Chlorophyll

CSE4015-Human Computer Interaction

in

Fall Sem 2021-2022

By

Nupoor Raj(19BCE2145)
Siddharth Singh(19BCE0370)
Dussa Lalith Kumar(19BCE0532)

BATCH - 3

Under the guidance of
Joshva Devadas T

Review 3 of J-component

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

TABLE OF CONTENT

ACKNOWLEDGEMENT

We would like to express our gratitude to our project guide Prof. Joshva Devadas T. This undertaking has certainly tossed light into this field and opened another skyline for us. The project could additionally move students and any remaining students to move forward the model to another level. The task was additionally effective in enhancing our viable information which is a lacking quality among present-day students.

ABSTRACT

The emerging resistance of crop pathogens to fungicides poses a challenge to food security and compels the discovery of new antifungal compounds. Productive agriculture systems are always vulnerable to hazards of climate and pests and diseases causing threats to the food security of any nation. Healthy and productive plants not only are essential but are the very essence of humankind, environment, for food, fiber, energy, and general well-being. Therefore we wanted to do our part and came up with the idea of Chlorophyll- The Crop Protection App.

The artificially intelligent mobile app that identifies if your crop is diseased. We plan on creating a pest detection algorithm that provides farmers with the phone numbers of local pest control facilities. We also show effective tips on agriculture and display predicted weather data so that farmers can plant their crops accordingly.

PROPOSED FEATURES:

1. Predicting disease of the crop Detection of plant disease through some automatic technique is beneficial as it reduces a large work of monitoring in big farms of crops, and at a very early stage itself it detects the symptoms of diseases i.e. when they appear on plant leaves. The algorithm would help the farmers to identify the diseased crop using a deep learning algorithm image segmentation technique, which is an important aspect for disease detection in plant leaf disease.
2. Providing remedies Now that the farmer knows the plant disease the app also provides information about the disease and provides measures to eradicate the disease, and lists necessary precautions from the database.
3. Reminder to water the crops Create an alarm-based schedule that alerts the farmer to water the crops and provides information about the number of fertilizers and pesticides needed to monitor plant health.

4. Accurate weather data in the app Provide weather information with good accuracy so that the farmers can schedule the harvest accordingly using an API.
5. Hand on the degree of soil fertility The user can determine if the fertility of the soil is suitable for the healthy growth of crops using an API.

INTRODUCTION

Every year, an estimated 10% of the global produce goes waste due to pests and crop pathogens. This is the reason why crop productivity in many countries is declining. For example, India is among the top producers of several crops such as wheat, rice, pulses, sugarcane, and cotton. Most farmers cannot tell if a crop is diseased or not just by looking. Farmers rely on the income they get from their crops and many go broke if their crops turn bad.

We have decided to choose the Agile Software development process model for our project “Chlorophyll”. Unlike the traditional Waterfall model where each step is done sequentially, we have the flexibility to work on overlapping activities through the Agile process model. Since we are a team of 3 and we would be working on different areas such as creating the deep learning module and application development at the same time. At some point in time, we would have the need to overlap different activities. (For Example: Cleaning the model train data and training the model while testing the application for the above-mentioned features)

Since the agile SDLC model is a combination of iterative and incremental process models with a focus on process adaptability, customer satisfaction is achieved better in the Agile Model. In our project, we are trying out some features that were never tried or implemented before. (Example: Improving the deep learning model accuracy at each iteration and asking the clients if all the proposed features work efficiently). So, it is good that we take in feedback by interacting more with the client. The agile model accounts for better interaction with the client.

REQUIREMENT ANALYSIS

We have Identified two target users for our application, they are the following:

i. People having plants at their home, belonging to any age group

ii. People whose occupation requires them to work with plants on a daily basis

To understand our user better we took our surveys of both groups in two different forms, to the casual plant keepers we gave a google form (sent to multiple college groups) and to the day to day workers we took a one on one survey and jotted down the answers they gave.

Link to Google Form (College students): https://docs.google.com/spreadsheets/d/1pHynP3pSvbmJuxqUiXrB9trNxdFSM1DvSMyu18Lr1lU/edit?usp=sharing

Link to In-Person Survey (Workers): https://docs.google.com/document/d/1KsnZzssdjn3g4CaLXDLBeDYNi1wFEVI9P71phWpzKE4/edit?usp=sharing


We have come to two User Profiles from these surveys

- Casual User

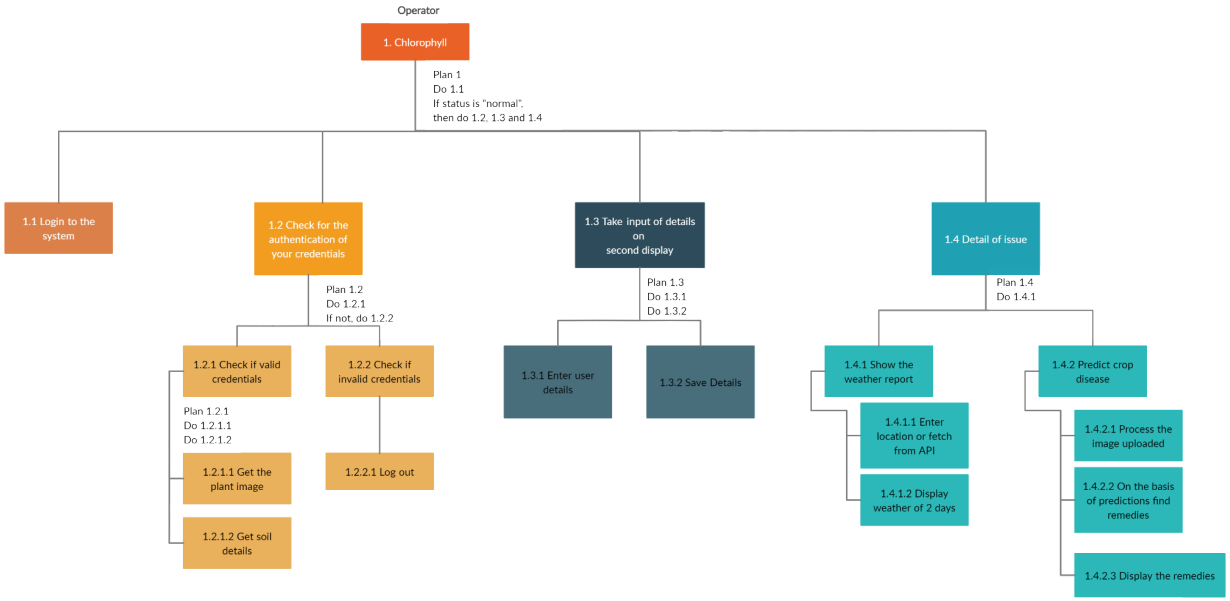
  Age: 18 - 50 years
  Gender: Both male and female
  Job Titles: Can be anything
  Education: Undergraduate level
  Language: Native and English
  Technology: High experience in technology
  Family: Single or Married

- Daily Worker

  Age: 30 - 50 years
  Gender: Both male and female
  Job Titles: Gardner - Farmer
  Education: School level

Language: Native language
Technology: Inexperienced
Family: Single or Married

DATA FLOW(HTA)



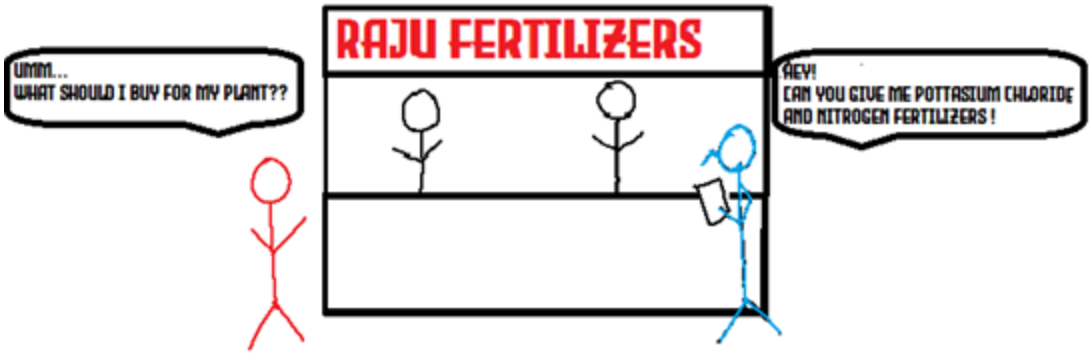
Note: The operator is the actor performing the task. The HTA assumes the existence of a control system but does not describe, in detail, actions for excercising control. This task analysis describes how the operator determines and evaluates a problem in the water system. The displays aid the operator to analyze and subsequently decide what control measures need to be performed to fix or regulate the flow of water.

STORY BOARD

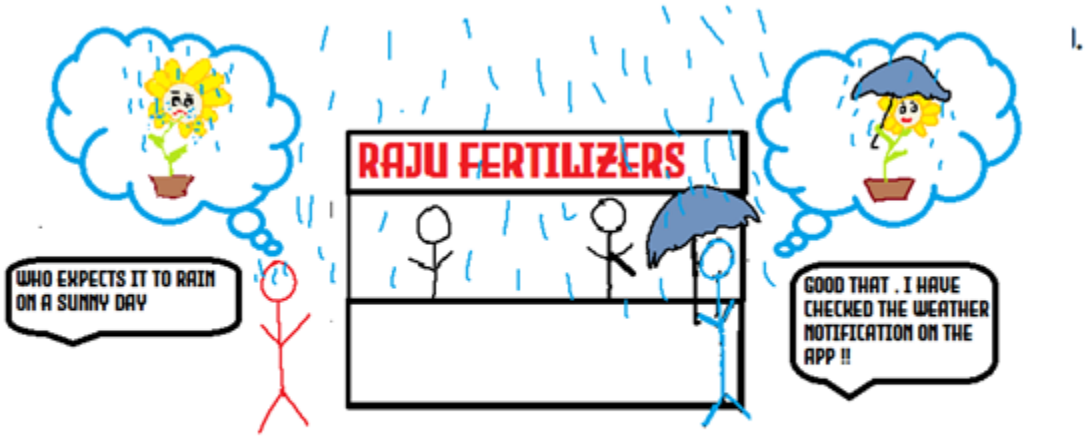
DOLAKPUR PLANT FESTIVAL!!

RAM AND SHYAM DECIDE TO PARTICIPATE IN THE PLANT FESTIVAL.

RAM FORGETS TO WATER HIS PLANT BUT SHYAM'S CHROLOPHYLL APP KEEPS HIM NOTIFIED.

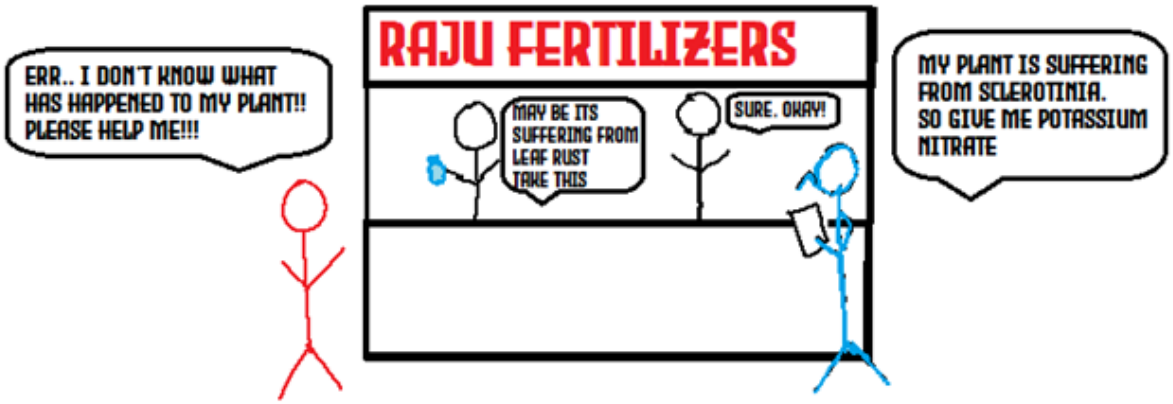


BOTH OF THEM GO TO THE BUY FERTILIZERS. RAM IS UNSURE ABOUT THE BEST FERTILIZERS WHERE AS SHYAM HAS ALREADY CHECKED UP THE BEST FERTILIZERS IN THE APP.



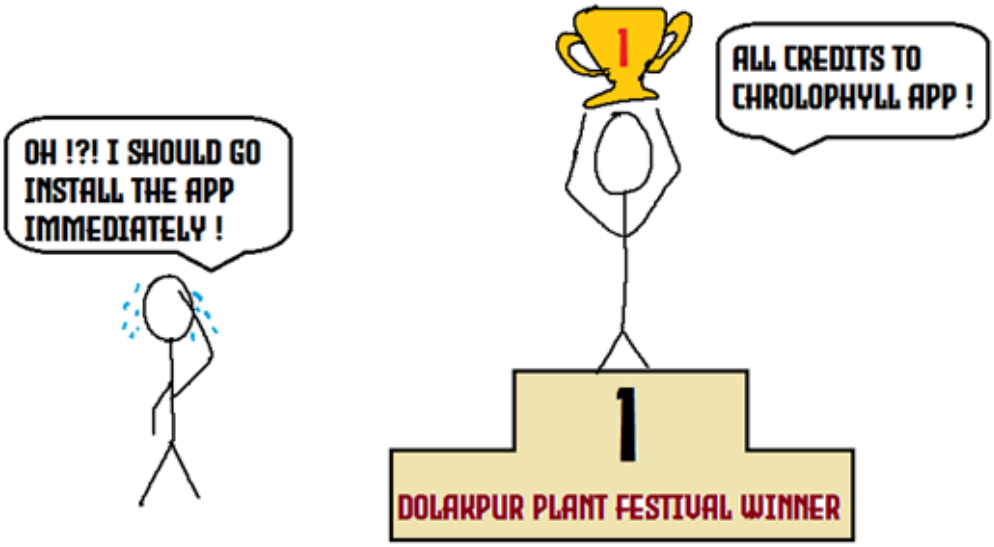
IT STARTS TO RAIN . SHYAM HAS ALREADY CHECKED THE WEATHER REPORT IN THE APP AND HAS TAKEN THE NECESSARY PRECAUTION FOR THE PLANT AND HIMSELF WHERE AS RAM WAS NOT PREPARED FOR THIS.

BOTH PLANTS ARE EFFECTED BY SCLEROTINIA . RAM IS CONFUSED BUT SHYAM USES THE APP TO TAKE A PICTURE OF THE PLANT LEAF AND IS NOTIFIED ABOUT THE DISEASE AND THE MEDICATION.



RAM CLUELESSLY ATTEMPTS TO EXPLAIN THE DISEASE TO THE SHOPKEEPER WHILE SHYAM KNOWS EXACTLY WHAT HE WANTS.

RAM'S PLANT IS STILL SUFFERING FROM THE UNKOWN DISEASE WHILE SHYAM'S PLANT IS BACK TO BEING HEALTHY.



USING THE CHORLOPHYLL APP SHYAM GOT TIMELY NOTIFICATIONS . WEATHER REPORTS. DATA ABOUT FERTILIZERS AND ALSO WAS ABLE TO FIGURE OUT THE PLANT'S DISEASE IMMEDIATELY SAVING EFFORTS AND TIME .

DESCRIPTION

(DESCRIBE YOUR PROJECT IN TERMS OF MODULES/METHODS/PROCEDURES)

1. Login and authentication:

We Sign up our users via their google accounts, the users are required to sign up using their email id and password, and we fetch their name and profile picture directly from their google account. We do not have to manage the security of their credentials, as google does it by default. The username and password are under protection from one of the most reliable companies on the planet.

2. Weather data:

We fetch the weather data from a third-party open-source API, that enables us to fetch the temperature, humidity, possibility of rain, and other important data, based on the location of the user. Our app sends the geographic coordinates of the user to the third-party API, and the API reverts with all the data that we project on our application. This keeps the user updated with the weather-related data, which helps them make better-informed care about their crop.

3. Soil fertility:

We use the third-party software, that enables us to determine the soil fertility of the geographic location, which helps the user to determine the type of crop and procedures to be followed to maintain the crop.

4. Reminders:

This module lets the user set alarms, with a title, and lets the user enable a repeat on a timely basis. It uses the phone's alarm module with our apps API to send out notifications to the user. The alarms help the user to keep following this schedule on a daily as well as a weekly basis to never miss any task that pertains to the care taking of their crop.

5. Identifying plant disease:

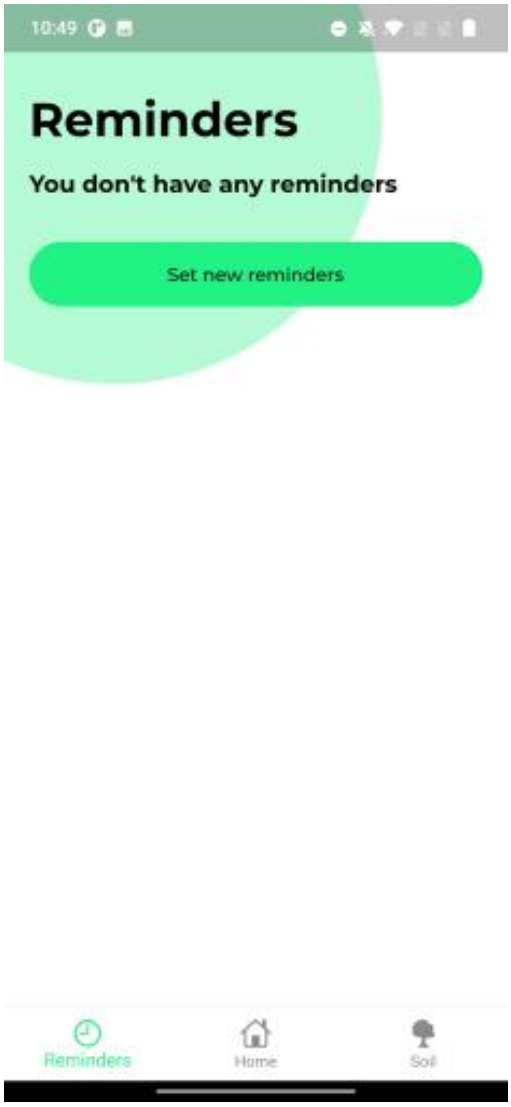
This module involves the analysis of the disease the plant might be having. The user is required to upload a picture of the plant's leaf into our application. Our application maps the image to a multi-layered convolutional neural network (CNN), the data is then converted to a JSON object and is mapped to the database to recognise the disease the plant might have. The data will be outputted in an easy-to-understand manner.

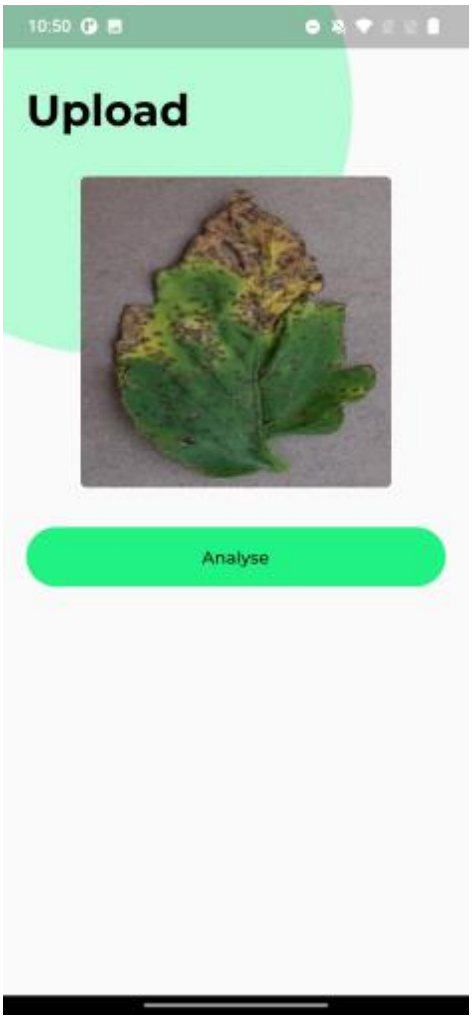
6. Suggesting cure for the disease:

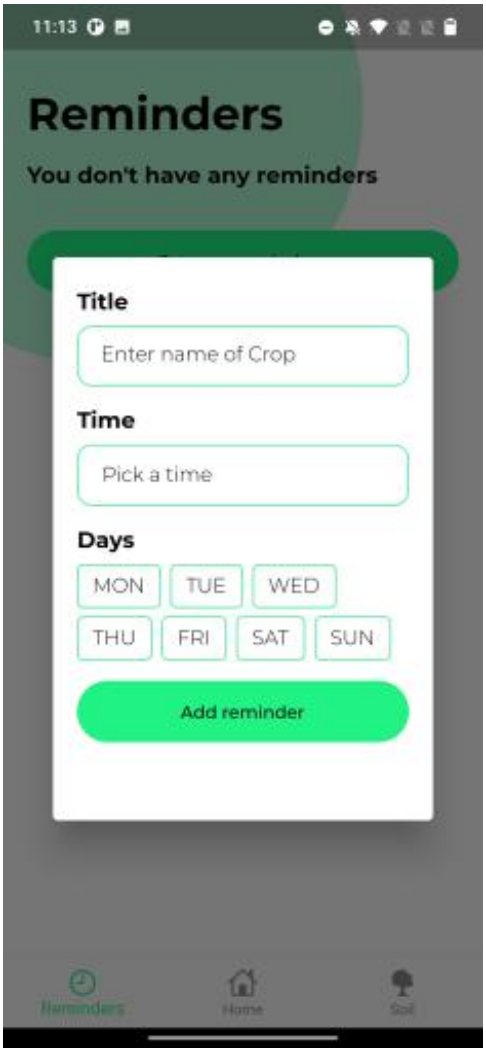
This module takes the data of the disease provided in the previous module and runs it through the database to fetch the remedy for the disease, the module also fetches data such as type of medicine, cost, time etc. This module helps the user to quickly find the remedy and start acting on the cure.

DESIGN SCREENSHOTS


10:49
Chlorophyll
Welcome!
To get started, sign in with Google
Google Sign In


10:49
Reminders
You don't have any reminders
Set new reminders
Reminders
Home
Soil


10:50
Upload
Analyse


11:13
Reminders
You don't have any reminders
Title
Enter name of Crop
Time
Pick a time
Days
MON
TUE
WED
THU
FRI
SAT
SUN
Add reminder
Reminders
Home
Soil

TEN HEURISTIC EVALUATION

1. Visibility of system status:

The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.

The users know on which page they are, and icons clearly direct them to other aspects of the app that are very easy to understand and navigate to on a single tap.

2. Match between the system and the real world:

The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon. Follow real-world conventions, making information appear in a natural and logical order.

The application uses simple words and phrases such as, “weather”, “alarm”, “soil”, “rain”, “scan the leaf” etc, which are used in general conversations. We have avoided using jargon and complicated or technical words, the diseases and their cure is also explained in an easy to understand manner.

3. User control and freedom:

Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process.

There are simple back options where the user can go back to the previous state, for uploading or scanning the leaf, the user can always redo the action and confirm upload when he/she is satisfied, or reupload a different photo. They reset alarms and delete them.

4. Consistency and standards:

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions.

The application uses standard icons and simple architecture. The navigation icons are always available for easy navigation and the UI is consistent throughout the application, making it very easy for the users to use the app, even on their first go.

5. Error prevention:

Good error messages are important, but the best designs carefully prevent problems from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

The application has barely any margin for error, it's a simple and regulated UI UX, with defined and allowed operations, the user cannot perform a mistake that might affect the performance or the outcome of the action, they can always undo or redo the action to get the desired output.

6. Recognition rather than recall:

Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g. field labels or menu items) should be visible or easily retrievable when needed.

The application uses very minimal menu items, the user can only switch between 3- 4 screens including their profile page, making it an easy interface. The navigation bar is situated at the same place across the application, and the user is free to navigate with a single click. There is practically nothing the user must memorize to be able to use the application.

7. Flexibility and efficiency of use:

Shortcuts — hidden from novice users — may speed up the interaction for the expert user such that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Our minimalistic and easy to use interface makes it very simple to navigate through and perform actions, there is practically only one way in which each operation can be performed and it is only a couple of clicks away(1 or 2 clicks), the only difference an experienced user can have over a novice one is using it faster.

8. Aesthetic and minimalist design:

Interfaces should not contain information which is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility.

A standout factor of our application is our minimalistic and aesthetic design, the content is visual and easy to understand, we have not cluttered any page with unnecessary information, and have not included irrelevant decorative items that ruin the aesthetic nature of the application.

9. Help users recognize, diagnose, and recover from errors:

Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.

10. Help and documentation:

It’s best if the system doesn’t need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.

The web application has been documented, and clear instructions have been scripted down which enable the users to walk through the application with ease. All the features and navigation pathways have been clearly documented to avoid any confusion and provide maximum clarity.

TESTING

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail	In Case of Failure
Chrpyl0001	Signing up for the first time.	1. Click on the sign in button. 2. Enter Gmail credentials.	The app will connect to google and check for the existence of the account and log the user in accordingly (User clicked on the icon, enter their email id and clicked on enter.)	The application will sign you in and redirect you to the home page, welcoming you with your name and climate data on it.	The application signed the user in successfully and greeted the user with a welcome message and showed the climate report.	PASS	Restart the application
Chrpyl0002	Logging out of the applicatio n.	1. Click on the profile icon. 2. Hit the log out button.	The app should log the user out of the current account.(User clicked on the log	The application will log the user out.	The application logged the user out successfully.	PASS	No failure possible

			out button)				
Chrpyl0003	Setting a reminder to water plants at 7am, on all days of the week.	1. Open the app and click on the reminder icon. 2. Choose and enter the appropriat e title. 3. Enter the time as 7am. 4. Select all the days Monday, Tuesday, Wednesday , till Sunday. 5. Click on 'Add reminder'	The user performs the mentioned steps and creates a reminder.	The app should be setting a reminder for 7am every day to water plants.	The app reminds the user everyday at 7am, to water his plants successfully.	PASS	Relogin to the application and set a reminder
Chrpyl0004	Relocates geographi cal location and tests climate data.	1. Open the app and check the climate report on the home page.	The user opens the app from pre-loaded cache memory (unclosed application) and checks the climate report. The user expected to see the	The application shows the climate data of the old geographical location, and not the current location.	The application still shows the climate data of the old geographical location.	PASS	Restart the application to get current data

			climate data correspon ding to the current location.				
Chrpyl00 05	Checkin g the soil report	1. Open the app and click on the soil report icon. 2. Check the data	The user clicks on the soil report icon and checks the soil data	The applicatio n should be showing accurate soil report	There was an error fetching this data. [the app failed to show soil report correctly]	FAIL	User may have exhausted the 100 request limit, we have to upgrade API plan
Chrpyl00 06	Setting more than one reminde r for the day	1. The user sets a reminder at 8am for Tuesday. 2. The user sets another reminder that goes off every Tuesday and Wednesd ay at 8am and 6pm.	The user follows the standard procedure of setting a reminder and sets two clashing reminders.	The most recently set reminder goes off, while the previous one doesn't ring.	The most recently set alarm goes off, while the previously set reminder doesn't get notified.	PASS	Cannot fail

CONCLUSION

Most deep learning models for automatic disease detection suffer from reduced performance when applied to real-world images previously unseen. In this paper, we show the feasibility of training a convolutional neural network (CNN) model using segmented and annotated images instead of full images. When the same CNN model is trained using the segmented images (S-CNN) as compared to training using full images (F-CNN), model performance on independent data increases from 42.3% to 98.6%. Furthermore, quantitative analysis of self-classification confidence showed a significant improvement with 82% of test datasets showing an increase in confidence. As better datasets become available in the future, pre-processing of images before model training in CNN can prove invaluable to achieve high real-world performance.
Hence, using a CNN model we were able to achieve an accuracy of about 95% and validation accuracy of 97%. And created suitable predictions on random images. The user can be used by farmers to enhance their productivity and increase the yield. The prediction model is hosted using a Flask API, and the model is created using a CNN model with weather and soil fertility data API based on location.

FUTURE SCOPE

- Training the model on more generic data
- Improving the reminders section with more personalization
- Creating our own API for rendering soil fertility data
- Creating a recommendation system for fertilizer and pesticides

REFERENCES

1. Shital Bankar et al, Plant Disease Detection Techniques Using Canny Edge Detection & Color Histogram in Image Processing / (IJCSIT) International Journal of Computer Science and Information Technologies, ISSN:0975-9646, Vol. 5 (2), 2014, pg. 1165-1168 (https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.658.416&rep=rep1&t ype=pdf)
2. A.S Deokar, Akshay Pophale, Swapnil Patil, Prajakta Nazarkar, Sukanya Mungase, Plant Disease Identification using Content-Based Image Retrieval Techniques Based on Android System, International Advanced Research Journal in Science, Engineering and Technology Vol. 3, Issue 2, February 2016, ISSN (Online) 2393-8021, ISSN (Print) 2394-1588
3. Noor Hussain Ladkhan, Shaheen Mujawar, A Content-Based Image Retrieval System for diagnosing Agricultural Plant Diseases, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 3 Issue 3, March - 2014 pg. 878-880
4. S. D. Khirade and A. B. Patil, "Plant Disease Detection Using Image Processing," 2015 International Conference on Computing Communication Control and Automation, Pune, India, 2015, pp. 768-771, DOI: 10.1109/ICCUBEA.2015.153.
5. Priyanka Soni et al, Study of Image Processing in Agriculture for Detect the Plant Diseases, International Journal of Computer Science and Mobile Computing, Vol.4 Issue.7, July- 2015, pg. 581-587
6. Fang Y, Ramasamy RP. Current and Prospective Methods for Plant Disease Detection. Biosensors (Basel). 2015 Aug 6;5(3):537-61. DOI: 10.3390/bios5030537. PMID: 26287253; PMCID: PMC4600171. 7. Image dataset link: https://www.kaggle.com/kaustubhb999/tomatoleaf