

Occupation : House wife

DOCTOR PERSONA

Dr. Robert is working in both a hospital and private clinic, treating a high volume of patients with various skin conditions.

Challenges:

- Manages a large patient load daily, leading to limited time for each consultation and difficulty providing personalized care.
- Often receives incomplete descriptions of diseases, making it challenging to diagnose accurately without further testing or follow-ups.
- Struggles to track patient treatment outcomes over time, like eczema or psoriasis, due to irregular follow-ups or inconsistent data from patients.

Goals:

- Wants a platform of patient information, including images, and medical history, allowing for quick.
- Hopes to track patient progress over time, helping to adjust treatments.
- Aims to reduce the patient times.



Dr. Robert Smith

Age : 35

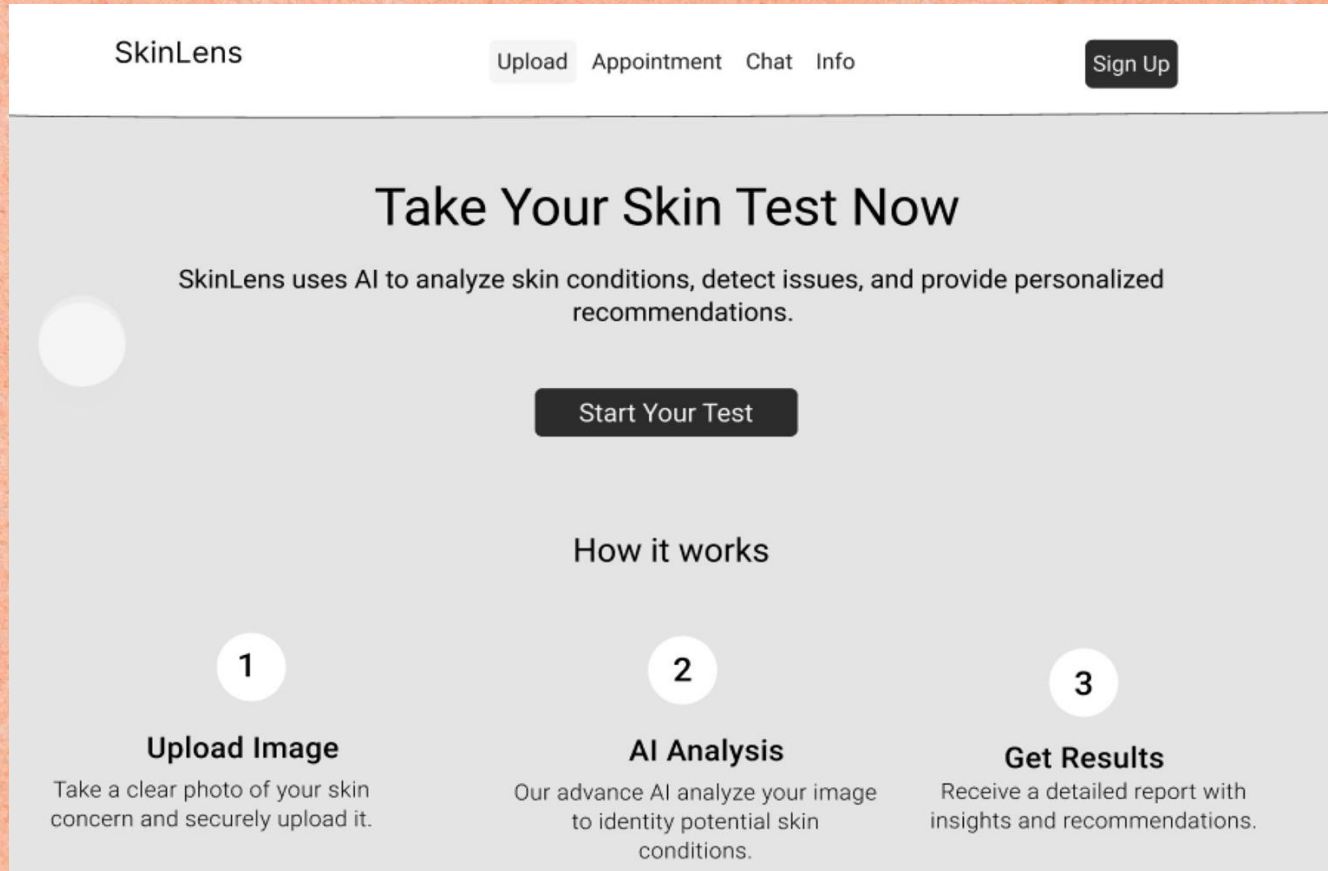
Gender : Male

Occupation : Dermatologist

MVP (Minimum Viable Product)

- Web based interface
- Patient Image Upload: Patients can upload images of their skin condition.
- AI-Powered Diagnosis: The image classification model processes the images.
- Diagnosis Display: A diagnosis result is shown to patient after images are processed.

Project Design (Figma)

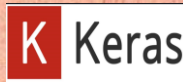


Technologies

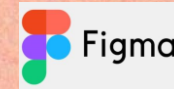
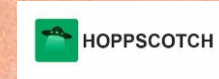
Front-end & Back-end & Database



AI Model



Tools



REACT+VITE AND TAILWIND CSS

REACT: It is an open source JavaScript library. It is used to create interactive and reactive user interfaces. We can split UIs into reusable components, which can make it easier to maintain and build large web applications.

VITE: It is a fast and good web application build tool. It stands out for its fast startup and instant compilation times during development.

TAILWIND CSS: It is an open source css framework. It encourages a more direct approach to styling.



FLASK

SkinLens utilizes a flask-backend server.

Lightweight Python Web Framework: Flask is a micro-framework that allows for easy and flexible development of web applications and APIs.

API development: Used to create RESTful API endpoints for handling requests from the frontend.

Routing: Flask handles routes that correspond to different functionalities (e.g., image upload, retrieving diagnoses).

Integration with AI Models: Flask acts as a middleware between the frontend and the ResNet50 image classification model for handling requests and returning diagnoses.

Easy Integration with Databases: Flask communicates with Firebase (Firestore and Storage) to store and retrieve data such as images and diagnosis reports.



CORS

Cross Origin Resource Sharing

Security Feature: Prevents web pages from making requests to a different domain than the one serving the web page.

Handling Cross-Origin Requests: Needed to enable requests from the frontend (React) to the backend (Flask), especially when hosted on different servers or ports.

Flask-CORS: A Flask extension used to handle CORS settings, allowing the frontend to securely access the API.

Configuration: Configured in the Flask app to allow requests from the frontend domain, specifying allowed HTTP methods (GET, POST, etc.)

Axios

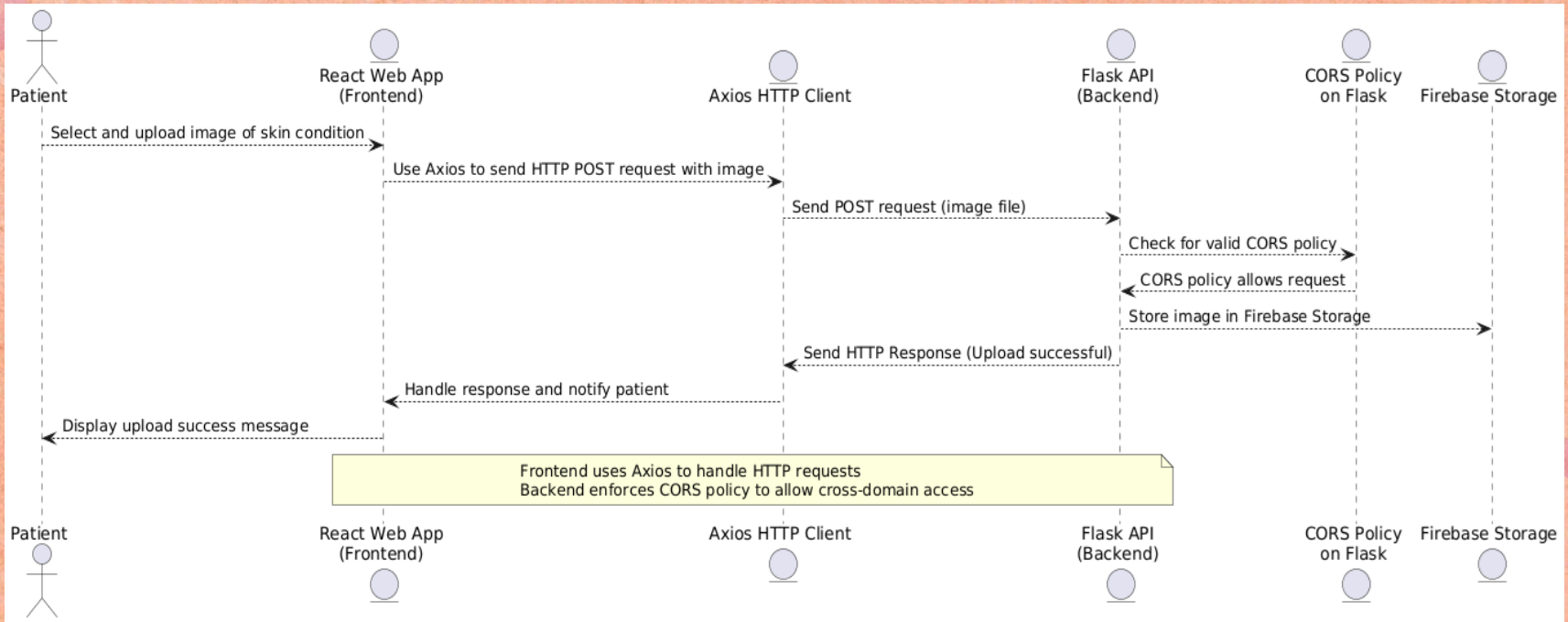
JavaScript HTTP Client: Axios is a promise-based library used to make HTTP requests from the frontend (React).

Simplifies API Calls: Handles API requests (GET, POST) and responses for operations like uploading images and retrieving diagnoses.

Error Handling: Provides built-in support for handling errors and network issues in the API requests.

Asynchronous Operations: Supports asynchronous requests, allowing the app to remain responsive while waiting for the backend to process data.

CORS and Axios functioning



ALGORITHMS:

Used TensorFlow and keras for the CNN Model and Data Augmentation.

OpenCv library for image preprocessing.

Matplotlib for visualization.

CNN Architecture:

Model Layers:

- **Convolutional Layers:** These layers extract features like edges, textures, and shapes from the images.
- **Max Pooling:** These layers reduce the spatial dimensions of the feature maps, making the model more efficient and reducing overfitting.
- **Dropout:** Randomly sets some neurons to zero during training to prevent overfitting.
- **Flatten Layer:** Converts the 3D feature maps into a 1D vector, preparing it for the fully connected layers.
- **Dense Layer:** The final output layer that classifies the images into one of the 5 classes using softmax activation.
- **Batch Normalization:** Normalizes the output from the convolutional layer to stabilize and accelerate training.
- **Softmax Activation :** The softmax function converts the raw output scores into probabilities. It ensures that the sum of the output probabilities for all classes equals 1, making it suitable for multi-class classification.

Key Parameters:

- **Optimizer:** Adam.
- **Loss Function:** Categorical Cross-entropy (for multi-class classification).

Project Schedule

Sprint 0

- Sep 6 - Sep 26
- Daily Scrum
- Pre Planning

Sprint 1

- Sep 26 - Oct 24
- Daily Scrum, Planning, Retrospective
- Image Upload Feature
- Potential Diagnosis as output.

Sprint 2

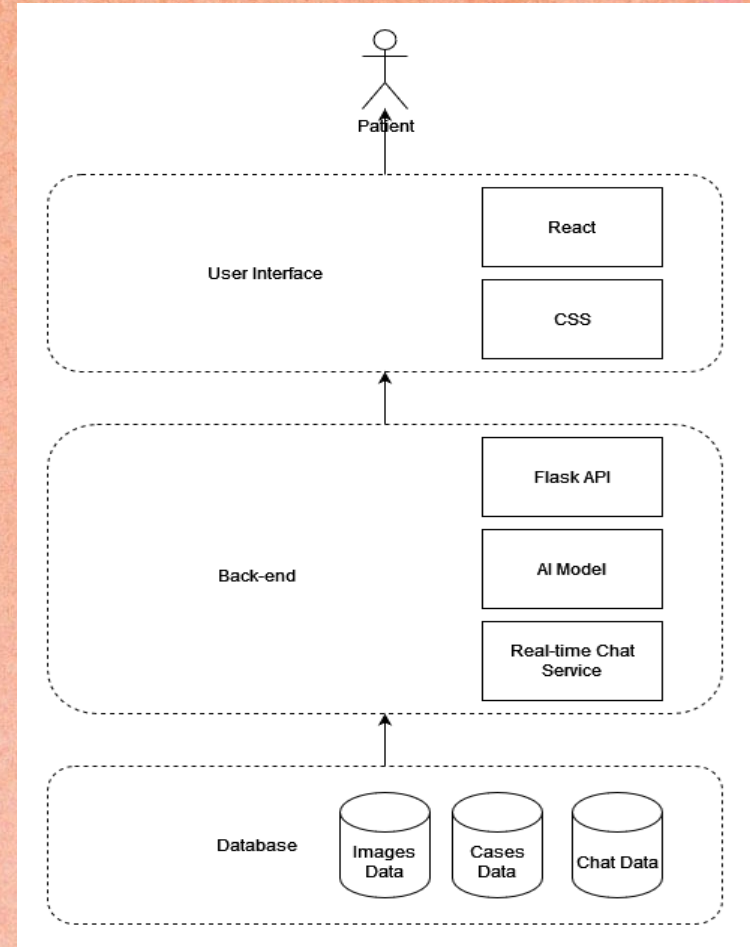
- Oct 24 - Nov 21
- Daily Scrum, Planning, Retrospective
- More Accuracy
- Treatment Recommendations

Sprint 3

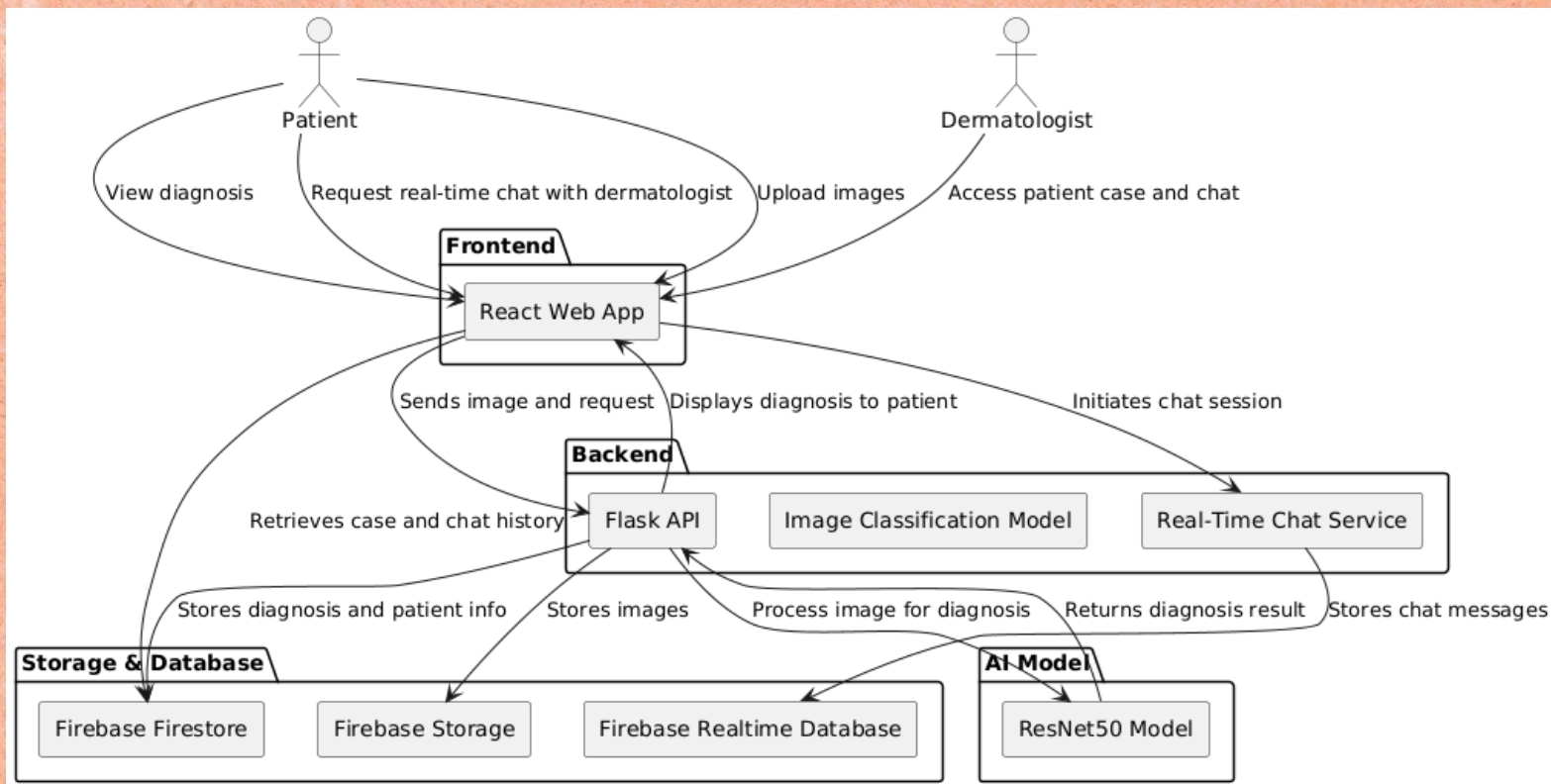
- Nov 21 - Dec 12
- Daily Scrum, Planning, Retrospective
- Real-Time Chat

Diagrams

Architecture Diagram



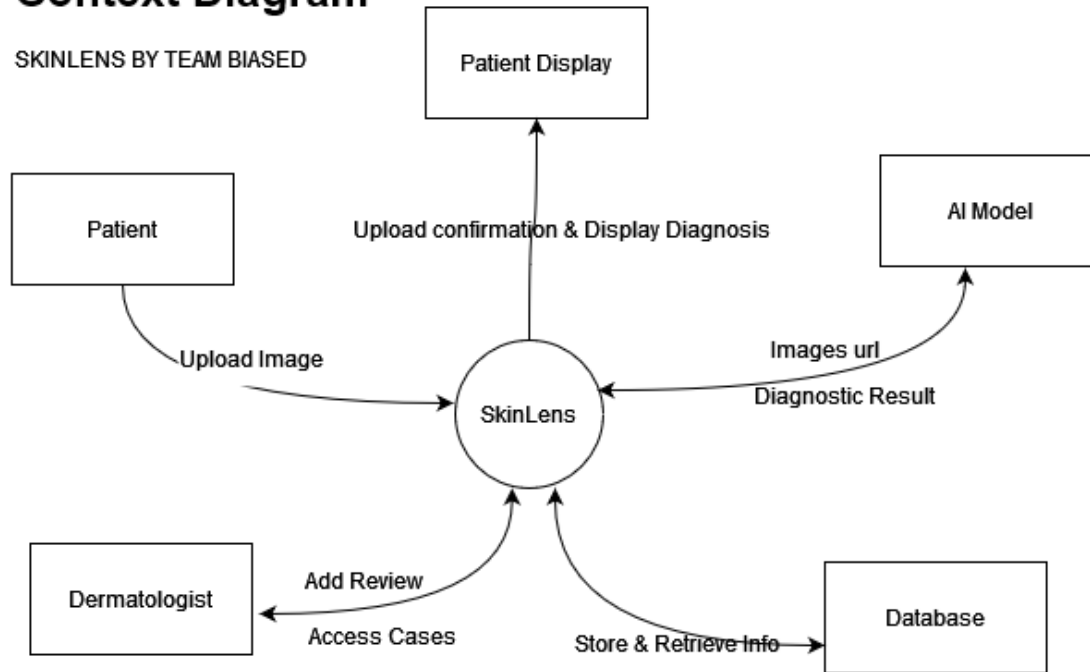
Architecture Diagram



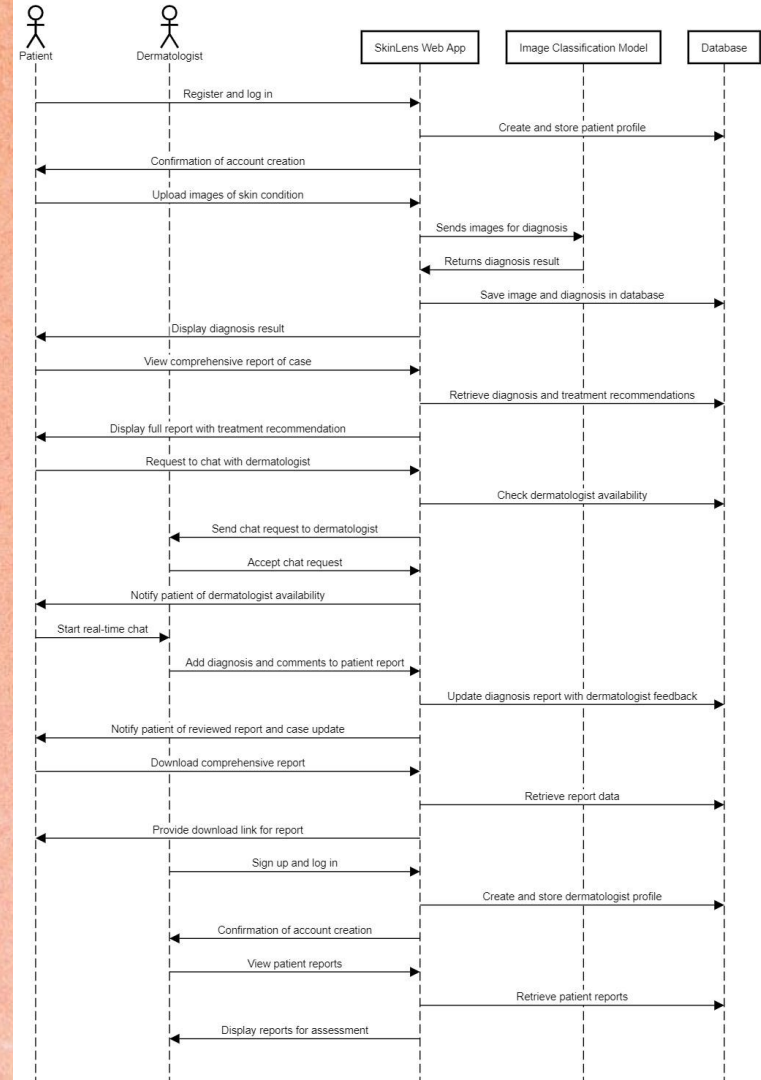
Context Diagram

Context Diagram

SKINLENS BY TEAM BIASED



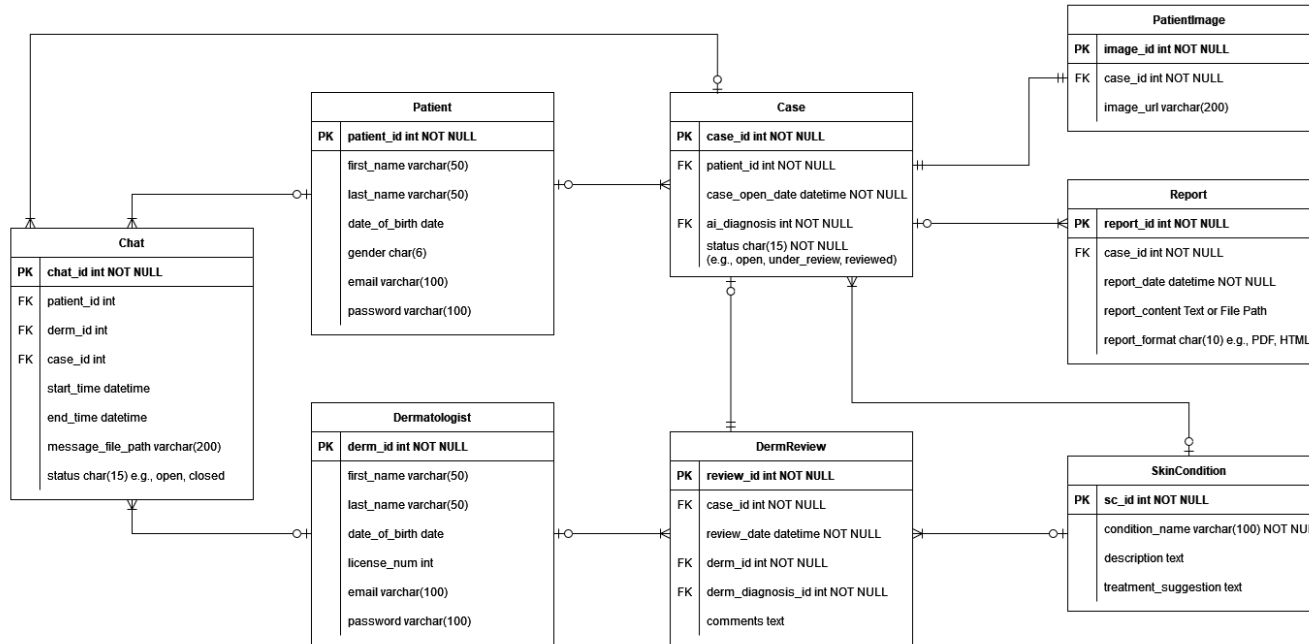
Sequence Diagram



ER Diagram

Entity Relationship Diagram (ERD)

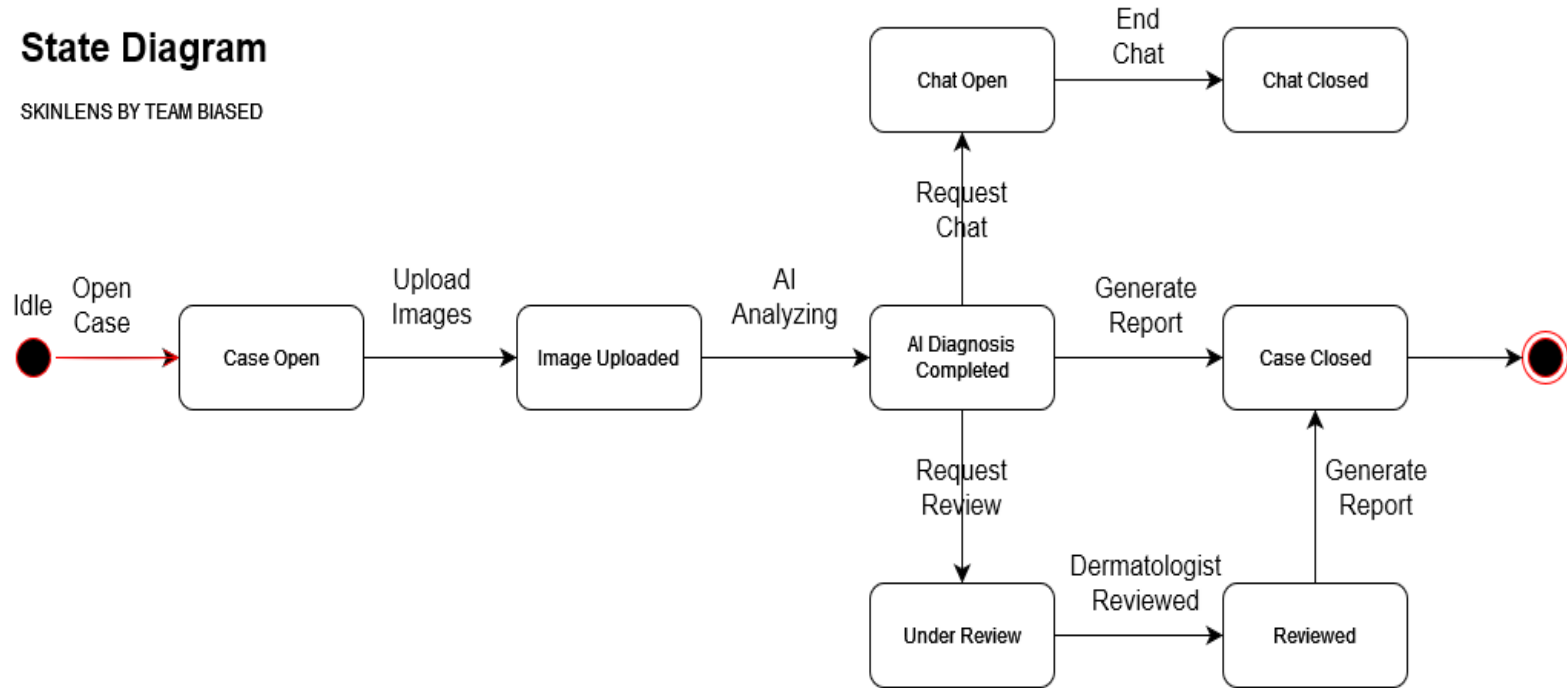
SKINLENS BY TEAM BIASED



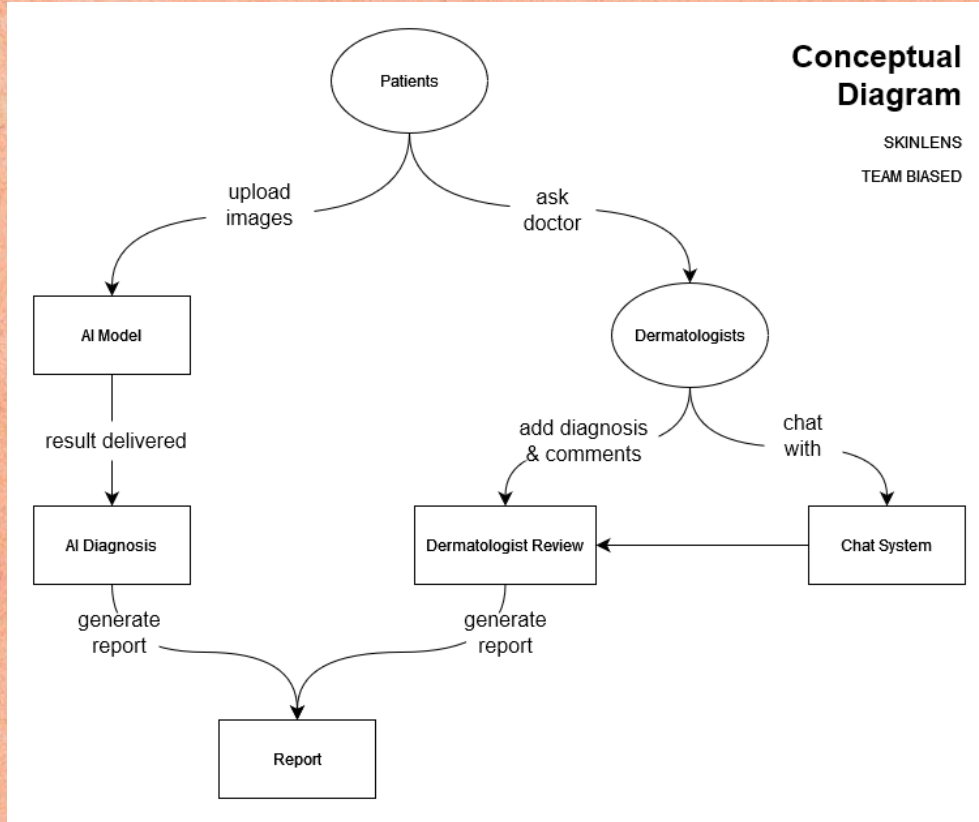
State Diagram

State Diagram

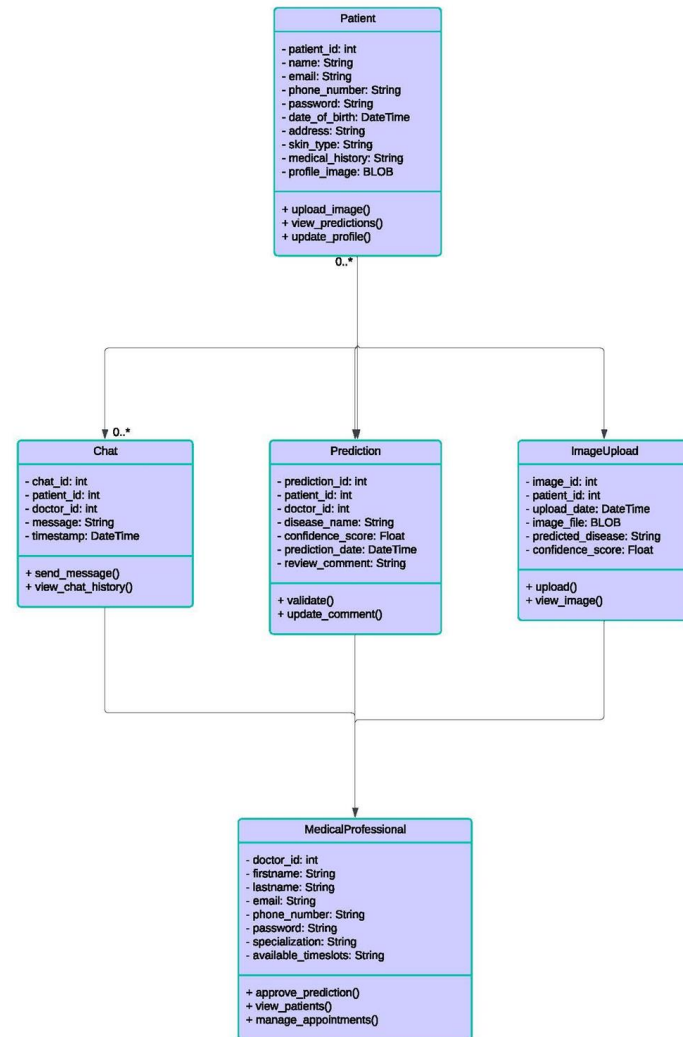
SKINLENS BY TEAM BIASED



Concept Diagram



Class Diagram



PRODUCT BACKLOG

No.	User Stories	Acceptance Criteria	Feature	Story Points
US_01	As a patient, I want to upload an image of my skin condition so that I can receive a diagnosis.	The patient can access an interface to upload an image. The patient can select and submit an image from device. The system accepts image files in specified formats, e.g., JPEG, JPG, PNG.	Image Upload	13
US_02	As a patient, I want to upload several images to one case so that I can provide more information about my skin condition.	The patient can upload multiple images (up to 3) to one case. The system accepts multiple images in specified formats (JPEG, JPG, PNG). All uploaded images are associated with the same patient case. The patient receives a confirmation message after uploading each image. The system displays a preview of all uploaded images for review before diagnosis.	AI Diagnosis	2
US_03	As a patient, I want to receive a diagnosis so that I know the skin issue I am facing	After submitting the image, the patient receives a diagnosis. The diagnosis is presented as the name of the skin condition. The system provides the diagnosis within a reasonable time, e.g., less than 1 min.	AI Diagnosis	20
US_04	As a patient, I want to receive the most accurate possible diagnosis so that I can trust the diagnosis provided.	The system continuously improves the model and algorithms for better diagnostic results. The diagnosis accuracy is optimized based on model refinements and external validation. The confidence level of the diagnosis should be high (e.g., 95%).	AI Diagnosis	11

US_05	As a patient, I want to receive a brief description of the diagnosed condition so that I can understand the skin issue I am facing.	After receiving the diagnosis, the patient is provided with a brief description of the diagnosed condition. The description is written in clear, understandable language. The description includes common symptoms and causes.	AI Diagnosis	2
US_06	As a patient, I want to receive treatment recommendations so that I can take action towards treating my skin condition.	After receiving the diagnosis, the patient is provided with a brief description of the diagnosed condition. The description is written in clear, understandable language. The description includes common symptoms and causes of the condition.	AI Diagnosis	3
US_07	As a patient, I want to view a comprehensive report of a case so that I have a complete record of my skin condition assessment.	Patients can easily create and access the report from their diagnosis page. The layout is user-friendly with clear sections and headings, and all images are displayed clearly within the report. Report contents: - Patient information, - Case details(case open date, unique case identifier, images uploaded by patient, AI diagnosis of skin condition, brief description of the diagnosed condition, treatment suggestion), - Report Metadata(report generate date, unique report identifier). - Derm's input(derm's diagnosis, derm's comments and recommendations, derm's review date, derm's name). - Disclaimers clarifying that any limitations of the AI diagnosis. If the dermatologist's review is pending, the report indicates which sections will be updated after the review. If the report generation fails, the system provides an error message and guide user to next steps.	Report	2
US_08	As a patient, I want to download a comprehensive report of a case, so that I can keep the document of my skin condition assessment.	An option to download the report is available. The report provides a downloadable format, e.g., PDF. The downloaded document displays the same layout as the web application.	Report	1

US_09	As a patient, I want to view my past cases and track their statuses, so that I can monitor my health progress and stay informed about the evaluations.	<p>The patient can access "Case History" section after logging in their account. All past cases are listed in a chronological order, top is the most recent cases, each case displays a summary information: case id, case open date, status.</p> <p>The system updates case statuses in real-time, without requiring the patient to refresh manually.</p> <p>Clicking on a case entry navigate the patient to that case's report.</p>	User Account Management	2
US_10	As a dermatologists, I want to access and view patient reports so that I can assess their conditions and provide professional feedback.	<p>The dermatologist can log in to the system using valid credentials. Upon logging in, they are directed to a dashboard displaying all assigned patient reports.</p> <p>Reports are presented in a list view.</p> <p>The dermatologist can sort reports by date, status, or patient ID.</p>	User Account Management	2
US_11	As a dermatologist, I want to add my diagnosis to patient reports so that I can contribute my professional assessment to the patient's case.	<p>The dermatologist can open a patient report and select their diagnosis from the skin conditions list.</p> <p>The dermatologist can submit the diagnosis, the status of the report changed to "review" after the submission.</p> <p>The dermatologist diagnosis is associated with patient's report, and visible to the patient in their report.</p>	Derm Review	1
US_12	As a dermatologist, I want to add comments to patient reports so that I can provide additional guidance if necessary.	<p>The dermatologist can open a patient report and there is a section to add comments.</p> <p>The field supports text input with a reasonable character limit (e.g., 500 characters).</p> <p>The dermatologist can submit the comment.</p> <p>The dermatologist comment is associated with patient's report, and visible to the patient in their report.</p>	Derm Review	2
US_13	As a patient, I want to be notified as soon as my report is reviewed by a dermatologist so that I can stay informed about my case status.	<p>The patient receives a notification immediately when the dermatologist submits their diagnosis and comments.</p> <p>Notifications can be delivered via: Email, In-app notification.</p> <p>The notification includes:</p> <p>A brief message indicating that the report has been reviewed.</p> <p>Instructions on how to view the updated report.</p>	Derm Review	2

US_14	As a patient, I want to chat with a dermatologist in real-time so that I can receive consultation and clarify my concerns.	<p>The patient can access a chat feature from report page.</p> <p>Patients can request a chat session with a dermatologist.</p> <p>The chat interface supports: Sending and receiving text messages, typing indicators, time stamps for each message, sending attachments (e.g., additional images, documents) if permitted.</p> <p>Only the patient and the assigned dermatologist can access the chat history.</p>	Chat	10
US_15	As a dermatologist, I want to chat with patients in real-time so that I can provide personalized consultation and answer their questions.	<p>The dermatologist has access to a chat interface showing active chat requests from patients.</p> <p>Dermatologists can accept or decline chat requests, upon acceptance, the real-time chat session begins.</p> <p>The chat supports: Sending and receiving text messages, typing indicators, time stamps for each message, sending attachments (e.g., additional images, documents) if permitted.</p> <p>The dermatologist can end the chat session when appropriate.</p>	Chat	5
US_16	As a patient, I want to register an account so that I can store my basic information and access personalized services.	<p>The patient can access a registration page requiring: Full name, Email address, Password creation, Date of birth.</p> <p>Post-registration, the patient can log in and update their profile information.</p>	User Account Management	2
US_17	As a dermatologist, I want to sign up so that I can access patient reports and provide professional feedback.	The dermatologist can access a sign-up page requiring: Full name, Email address, Password creation, Medical license number.	User Account Management	2
US_18	As a patient, I want to log in to my account so that I can manage my cases and view diagnoses.	The patient accesses a login page requiring: Email address, Password. Upon successful login, the dermatologist is directed to their dashboard.	User Account Management	1
US_19	As a dermatologist, I want to log in to my account so that I can manage patient reports and update case statuses.	The dermatologist accesses a login page requiring: Email address, Password. Upon successful login, the dermatologist is directed to their dashboard.	User Account Management	1

SPRINT 1 STORIES

US_ID	User Stories	Acceptance Criteria	Feature	Story Points
US_01	As a patient, I want to upload an image of my skin condition so that I can receive a diagnosis.	<p>The patient can access an interface to upload an image.</p> <p>The patient can select and submit an image from device.</p> <p>The system accepts image files in specified formats, e.g., JPEG, JPG, PNG.</p> <p>The uploaded image is stored and associated with the patient's case.</p> <p>A confirmation message is displayed upon successful upload.</p>	AI Diagnosis	13
US_03	As a patient, I want to receive a diagnosis so that I know the skin issue I am facing	<p>After submitting the image, the patient receives a diagnosis.</p> <p>The diagnosis is presented as the name of the skin condition.</p> <p>The system provides the diagnosis within a reasonable time, e.g., less than 1 min.</p>	AI Diagnosis	20

SPRINT 1 TEST CASES

Test Cases for US_01: As a patient, I want to upload an image of my skin condition so that I can receive a diagnosis.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_01_01	Access Image Upload Interface	The patient is logged into the system or visit as a guest. 1. Navigate to the application's "Home" page. 2. Click on the "Take Your Test" option.	The image submission page is displayed, good layout, and allowing the patient to select images to upload.	Passed	Oct 16	Tanzil
TC_01_02	Upload Supported Image Format JPEG	1. Click on the "Select Image" button. 2. Select the JPEG image from the device. 3. Click "Analyze Image" to confirm the selection.	The system uploads the image successfully. The image is stored and associated with the patient's new case. A preview of the uploaded image is displayed	Passed	Oct 16	Tanzil

TC_01_03	Upload Supported Image Format PNG	<ol style="list-style-type: none"> 1. Click on the "Select Image" button. 2. Select the PNG image from the device. 3. Click "Analyze Image" to confirm the selection. 	<p>The system uploads the image successfully.</p> <p>The image is stored and associated with the patient's new case.</p> <p>A preview of the uploaded image is displayed.</p>	Passed	Oct 21	Tanzil
TC_01_04	Attempt to Upload Unsupported Image Format GIF	<ol style="list-style-type: none"> 1. Click on the "Select Image" button. 2. Select the GIF image from the device. 3. Click "Analyze Image" to confirm the selection. 	<p>The system displays an error message: "Unsupported file format. Please upload a JPEG or PNG image."</p> <p>The image is not uploaded or stored.</p> <p>The patient remains on the image upload interface.</p>	Passed	Oct 21	Tanzil
TC_01_05	Attempt to Upload Without Selecting a File	<ol style="list-style-type: none"> 1. Click on the "Select Image" button. 2. Click on the "Analyze Image" button without selecting an image. 	<p>The system displays a warning message: "Please select an image to upload."</p> <p>No action is taken, and the patient remains on the image upload interface.</p>	Defect	Oct 21	Tanzil

Test Cases for US_ 03: As a patient, I want to receive a diagnosis so that I know the skin issue I am facing

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_03_01	Successful Diagnosis Delivery by Image Uploaded	The patient has uploaded images. 1. On the case submission page, click the "Analyze Image" button to send the case for diagnosis.	Processing begins immediately after submission. Within less than 1 minute, the system displays the diagnosis on the screen. The diagnosis is presented clearly as the name of the skin condition.	Passed	Oct 21	Abhi
TC_03_02	Diagnosis Presentation Format	Same as TC_03_01	The diagnosis is displayed prominently, not hidden or obscured by other interface elements. The name of the skin condition is clear and understandable (e.g., "Eczema"). No technical jargon or codes are used.	Passed	Oct 21	Abhi

TC_03_03	Timely Diagnosis Delivery	Same as TC_03_01	<p>The diagnosis is displayed within 60 seconds.</p> <p>The system may show a indicator while processing.</p>	Passed	Oct 21	Abhi
TC_03_04	System Behavior When Diagnosis Cannot Be Determined	Same as TC_03_01	<p>The system informs the patient that a diagnosis could not be determined.</p> <p>A message is displayed: "We are unable to determine a diagnosis based on the images provided." Suggestions are offered, such as "Please upload clearer images" or "Consider consulting a dermatologist."</p>	Defect	Oct 21	Abhi

SPRINT 1 STORIES COMPLETED AND NOT

US_ID	User Stories	Completed Date	Story Points	Completed
US_01	As a patient, I want to upload an image of my skin condition so that I can receive a diagnosis.		13	12
	Set up a React + VITE front-end server	Oct 9	1	1
	Set up Flask back-end server	Oct 9	1	1
	Finalize a Figma design for S1	Oct 13	3	3
	Create React front-end for homepage	Oct 12	2	2
	Create React front-end for image-upload page	Oct 15	3	3
	Integrate the front-end and back-end for image input	Oct 18	3	2
US_03	As a patient, I want to receive a diagnosis so that I know the skin issue I am facing		20	15
	Query dataset and format labels	Oct 12	2	2
	Preprocess images	Oct 12	2	2
	Develop the training file and run on colab	Oct 19	6	6
	Deploy the model to cloud server	Oct 23	5	1
	Evaluate and inference of the image classification model	Oct 23	2	1
	Integrate the image classification model and back-end	Oct 21	3	3

METRICS – TEAM VELOCITY

Sprint 1 Velocity

Total Story Points: 27

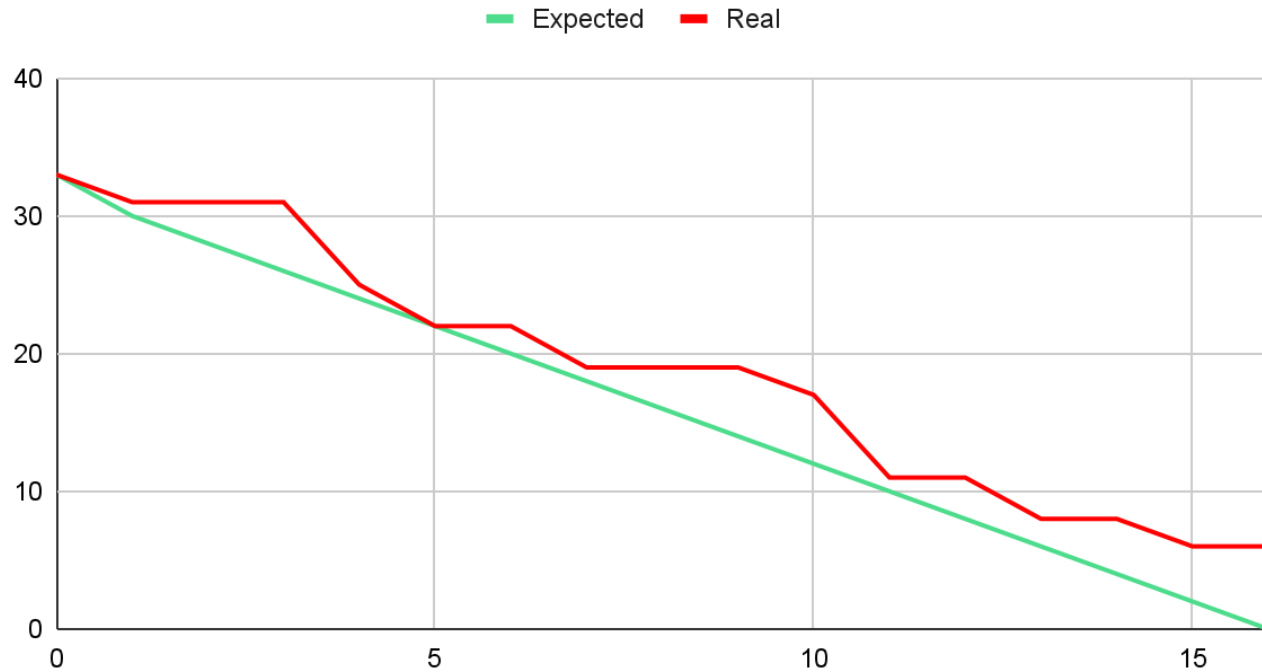
Completed User Stories:

- Set up a React + VITE front-end server - 1 point
- Set up Flask back-end server - 1point
- Finalize a Figma design for S1 - 3 points
- Create React front-end for homepage - 2 points
- Create React front-end for image-upload page - 3 points
- Query dataset and format labels - 2 points
- Preprocess images - 2 points
- Develop the training file and run on colab - 6 points
- Evaluate and inference of the image classification model - 1 point

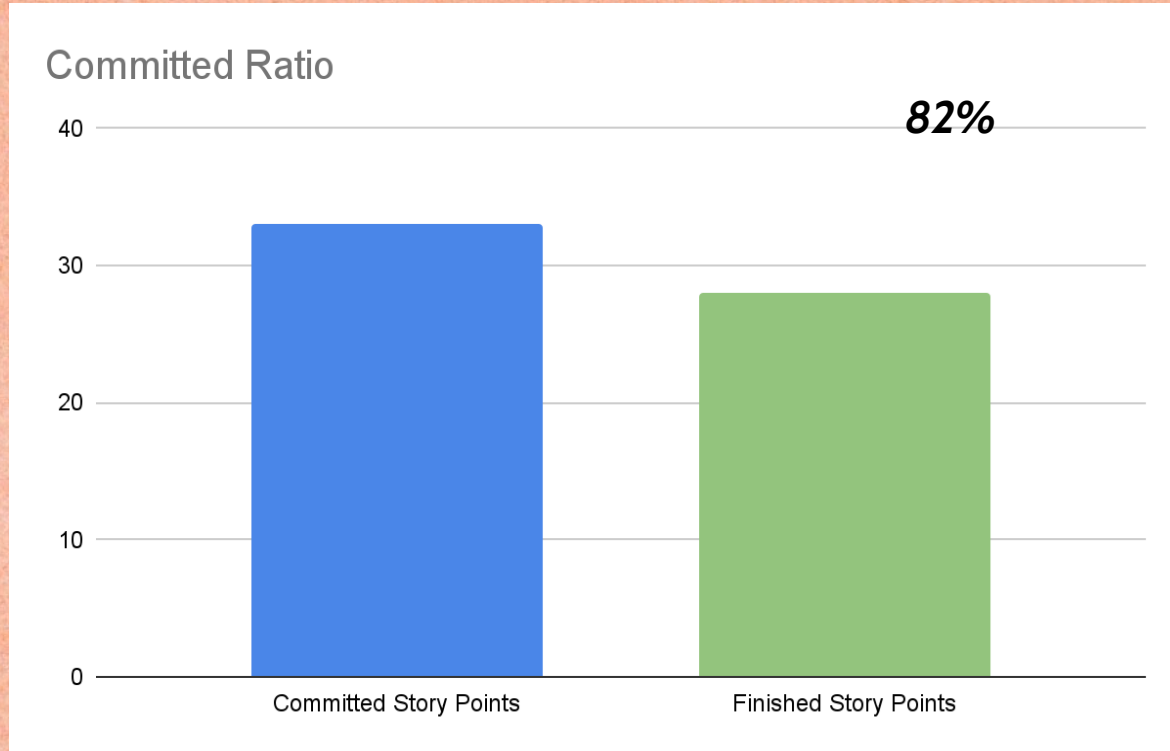
Velocity: 27 story points completed.

METRICS - BURNDOWN CHART

Burndown Chart for Sprint 1



METRICS - COMMITTED RATIO



Sprint 1 Retrospective

Few important points from the Retrospective

What did we do right?

- ❖ scrum meetings are going well
- ❖ If anyone missed the meeting, they provided a valid reason prior to the meetings.
- ❖ Individual research done by team members were quite resourceful and provided deep insights on the parts they were doing
- ❖ everyone were willingly helping each other on resolving the issues we were facing

Few important points from the Retrospective

If we were to do it again, what would we do differently?

- ❖ break down the tasks into more smaller and detailed sub-tasks
- ❖ Focus was more towards the entire project on how we are going to do it, that consumed a lot of time leading to less time for the user stories in this sprint
- ❖ Rather than spending much time researching and then begin to code, start developing the code parallel.
- ❖ cost a lot of time on persuading others and unity the opinion, we would make quick decisions rather than spending more time on it
- ❖ Setting deadlines a day or two prior to the actual course deadlines so we have enough time to resolve the issues we encounter at last

Few important points from the Retrospective

Action items

- ❖ Assign to Zaid: break down the tasks into more more smaller and detailed sub-tasks ----- Assign to Everyone: update the tasks during developing
- ❖ Focus was more towards the entire project on how we are going to do it, that consumed a lot of time leading to less time for the user stories in this sprint.
Assigned to Pujitha: Keep discussions and focus on user stories of current sprint
- ❖ Assign to everyone: Rather than spending much time researching and then begin to code, start developing the code parallel.
- ❖ cost a lot of time on persuading others and unity the opinion, we would make quick decisions rather than spending more time on it. —>Action Item: Assigned to Tanzil (Voting on disagreement)
- ❖ Assigned to Bangling: Setting deadlines a day or two prior to the actual course deadlines so we have enough time to resolve the issues we encounter at last moment and Assigned to Zhuowen: Make a checklist

SPRINT 1 RETROSPECTIVE

IdeaBoardz

start typing to filter stickies

Export

Login

View Section All Sections

Sort By created time

Sprint 1 Retrospective

What did we do right? +

scrum meetings are going well + 5	Individual research done by team members were quite resourceful and provided deep insights on the parts they were doing + 3	The action items from sprint zero retrospective like, meetings with proper agenda, using zoom hand raise and tech familiarity were applied. + 5
good supporting team + 0	Everyone is brave to go out their comfort zone + 0	Team Leader and Scrum master were responsible + 0
If anyone missed the meeting, they provided a valid reason prior to the meetings. + 4	everyone were willingly helping each other on resolving the issues we were facing + 3	good communication with each other + 2
have a good Jira, and keep update it + 0	good and detailed user stories + 1	meetings went well according to the agenda. + 0
everyone are committed towards assigned work and deadlines. + 0	team is very focused about their work + 0	Everyone adhered to the team working agreement + 1
Knowledge sharing among team members enriched the project + 0	everyone gave their points about project + 1	

If we were to do it again, what would we do differently? +

we would focus more time on techniques + 1	We would explore more tools + 1	If someone disagree, come up with an alternative that will add value rather than just disagreeing the points made + 1
Address issues immediately as they arise + 0	try more and achieve more possibilities + 0	Few members took a day or two more to complete their part which lead to updating the entire sprint schedule. We can set strict deadlines. + 1
we would begin testing earlier in the process to catch issues sooner rather than waiting until the end + 0	sometimes teams feel anxious but don't complain without actions + 1	we would do team work more efficient, 1-1 should >= 2 + 1

Action items +

Assign to everyone: Rather than spending much time researching and then begin to code, start developing the code parallel. + 4	Assigned to Bangling: Setting deadlines a day or two prior to the actual course deadlines so we have enough time to resolve the issues we encounter at last + 2 Make a checklist	Assign to Zaid: break down the tasks into more smaller and detailed sub-tasks ----- Assign to Everyone: update the tasks during developing + 6
cost a lot of time on persuading others and unity the opinion, we would make quick decisions rather than spending more time on it. ----- Action Item: Assigned to Tanzil (Voting on disagreement) + 3	Focus was more towards the entire project on how we are going to do it, that consumed a lot of time leading to less time for the userstories in this sprint. Assigned to Pujitha: Keep discussions and focus on user	

SPRINT 2 PLANNING

TOTAL: 27 USER STORIES

No.	User Stories	Feature	Story Points
US_02	As a patient, I want to upload several images to one case so that I can provide more information about my skin condition.	AI Diagnosis	2
US_04	As a patient, I want to receive the most accurate possible diagnosis so that I can trust the diagnosis provided.	AI Diagnosis	11
US_05	As a patient, I want to receive a brief description of the diagnosed condition so that I can understand the skin issue I am facing.	AI Diagnosis	2
US_06	As a patient, I want to receive treatment recommendations so that I can take action towards treating my skin condition.	AI Diagnosis	2
US_07	As a patient, I want to view a comprehensive report of a case so that I have a complete record of my skin condition assessment.	Report	3
US_08	As a patient, I want to download a comprehensive report of a case, so that I can keep the document of my skin condition assessment.	Report	1

User Stories Carry Over from Sprint 1

US_ID	User Stories		Story Points
US_01	As a patient, I want to upload an image of my skin condition so that I can receive a diagnosis.		1
	DEFECT: Integrate the front-end and back-end for image input		1
US_03	As a patient, I want to receive a diagnosis so that I know the skin issue I am facing		5
	Deploy the model to cloud server		1
	Evaluate and inference of the image classification model and integration		4

PROJECT DEMO – SCREENSHOT

Upload Your Skin Image

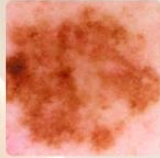


Image selected

or

Select Image

skindisease.jpg

Analyze Image

Prediction Result:
Melanoma

Upload Instructions

- ✓ Ensure the image is clear and well-lit
- ✓ Focus on the specific skin area you want analyzed
- ✓ Remove any accessories or clothing that might obstruct the view
- ✓ Accepted file formats: JPEG, PNG
- ✓ Maximum file size: 10MB

Once you upload an image, our AI will analyze it and provide insights about your skin condition. This process is completely secure and your privacy is our top priority.

PROJECT – APIs

Project APIs

- **Endpoint:** `/predict` (POST)
 - **Description:** Accepts an image file from the frontend, processes it through the machine learning model, and returns a diagnosis.
 - **Request:** `POST`, (Image upload)
 - **Response:** JSON with the prediction result (e.g., skin condition name).
- **Axios for API Requests:**
 - Used in the frontend to send HTTP requests to the backend.
 - Handles image upload, async operations, and error handling.
- **CORS (Cross-Origin Resource Sharing):**
 - Enabled in Flask to allow secure requests from the React frontend, ensuring safe cross-domain interactions.
- **Model Prediction:**
 - The Flask backend uses a model to analyze the uploaded images and return the skin condition diagnosis.

PROJECT – APIs

Flask API and CORS

```
# Initialize Flask app
app = Flask(__name__)
CORS(app) # Allow cross-origin requests from React frontend

# Load pre-trained model
model = keras.models.load_model("models/skinlensAI.keras")

# Define route for prediction
@app.route('/predict', methods=['POST'])
def predict():
    if 'image' not in request.files:
        return jsonify({'error': 'No image uploaded'}), 400

    # Get the uploaded image
    f = request.files['image']

    # Save the file temporarily
    basepath = os.path.dirname(__file__)
    filepath = os.path.join(basepath, 'uploads', f.filename)
    f.save(filepath)
```

Axios

```
const handleSubmit = async () => {
    if (!file) {
        alert("Please upload an image first.");
        return;
    }

    // Create FormData object and append the image file
    const formData = new FormData();
    formData.append("image", file);

    try {
        // Send POST request to Flask backend
        const response = await axios.post("http://localhost:5000/predict", formData, {
            headers: {
                "Content-Type": "multipart/form-data",
            },
        });

        // Set the prediction result to state
        setPrediction(response.data.prediction);
    } catch (error) {
        console.error("Error uploading file:", error);
    }
};
```


WIKI PAGE LINK

To see our progress

<https://github.com/htmhw/2024F-Biased/wiki>





DEMO

THANK YOU!