

Depression detection app



Subject: CS691
Prof. Henry Wong



Agenda

01



Project Description and problem statement

02



Description of minimal viable product and all attributes, including product backlog

03



Acceptance Criteria & user stories

04



Sprint 4 backlog and test cases

05



Team metrics for every sprint & plans for sprint 5

06



Application screenshot & ML model description

TEAM Members



Omkar Shitole
Developer & DBA



Yuxiang Liu
Developer



Shivani Chavan
Developer



Wangbo Gu
Developer



Artem Kolmogorov
Developer & Project Manager



Siddarth Ravirala
Developer

Improvements made from professor's feedback

Order of presentation

We have rearranged our presentation to better suit presentation checklist

Retrospective

We have restructured our retrospective to reflect three main question

Correction of Sprint backlog

Sprint backlog now reflects the stories the team committed to working on

Project Description



Our detection system will be trained by a unique dataset for depression faces from messages and images. We believe that certain linguistic traits can be examined and linked to possible depressive symptoms as well as used to forecast self-destructive behavior. The training result we want is that the system can analyze the input (messages, images, and video) from users, and detect the type of depression such as anxiety, bipolar, or paranoia. Finally, the possibility of different depressions that the person might have will be provided.

Problem Statement

Depression is a prevalent mental health disorder that affects millions of people worldwide. The condition is characterized by a persistent feeling of sadness, hopelessness, and loss of interest in daily activities. It significantly impairs a person's ability to function and can lead to a range of physical and emotional problems. Early detection of depression is crucial for effective treatment, as it can reduce the severity and duration of symptoms and improve a person's quality of life.

However, traditional methods of detecting depression, such as self-report questionnaires or clinical interviews, have several limitations. Self-report questionnaires rely on a person's ability to accurately report their symptoms, which can be influenced by various factors, such as social desirability bias, memory recall bias, and language barriers. Clinical interviews can also be subjective and may vary depending on the clinician's experience and training.

As a result, there is a need for an objective and reliable method of detecting depression. One potential solution is the use of machine learning algorithms to analyze various data sources, such as speech patterns, facial expressions, and physiological signals, to identify indicators of depression. These systems have shown promising results in early studies and have the potential to revolutionize the field of mental health.

Detecting depression early is essential, as the condition can lead to a range of negative outcomes, including increased risk of premature death, suicidal thoughts, and impaired daily functioning. By developing accurate and reliable detection systems, we can improve the lives of millions of people who suffer from depression worldwide.

Teamwork agreement

CS-691

Team Agreement

Communication

- The team will communicate with each other through a variety of channels. For weekly meetings for meaningful team discussions, zoom meetings will be used. All the team members are highly encouraged to keep their cameras on, which will be able to build trust between the team members and reflect transparency;
- To discussion regarding minute details and doubts or anything urgent, a WhatsApp messenger group will be used.
- To share the final deliverables, Google docs will be used where all the team members can edit the document.
- A common platform called Trello has been set up for all team members, where designated groups have been created, such as Developers, Business Analyst, Product Owner.
- Database management, bugs, attendance, weekly plan, and meeting minutes. This manages all the bits and pieces of the project and makes the project management efficient.

Work division and Participation

TEAM AGREEMENT

CS-691

1

TEAM AGREEMENT

CS-691

2

TEAM AGREEMENT

CS-691

3

- The entire project work should be divided into equal parts, and equal responsibilities should be given to all the team members.
- Each team member should complete their division of work before the deadline. If they are unable to complete the work on time, that hinders the performance of the entire team. If in case a team member is facing trouble and issues at some point, they can share it with others so that they can help each other and complete the work before the deadline.
- All the team members are expected to attend the meetings promptly.
- Absence during multiple meetings will affect the team's performance and efficiency. The team member can discuss beforehand with the team leader if he/she is going to miss the meeting or make it up for it before the next meeting is scheduled.
- Work is separated between members of the group separated voluntary, however if members lacks participation product owner is entitled to assign necessary tasks to absentee members.
- In case member is absent during meetings, member pledges to support whichever decision is approved during that meeting.

Meetings

- All the team members will meet on zoom virtually every Tuesday and Friday. All the team members have to be present, as attendance is mandatory unless there is an exceptional case.
- The team leader would be responsible for sending meeting details and conducting the meeting.
- A meeting track or meeting minutes report would be listed after every meeting to keep track of the project and its progress.
- Every team member is expected to come up with ideas, participate in the discussion, and give an update on their progress for their part of the work.

Respect

- Making sure all team members always have chance to share their opinion

*All members agree to respect each others personal time and try not bother members during night time unless it is urgently required by the project.

Team Member	Email
Shivani Chavan	shivani.chavan@pace.edu
Yuniang Liu	yliu6417@pace.edu
Omkar Shinde	os33654n@pace.edu
Wangbo Gu	wgu10154n@pace.edu
Siddharth Ravula	srid4139n@pace.edu
Artem Kolmogorov	ak71778n@pace.edu

Personas

Jack

Profile



Jack, a 35-year-old software engineer who has a history of depression and anxiety. He often finds himself feeling overwhelmed at work and is struggling to balance his job and personal life. He wants to find a way to manage his symptoms so that he can be more productive and happier.

Name: Jack

Age: 35

Location: Chicago, MI

Job: Software Engineer

Salary: 92 000 – 110 000\$/annually

Family: Single

Interests

- Riding Bike
- Goes Kayaking
- Attending live concerts

Frustration

- Managing depression
- Wants to change job, but unable

Goals

- Moving to south
- Finding friends
- Buying a bigger house

Personas

Sarah



Profile

Sarah, a 25-year-old graphic designer who has been feeling down for the past few months. Despite her successful career and supportive friends and family, she feels unfulfilled and struggles to find joy in her everyday life. She wants to find a way to manage her depression and is open to using technology to help her do so.

Name: Sarah

Age: 25

Location: Los Angeles, CA

Job: Graphic Designer

Salary: 66 000 – 70 000\$/annually

Family: Single

Interests

- Writing own comic books
- Everyday swimmer
- Loves animals
- Watching series

Frustration

- Traffic in Los Angeles area
- Living alone for a long time

Goals

- Buying her own house
- Building lifelong relationships
- Finding friends

Personas

Lisa

Profile



Lisa, a 40-year-old stay-at-home mom who is feeling overwhelmed and exhausted. She is struggling to keep up with the demands of taking care of her children and household, and she often feels like she is failing as a mother. She wants to find a way to manage her stress and feelings of inadequacy.

Name: Lisa

Age: 40

Location: New-York, NY

Job: Unemployed

Salary: N/A

Family: Married, two kids

Interests

- Reading romantic novels
- Going for beach holidays
- Loves animals
- Watching series

Frustration

- Her husband going for a long business trip
- Having troubles with her kids

Goals

- Taking dog from shelter
- Helping kids with college

Personas

Tom

Profile



Tom, a 50-year-old small business owner who is feeling stressed and burnt out. Despite the success of his business, he is feeling overwhelmed by the responsibilities and pressure of running it. He wants to find a way to manage his stress and anxiety so that he can enjoy his success and have a better work-life balance.

Name: Tom

Age: 50

Location: Austin, TA

Job: Self-Employed

Salary: 150 000 \$/annually

Family: Divorced

Interests

- Going hunting & fishing
- Taking rides in his chopper

Frustration

- Declining business profits
- Having health problems

Goals

- Building strong portfolio for his 401-k
- Finding new friends

MVP, Technologies & Algorithms

Minimal Viable Product



Home Page

- A welcome message that briefly introduces the app and its purpose
- A button or link to create an account or log in if you already have one
- A list of features and resources available within the app, such as self-assessment questionnaires, mood trackers, and mental health resources
- A prominent call-to-action encouraging users to take a self-assessment or start tracking their mood



Profile/History of Consultation

- A section where you can edit your personal information, such as your name, email address, and password

Minimal Viable Product



Image upload/Take Picture

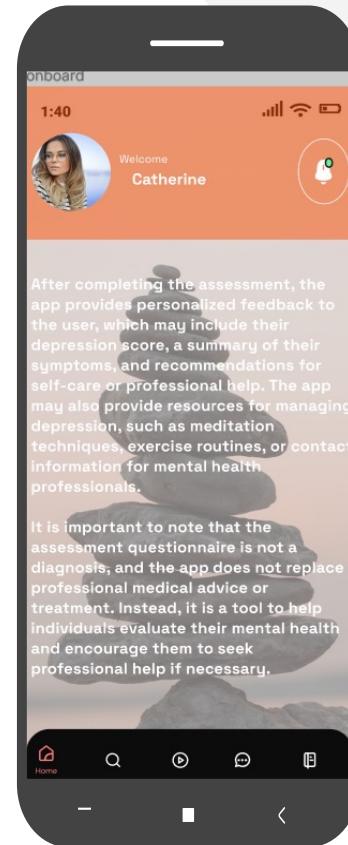
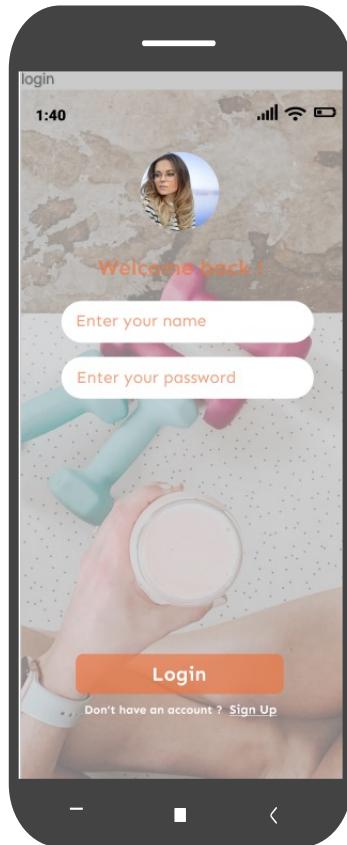
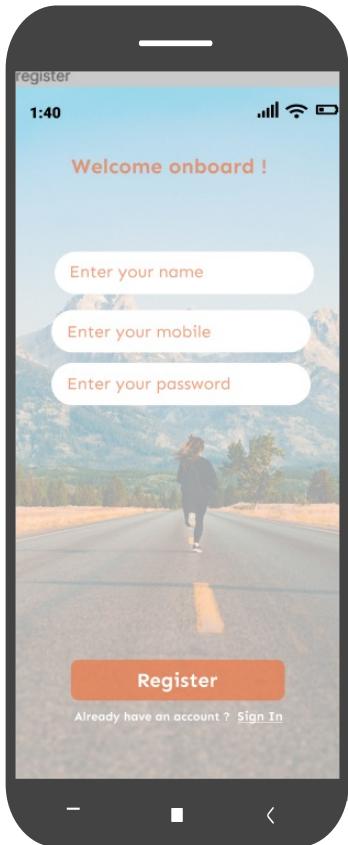
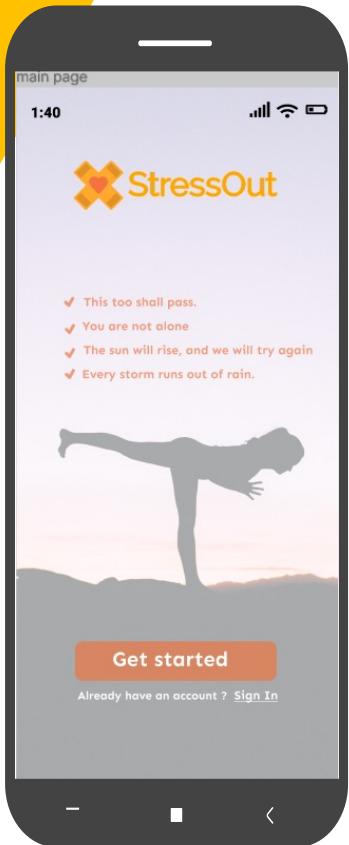
- A camera button that allows you to take a picture directly within the app
- A gallery button that allows you to upload pictures from your camera roll
- An option to add captions or notes to each image, to help you remember important details or context



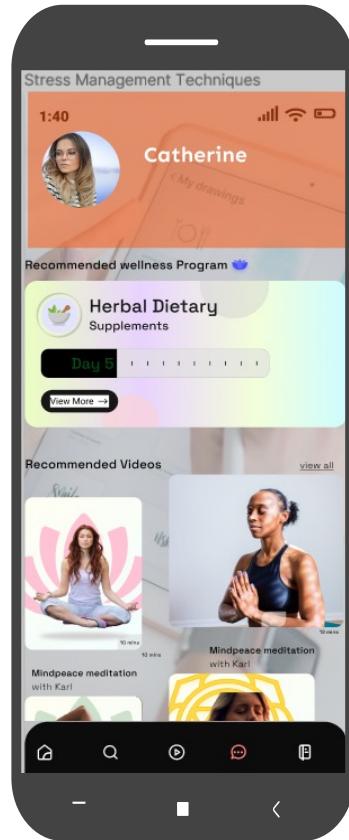
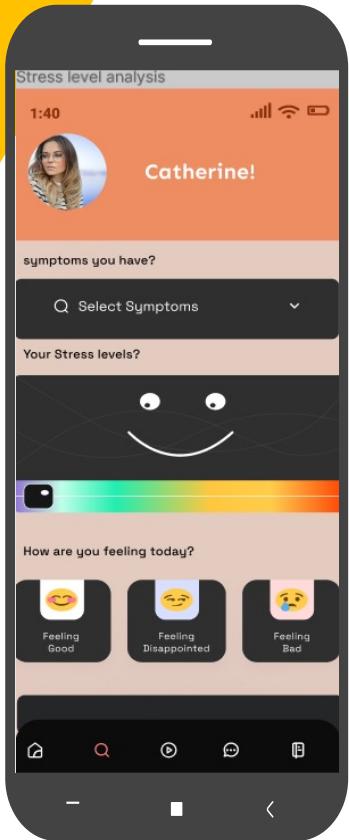
Recommendation

- A dashboard that displays any recommendations or next steps based on your self-assessment results, including suggestions for lifestyle changes, self-care practices, or professional treatment options

Prototype



Prototype



Algorithms & Technologies



TensorFlow



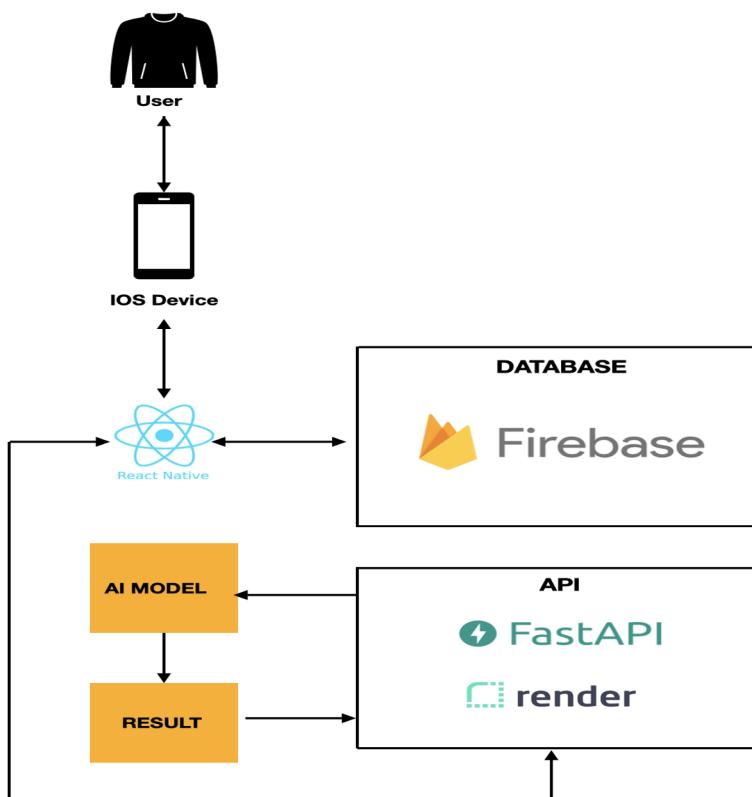
Keras



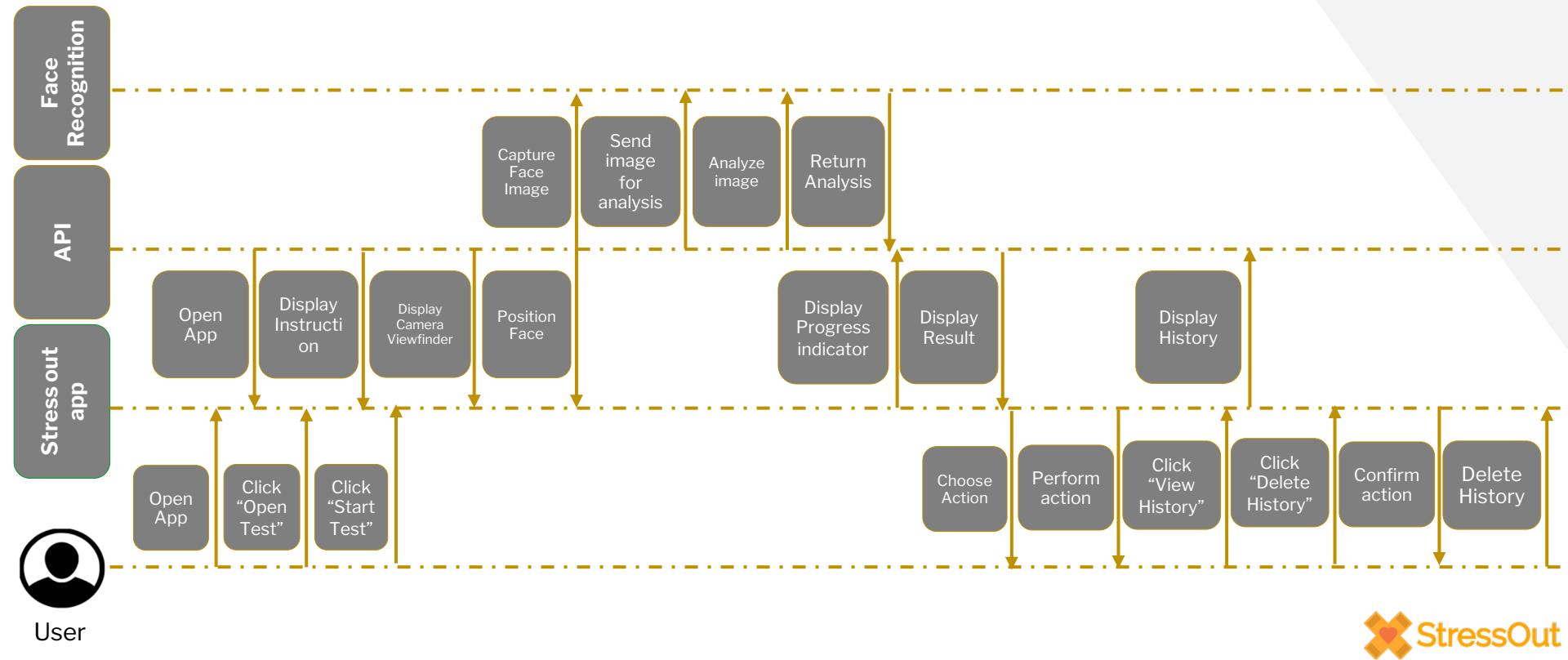
Firebase



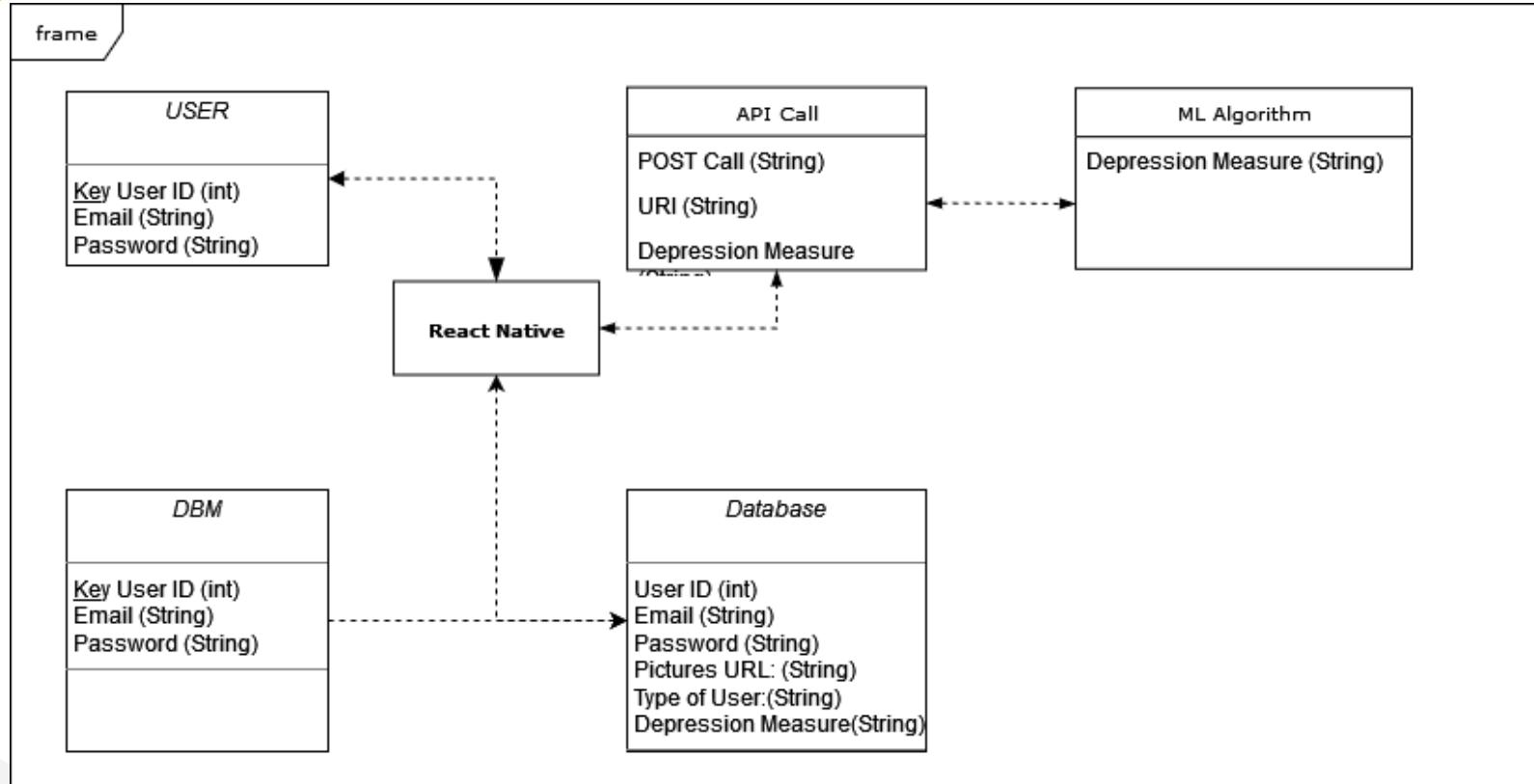
Conceptual Architecture Diagram



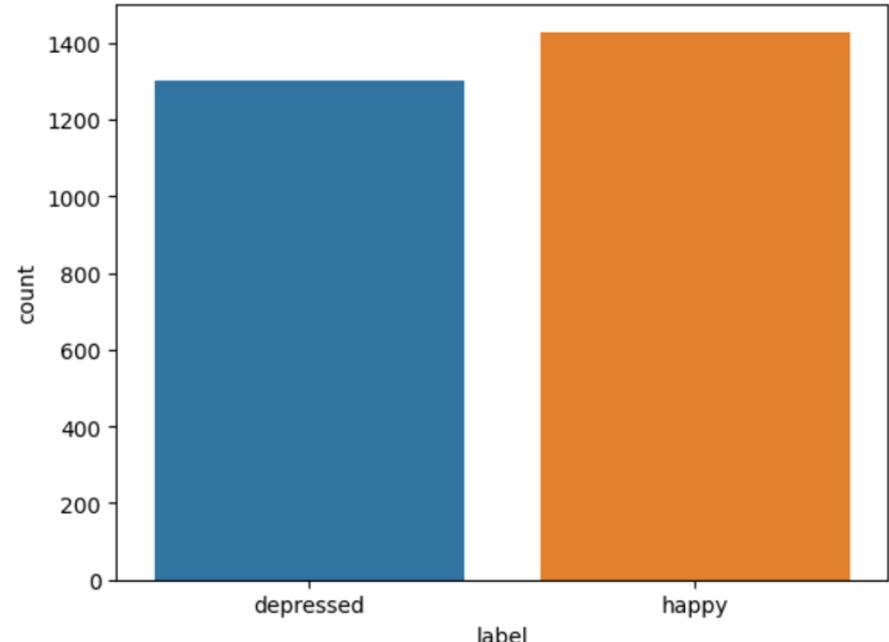
Sequence Diagram



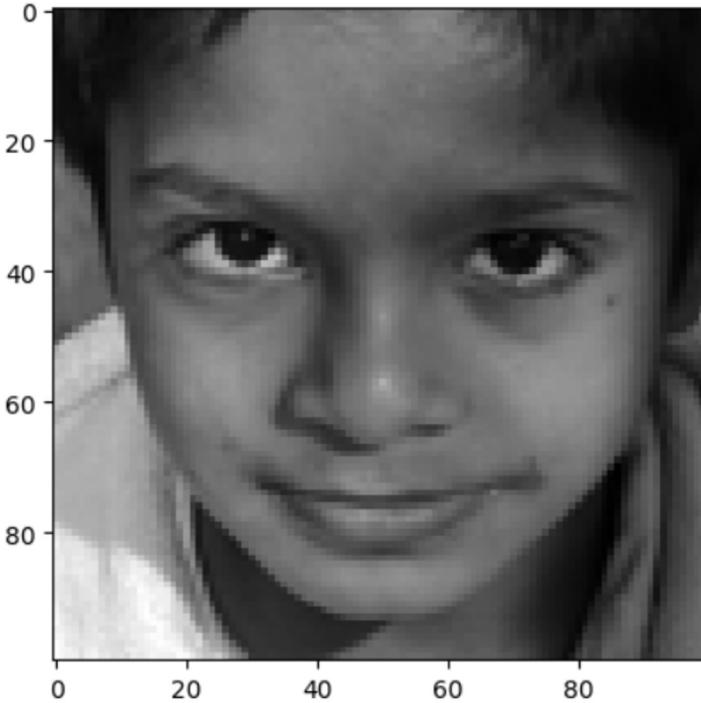
UML Class Diagram



ML Model : Dataset Adjustment

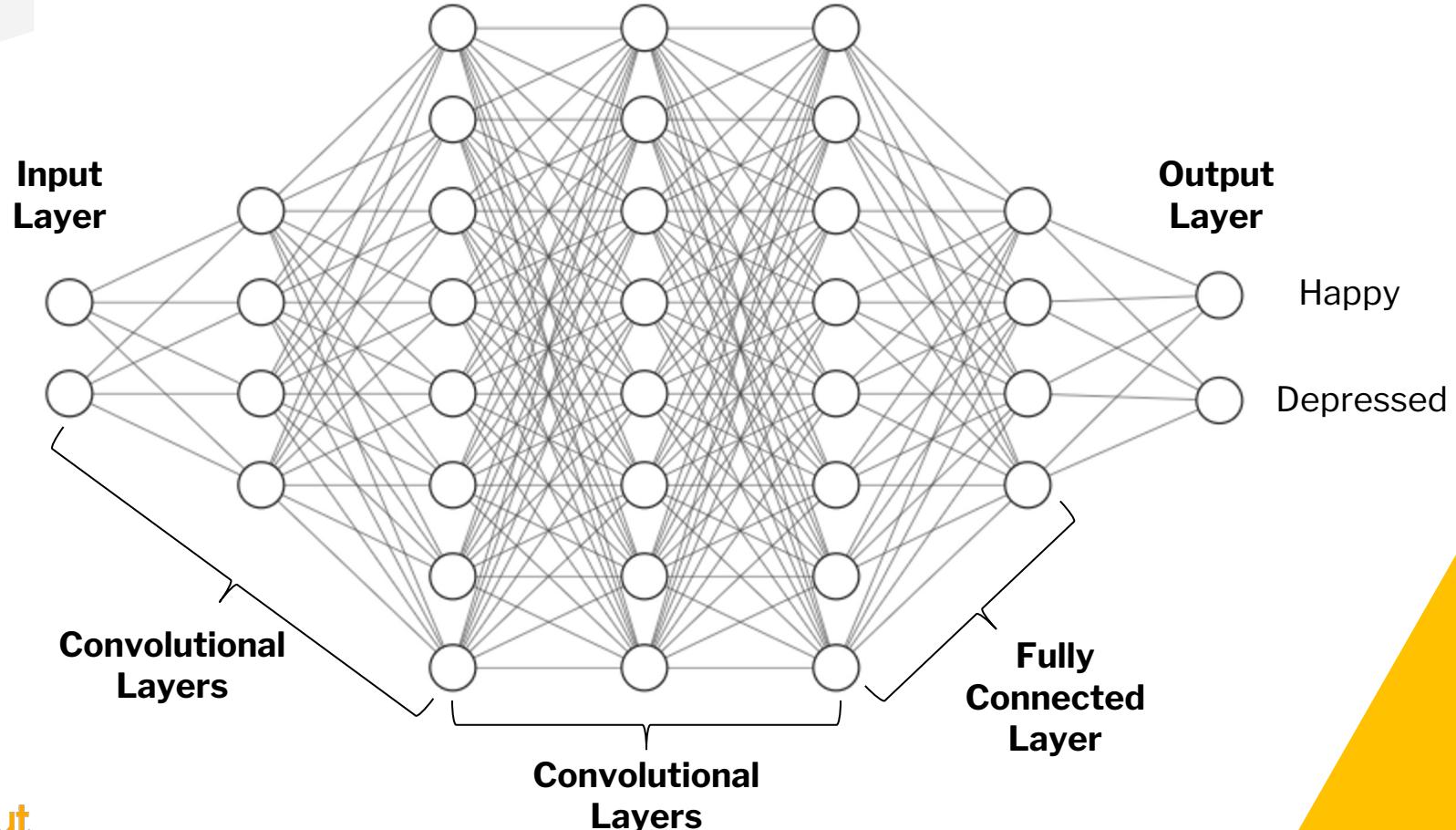


ML Model : Pre-process



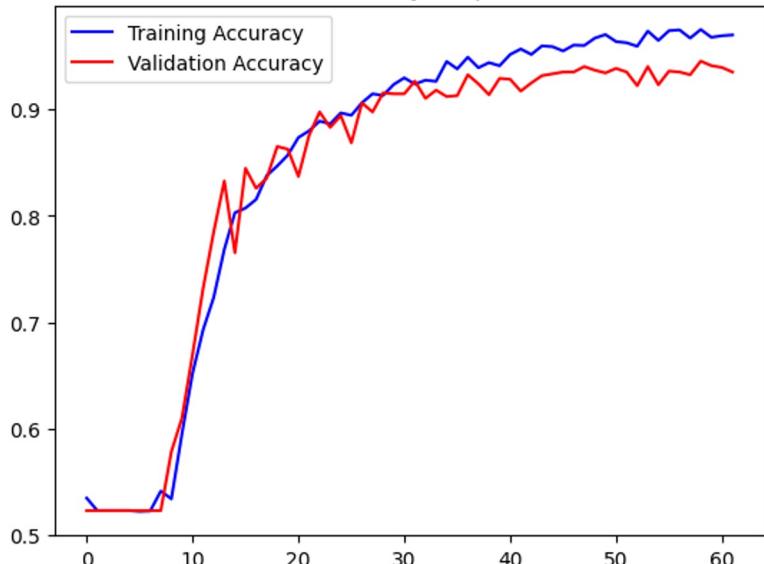
- Training dataset: 70% of the images
- Testing dataset: 30% of the images
- Resizing all images to 100x100 pixels
- Grayscale all images

Model Structure

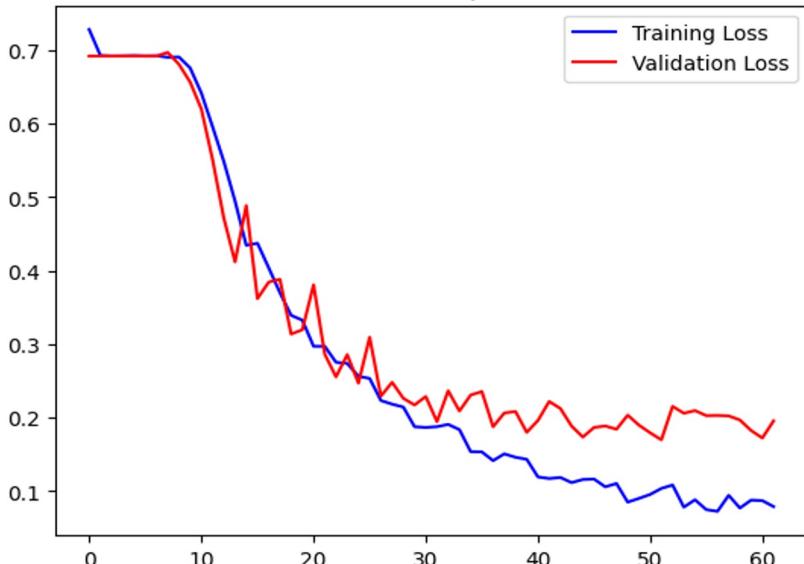


Model Training

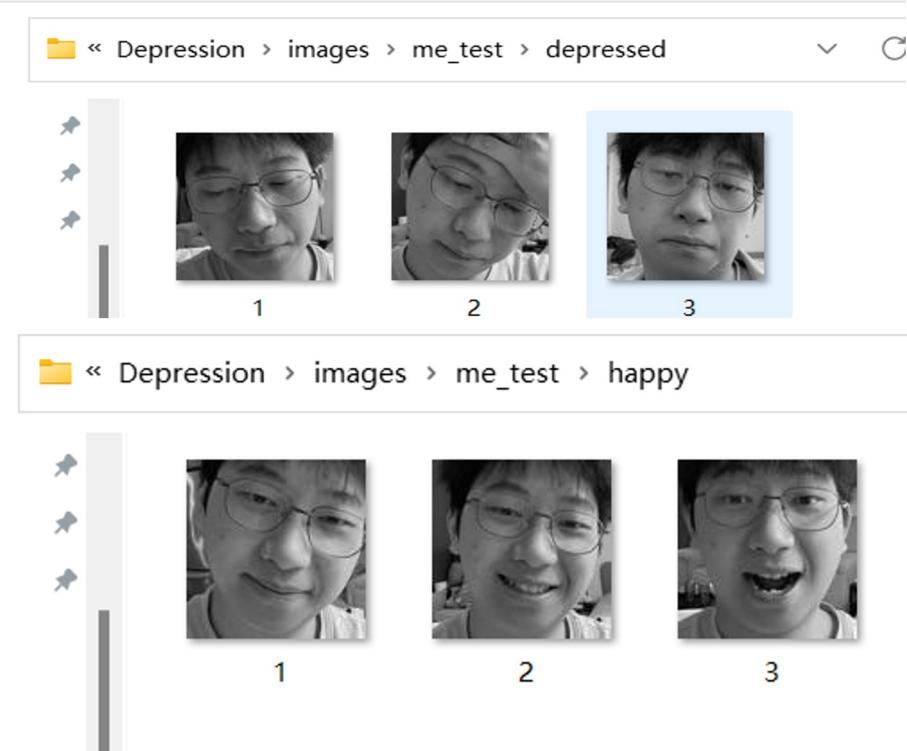
Accuracy Graph



Loss Graph



Model Testing



```
.. 1/1 [=====] - 33s 33s/step
Original : Predicted, depressed : depressed
1/1 [=====] - 0s 47ms/step
Original : Predicted, depressed : depressed
1/1 [=====] - 0s 12ms/step
Original : Predicted, depressed : depressed
1/1 [=====] - 0s 12ms/step
Original : Predicted, happy : happy
1/1 [=====] - 0s 41ms/step
Original : Predicted, happy : happy
1/1 [=====] - 0s 13ms/step
Original : Predicted, happy : happy
1/1 [=====] - 0s 12ms/step
Original : Predicted, happy : happy
{}
Overall Accuracy: 1.0
```

Sprint 3 Recap

Issue type	Key	Name
STRES-36	task	Develop Front End - homepage
STRES-39	story	As a user, I want to check the history of my photo. So that the app can keep checking my depression level.
STRES-48	story	As a user, I want to know some tips to help me. So that my depression doesn't get worse.
STRES-50	task	Further improve the accuracy of detecting happiness and depression

Product Backlog

User Stories

Key	Summary	Place
STRES-19	As a user, I want to register myself to the application. So that I can keep my account private and login using email/phone and password.	Home Page
STRES-20	As a user, I want to login in app. So that I can use it to store all my information.	Home Page
STRES-42	As a user, I want to upload my picture from phone gallery to analyze so that get more accurate information.	Image Upload/Take Picture
STRES-22	As a user, I want to take a picture from my front and back camera to upload so that the app can analyze	Image Upload/Take Picture
STRES-23	As a user, I want to view my upload history and their results	Profile/History of consultation
STRES-24	As a signed in user, I want to log out So that I can have my privacy in the app.	Logout/Login
STRES-25	As a user I want to be able to find contacts for professional help	Recommendation
STRES-26	As a user I want to be able to view history of my recommendation	Recommendation
STRES-27	As a user, I want to create my profile So that I can store my account information.	Profile/History of consultation
STRES-28	As a user, I want to be able to edit my information detail	Profile/History of consultation
STRES-29	As a user I want to be able to add captions for pictures I have uploaded	Image Upload/Take Picture
STRES-30	As a user, I want to be able to view personalized tips and strategies for managing symptoms	Recommendation
STRES-31	As a user, I want to know the sign of depression, and the risk of depression	Recommendation
STRES-48	As a user, I want to link to the hospital so that I can make a appointment with doctor	Recommendation
STRES-52	As a user, I want the app to keep checking my depression level so that I know when my depression level is worse	Profile/History of consultation
STRES-34	As a user, I want to know the risk of depression about me after I take pictures	Image Upload/Take Picture
STRES-35	As a user, I want to know some tips for relieving depression so that I can become better	Recommendation

Test cases & Acceptance Criteria

Acceptance criteria

Scenario	Summary	Criteria
1. User needs to get his mental health to be checked without going to the clinic	<p>Given I'm in the role of customer searching for a way to get his/her treatment from home.</p> <p>When I open the app Then the app asks me to register and login And then after my successful login, I should be able to tell my problems And I click the "submit" button Then the system starts to depict and analyze my Depression status.</p>	<p>Accuracy: The app should provide the user to login and provide an accurate and reliable depiction of depression, including its symptoms, causes, and treatments. This information should be based on scientific research and clinical evidence.</p> <p>User Experience: The app should have an easy-to-use interface and be designed with the user's experience in mind. It should be visually appealing, intuitive, and engaging.</p>
2. User wants to access account from a different device	<p>Given I'm in the role of the user trying to login to my account from another device When I try to login from another device Then the app asks me for my login details on the other device now Then I login to my account</p>	Compatibility: The app should be compatible with a wide range of devices and operating systems, including smartphones and tablets.

Acceptance criteria

Scenario	Summary	Criteria
3. User forgot password	<p>Given I'm in the role of the user trying to login but I have forgot my password</p> <p>Then the app shows me the option of 'forgot password'</p> <p>Then the app asks me for my registered phone number or the registered email id</p> <p>Where I will get the opt used for logging in my account again</p>	The app should save the users registration details so that if they forget their passwords then the app will be able to connect to them through their registered phone number or email id.
4. User wants to contact proper counselors or mental health professionals.	<p>Given I'm in the role of the user looking for proper mental health professionals</p> <p>Then the app shows me the option to connect to the app's support team</p> <p>Where I can get the relevant details, I want.</p>	The app should let the user to connect to the customer service team.

Acceptance criteria

Scenario	Summary	Criteria
5. User wants to give his feedback.	<p>Given I'm in the role of the user who have got my depression and mental health checked by the app</p> <p>Then the app gives me my results</p> <p>And allows me to give my feedback.</p>	<p>Continuous Improvements: the app should be regularly updated and improved based on user feedback and changes in the field of depression research and treatment.</p>
6. User want to keep personal information and results to himself.	<p>Given I'm in the role of the user trying to see if anybody else can see my information</p> <p>Then I go to my home page and the settings section</p> <p>There I set my privacy settings.</p>	<p>Privacy: The app should be designed with privacy in mind, ensuring that users' personal information and data are kept secure.</p>

Test cases

Unit to test	Scenario	Test data	Expected Results
Login	Successful login	Email – <u>user@stressout.com</u> Password - userpass	Check results on entering valid User id & Password
Login	Invalid Username or Password	Email – <u>user@stressout.com</u> Password - userpass	Check results on entering valid User id & Password
Login	Invalid Username or Password	Email – <u>user@stressout.com</u> Password - userpass	Check results when User id is Empty & Login Button is pressed
Sign Up	Successful login	Email – any, Password -anything between 8 to 16 characters	Check results on entering valid User id & Password
Sign Up	error message as “username already in use”	Email – <u>user@stressout.com</u> , Password - userpass	Check results on entering valid User id & password
Forgot password	Will direct the email to reset the password	Email – <u>user@stressout.com</u> , Password - user	Check respond when a Password is containing all the given parameters
Profile	Profile creation		A list of features and resources will show up

Test cases

Unit to test	Scenario	Test data	Expected Results
Privacy	User data		All the user data is safe within their profile.
Profile	Show an error with missing fields		Check to missing/mandatory fields
Homepage	Services on the homepage will show up		Check if the categories are correct
Homepage	User should be able to access the homepage		Once directed with url, you will be first redirected to homepage
Homepage	User should be able to contact the customer service		Homepage must provide resources
Homepage	User should be able to search the desired resources		Contact information should be present for assistance

Completed Stories

Date	Name	Status
4/16/2023	As a user, I want to be able to add captions for pictures I have uploaded so that the app can get more accurate information.	Done
4/19/2023	As a user, I want to check the history of my photo. So that the app can keep checking my depression level	Done
4/23/2023	Create a database with Firestore so users details and images can be saved in the database	Done
4/29/2023	Create an API endpoint	Done
5/01/2023	Finish technical paper	Done

Sprint Summary

Sprint 1

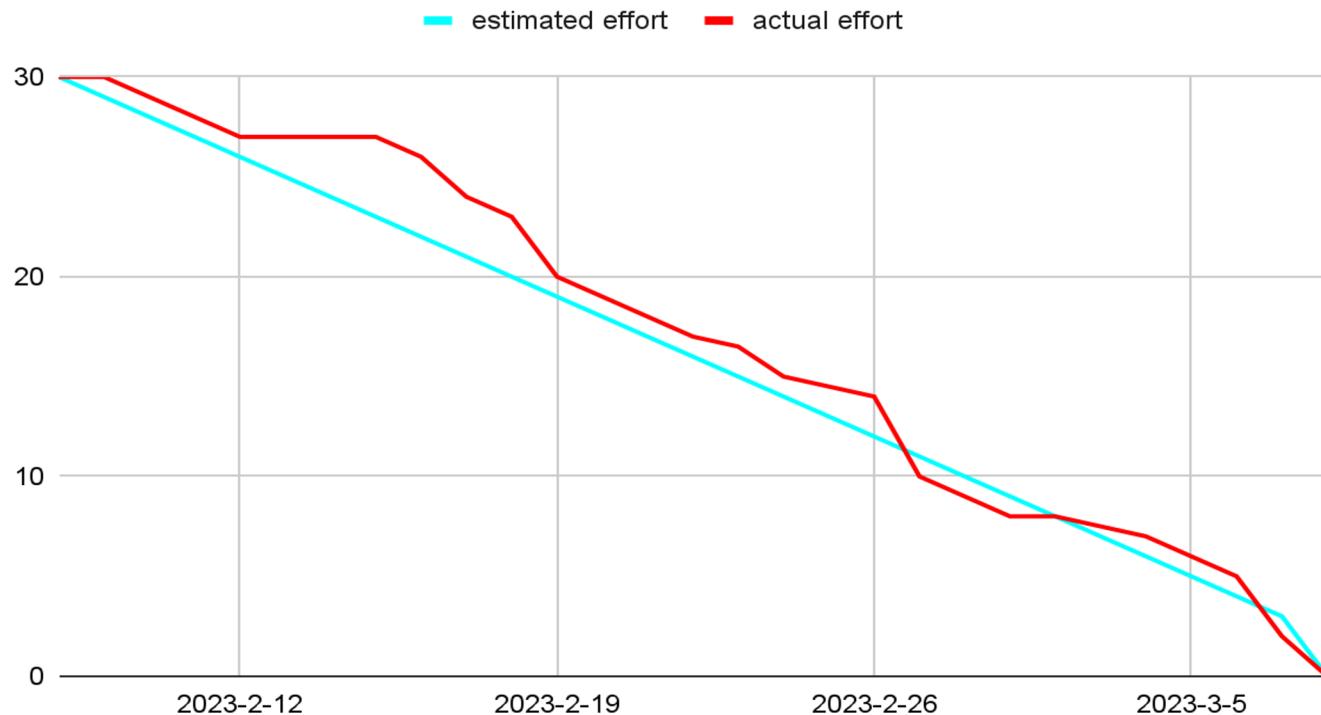
Key	Issue type	Name
STRES-2	Task	Establish teamwork agreement
STRES-3	Task	Set up development tools
STRES-4	Task	Establish the project schedule
STRES-5	Task	Establish each team members roles
STRES- 6	Task	Work on deliverable presentation

Sprint Summary

Sprint 2

Key	Issue type	Name
STRES-11	Story	As a user, I want to register, so that I know what this app can do
STRES-20	Story	As a user, I want to login by using email and password so that I can view this app
STRES-14	Story	As a user, I want to login So that I can store my information
STRES-9	Story	As a user, I want to know where I can take a picture So that I can detect my depression level.
STRES-18	Task	Start work on ML model

Sprint 2 burndown chart

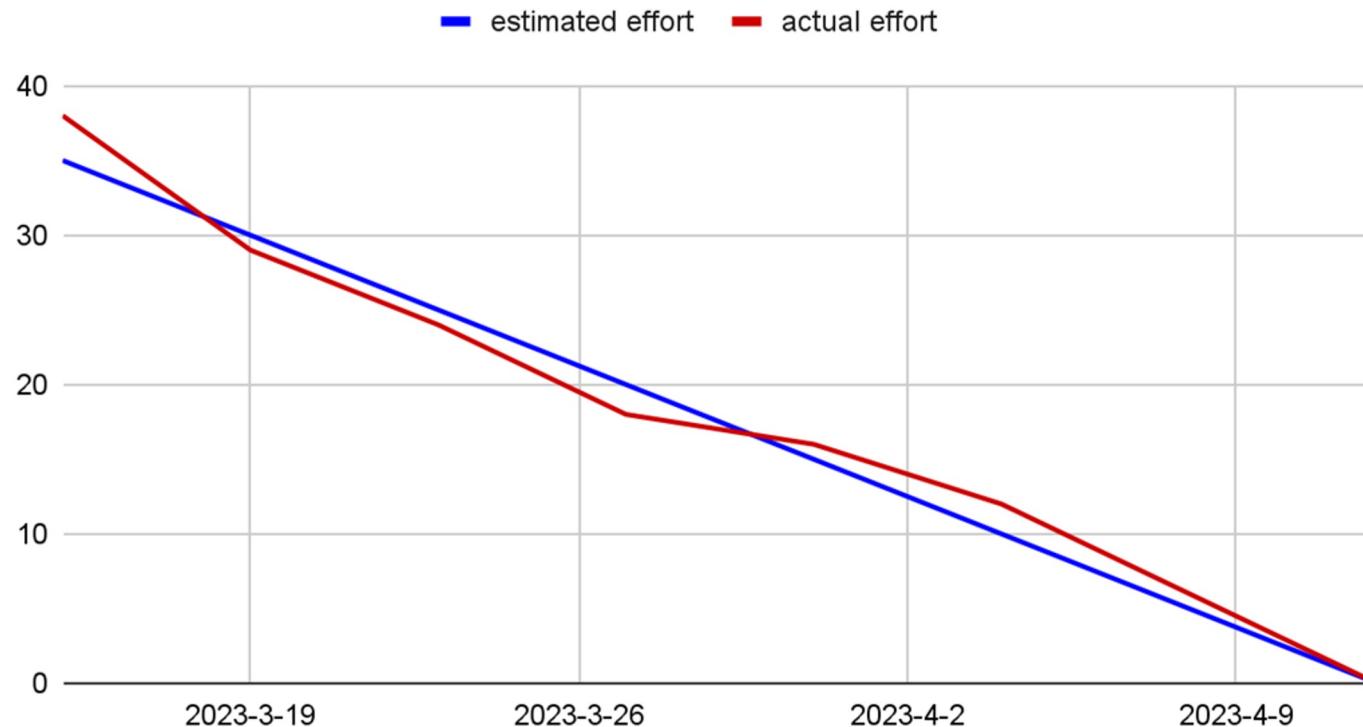


Sprint Summary

Sprint 3

Key	Issue type	Name
STRES-36	Task	Develop Front End - homepage
STRES-39	Story	As a user, I want to check the history of my photo. So that the app can keep checking my depression level.
STRES-42	Story	As a user, I want the app refer me to some therapists. So that I can get in touch with them
STRES-48	Story	As a user, I want to know some tips to help me. So that my depression doesn't get worse.
STRES- 50	Task	Further improve the accuracy of detecting happiness and depression

Sprint 3 Burndown Chart

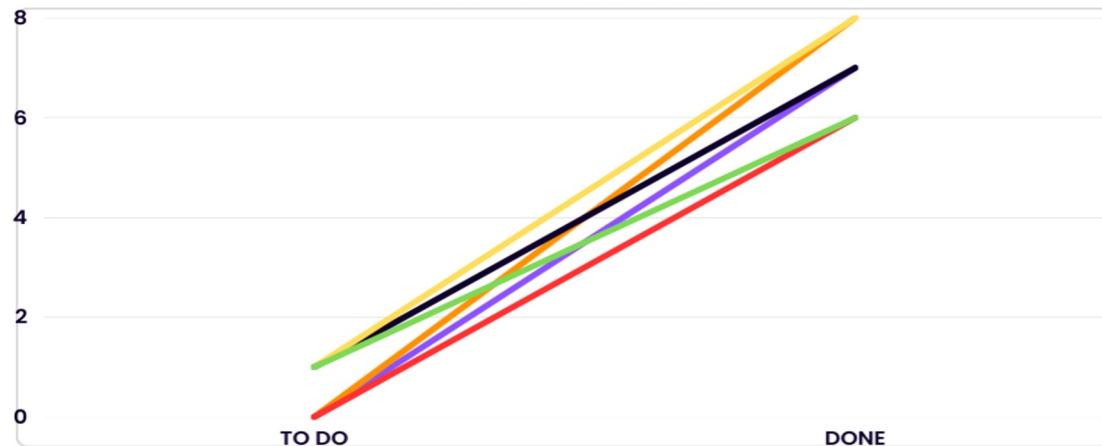


Sprint 4 Backlog

Key	Issue type	User stories
STRES-33	Story	As a user, I want to create my profile so that I can store my account information.
STRES-41	Story	As a user, I want to be able to edit my information details so that the app can keep tracking my information.
STRES-42	Story	As a user, I want to be able to add captions for pictures I have uploaded so that the app can get more accurate information.
STRES-54	Task	Finish the technical paper
STRES- 52	Story	As a user I want the app to keep checking my depression levels so that I can know when the status is high.

Team Members Velocity Report

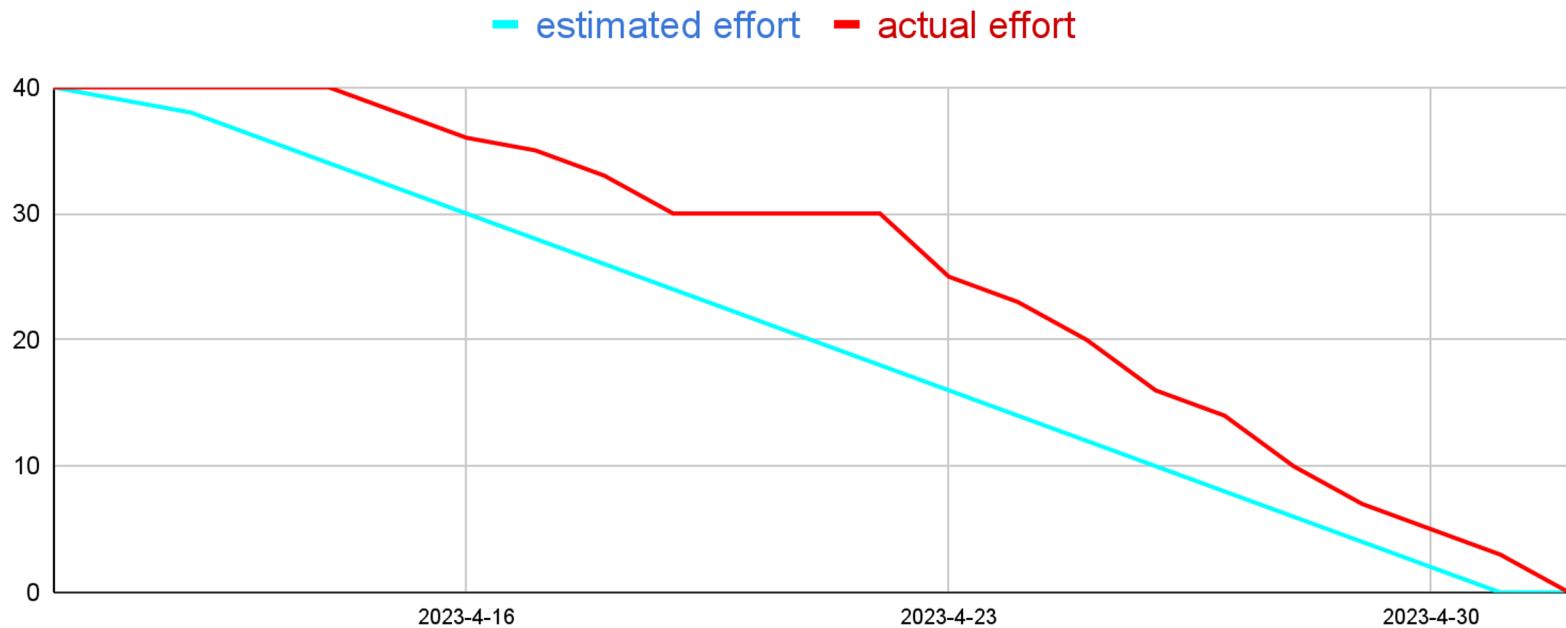
TEAM MEMBERS VELOCITY REPORT



● YUXIANG ● OMKAR ● SHIVANI ● WANGBO
● ARTEM ● SIDDHARTH

Sprint 4 Burndown Chart

sprint4 burndown chart



Changes Introduced with Sprint 4

Model Refinement

Model was refined with further training and achieved progress in model accuracy. Current model provides more than 95% accuracy.

Added functionality

We were able to add ability to take picture from front/back facing cameras in user's smartphone.

Text based quizzes added

We have added quizzes for the user that will be able to contribute towards accurately detecting depression.

App Screenshots

Home Page

App has 5 Functionality Buttons

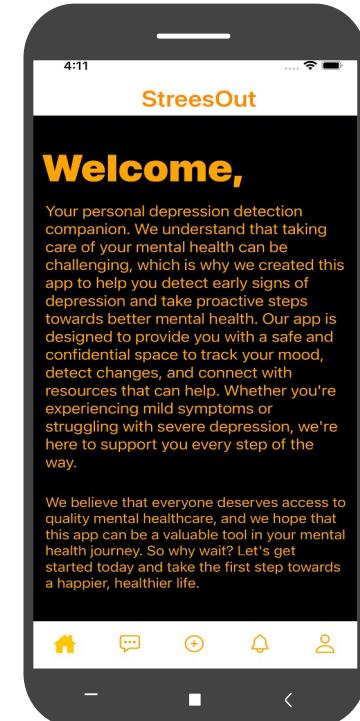
Onboarding Screen: A screen that welcomes the user to the app and provides an overview of its app.

Stress level analysis screen: A screen that displays the user's stress level based on the biometric data collected.

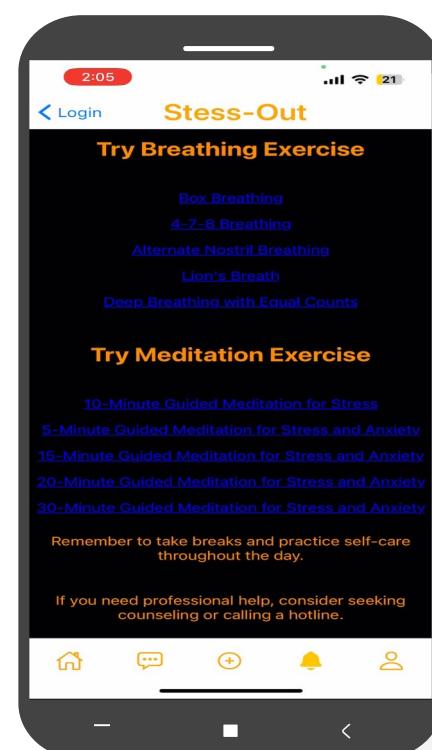
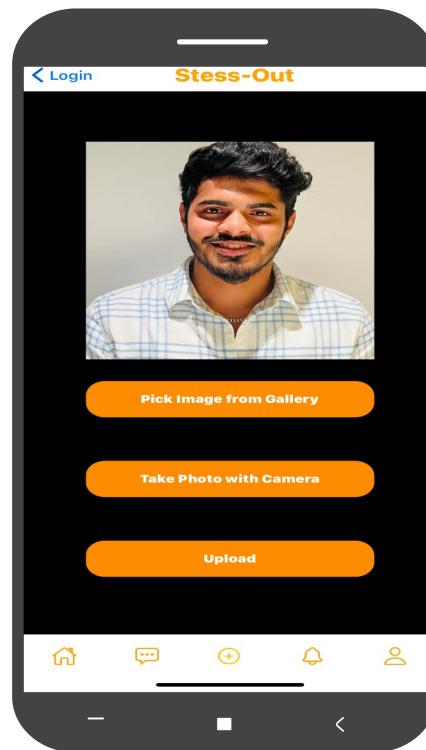
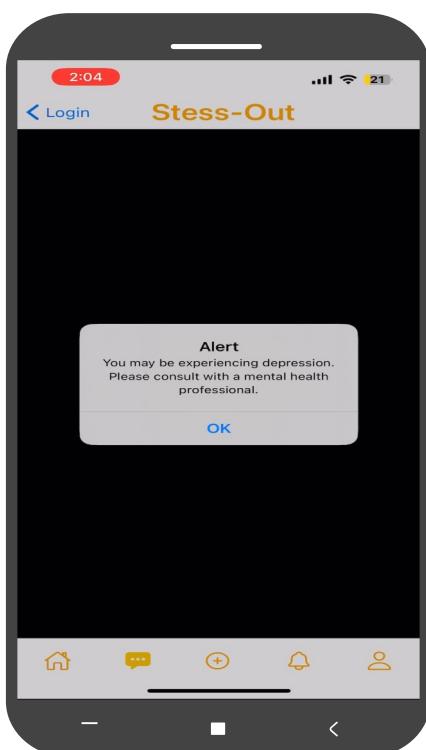
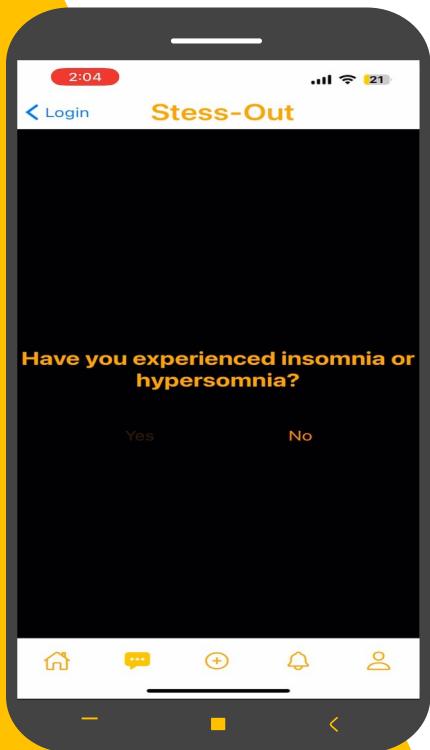
Quiz / Upload Images Screen: Quiz screen that asks the user questions to identify their sources of stress. Upload screen that allows the user to upload images

Stress management techniques Screen: A screen that provides stress management techniques, such as breathing exercises, meditation, or cognitive behavioral therapy.

Registration / Login Screen: A screen that allows the user to create an account or log in to an existing one.



API Screenshots



Firebase Authentication

The screenshot shows the Firebase Authentication console interface. On the left is a sidebar with various icons for stress testing and monitoring. The main header says "Authentication". Below it, there are tabs for "Users", "Sign-in method", "Templates", "Usage", and "Settings". A "Extensions" section is also present. The main area is a table showing two users:

Identifier	Providers	Created	Signed In	User UID
shivani@gmail.com	✉️	May 1, 2023	May 1, 2023	VYMagm94lEdo7pYEpzzTTmpYPL...
test@test.com	✉️	May 1, 2023	May 1, 2023	44i5VcXZZaZ5Bk7JaWWvc42Hr5...

Below the table are buttons for "Add user" and "CSV". At the bottom, there are pagination controls for "Rows per page" (set to 50), "1 – 2 of 2", and navigation arrows.

When a new user creates an account using Firebase Authentication in our app, the user's email address and a unique user ID (UID) are automatically generated and stored securely in the Firebase Authentication server.

Firebase Storage

The screenshot shows the Firebase Storage console interface. On the left is a sidebar with icons for Stress-out, Storage, Files, Rules, Usage, and Extensions. The main area is titled 'Storage' and shows a file named 'B703A19E-51C5-473C-998C-FE18AF47650D.jpg'. The file is 52.74 KB in size, an image/jpeg type, and was last modified on May 2, 2023. To the right of the file list is a detailed view of the selected file, showing its name, size (54,009 bytes), type (image/jpeg), and creation and update times (May 2, 2023, 11:20:43 AM). It also includes sections for 'File location' and 'Other metadata'.

When a user uploads or takes a picture in our app, the image is stored in Firebase Storage using the Firebase Storage client SDK. To maintain a separation between different users' data, we create a separate folder for each user in Firebase Storage, with the user's UID associated with that folder. This ensures that each user's data is stored in separate location. Firebase Storage generates a unique URL for each file that is stored, which can be used to retrieve the file later.

Firestore Database

The screenshot shows the Cloud Firestore console interface. On the left is a sidebar with various icons. The main area has a header "Cloud Firestore" with tabs for "Data", "Rules", "Indexes", "Usage", and "Extensions". A modal window is open, titled "Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing" with a "Configure App Check" button. Below the modal, there are two tabs: "Panel view" and "Query builder". The "Panel view" tab is selected, showing a tree structure of collections and documents. The path "users > 44i5VcXZZaZ5B" is selected. Under this path, there are two documents: "stress-out-a0631" and "44i5VcXZZaZ5Bk7JaWWvc42Hr5m1". The "44i5VcXZZaZ5Bk7JaWWvc42Hr5m1" document has fields: "downloadURL" (with a value of a Cloud Storage URL), and "email" (with a value of "test@test.com").

When a user signs up for the app, we create a new document for the user in the collection called users in firestore database using the Firebase Authentication UID as the document ID.

The image which is uploaded to a Cloud Storage bucket under a unique file name that includes the user's UID and the current timestamp. The getDownloadURL method is used to obtain a download URL for the uploaded file. Finally, the user's document in Firestore is updated with the URL of the uploaded image.

Depression Detection API



Python web application that will eventually run on the server (Render).



Intended to analyze facial images using machine-learning techniques and output the result: the probability of depression.

API Endpoints

Built on top of tensorflow model

The POST request expects a base64-encoded image in the body of the request

If the probability of depression is less than 0.01, the function returns a “<0.01” prediction

Otherwise, it rounds the probability to two decimal places and returns it as a string.

Request and Response

Request:

```
1 {  
2   "image": "/9j/4AAQSkZJRgABAQAAAQABAAAD/  
    2wBDAIABAQEBAQIBAQECAgICAgQDAGICAgUEBAMEBgUGBgYFBgYGBwkIBgcJBwYGCAsICQoKCgoKBggLDAsKDAkKCgr/  
    2wBDAQICAgICAgUDAwUKBwYHCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgr/  
    WAARCARWBFYDASIAhEBAxEB/8QAHwAAAQUBAQEAAAAAAAAAAECAwQFBgcICQoL/
```

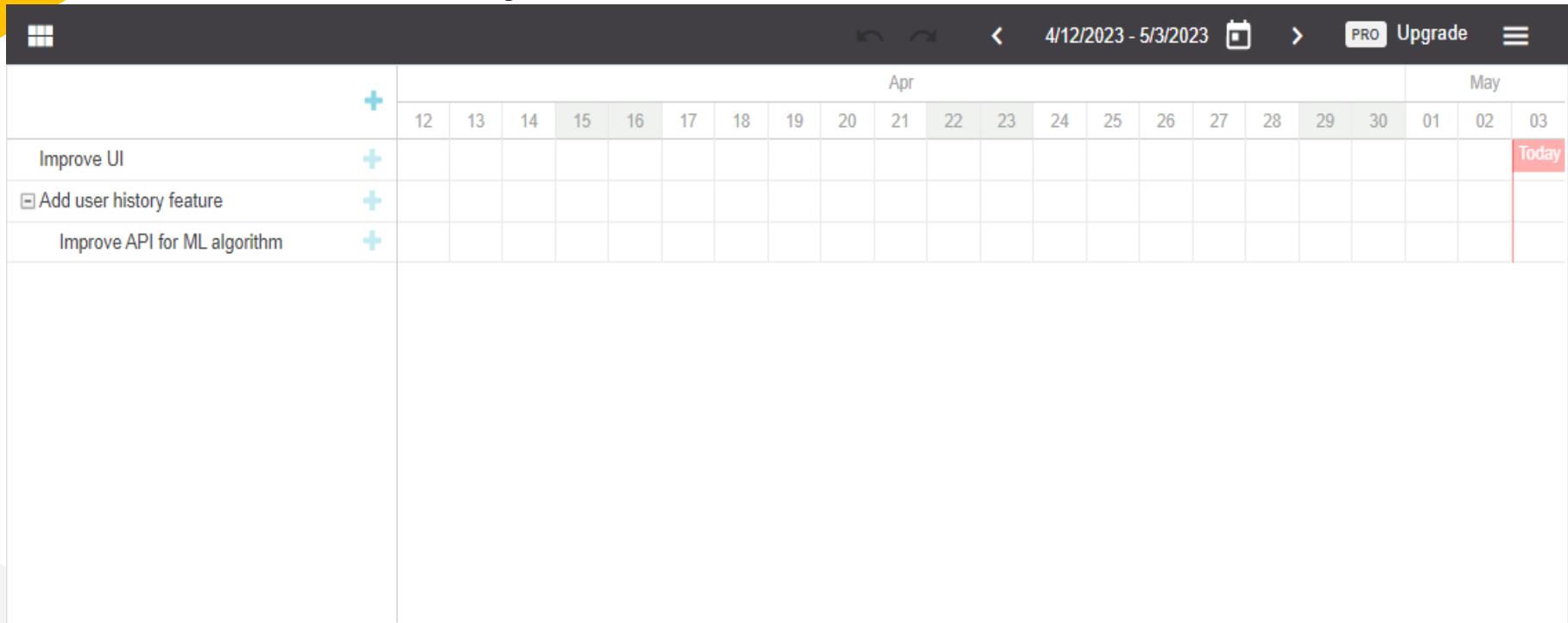
Response:

```
1 {  
2   "prediction": "0.8"  
3 }
```

Sprint 4 Schedule

	Apr																		May			
	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	
Completed API	Completed API																					
Created proper database								Created proper database														
Modified Technical paper														Modified Technical paper								
<input checked="" type="checkbox"/> Created Installation guide																			Created Insta			
API Documentation																			API Documer			

Sprint 5 Schedule



A screenshot of a digital sprint planning tool. At the top, there's a navigation bar with icons for calendar, search, and filters, followed by the date range "4/12/2023 - 5/3/2023" and upgrade buttons. Below the header is a grid-based calendar for April and May. The grid shows three tasks assigned to specific dates:

	Apr																			May			
	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	01	02	03	
+																							
Improve UI	+																						
☐ Add user history feature	+																						
Improve API for ML algorithm	+																						

The "Today" button is highlighted in red at the bottom right of the grid.

Retrospective

What went well: Team collaborated effectively with each other. Despite short sprint team managed to find solution for API issues we have had with our application. All changes were introduced in timely manner allowing team members to cross check each other's work

What needs improvement: We need to investigate cutting number of slides and reducing the length of presentation. Currently we have over 52 slides which makes presentation lengthy and difficult to render and upload to wiki.

Next Steps: In the next sprint members of the team will be asked to combine information from their slides in order to reduce the number of slides in presentation.

Thank you



CS-691

Team 2: Bug Terminator

Depression Detection System

[Github Link](#)

