

7/20/2024

Project Plan

Milestone 1



P-PRJ371: Project Plan | Milestone 1

Table of Contents

Group Members	2
Project: Phenology Data Analytics of Aloe Ferox	3
1. Project Overview	3
2. Project Sprints and Tasks	5
3. Project Schedule & Timeline	8
4. Responsibilities	9
5. Release Planning for Software Development	9
6. Risk Management	10
7. Communication Plan	11
8. Budget and Resources	11
9. Recommended Software Tools	12
10. Success Criteria	12
11. Documentation/FSD Prep:	13
12. Class Diagram and High-level overview of user flow of the Proposed Solution	13
13. Proposed Technical Direction & Details:	15
14. Project Lifecycle Details:	16
15. Additional Considerations for The Project Life Cycle:	18

Group Members

- Ajay Ramkelawan	577913@student.belgiumcampus.ac.za
- Reino Powell	576713@student.belgiumcampus.ac.za
- Reon Gottsche	576949@student.belgiumcampus.ac.za
- Elias Modiga	576891@student.belgiumcampus.ac.za
- Andre Nienaber	576207@student.belgiumcampus.ac.za
- Johan Hattingh	574434@student.belgiumcampus.ac.za
- Jamie Frances Kench	577255@student.belgiumcampus.ac.za
- Thabiso Mokoena	1537@student.belgiumcampus.ac.za
- Tshepo Tsibolane	22143@student.belgiumcampus.ac.za

Project: Phenology Data Analytics of Aloe Ferox

Project Title: PID – Plant Identification Programme | Aloe Ferox

Project Date: June 03, 2024

Supervisor: Mr. Dino Giovannoni

Co-Supervisor: Mr. Phillip van Huyssteen.

Project Sponsor: Prof. Craig Peter, Department of Botany, Rhodes University

1. Project Overview

Objective:

- To use the provided dataset to characterize the phenology of *Aloe Ferox* based on the flowering stages of development.
- To use dates, seasonal and location data to identify the characteristics of the phenology.
- To calculate relevant phenological parameters like the average First Bloom Date (FBD), Last Bloom Date (LBD), Flowering Duration (FD), Peak Bloom Date, etc. for *Aloe Ferox* across the dataset in different geographic regions of the country.
- To use additional environmental weather data (e.g., temperature, rainfall, humidity, daylight hours, etc.) to explore potential correlations between these factors and flowering times. This could involve creating scatter plots or using correlation coefficients.
- To create maps visualizing the spatial distribution of FBD or LBD across the observed regions. Heatmaps or choropleth maps could be used.

Scope:

- Collecting images of *Aloe Ferox* from iNaturalist based on the identifier in the provided spreadsheet.
- Using the provided phenological data, together with weather and geographical data to provide a complete picture of the phenological characteristics of *Aloe Ferox*.
- Developing a suite of analysis tools to understand the temporal, geographic and environmental phenological characteristics of *Aloe Ferox*.
- Obtain weather data for the particular location of the images from the South African weather service, Google Earth Engine, or other sources.
- Using appropriate maps to superimpose location and other phenological data.
- Perform some detailed data analytics to characterize the phenology of the data. This could be used to identify the time of year for each stage of flowering, the

relationship between various weather data for the various stages of development and any geographical characteristics.

Key Deliverables:

- Design a web interface with an appropriate dashboard to visualize the Aloe Ferox flowering data for easy exploration by users.
- An interface that allows for the selection of the various phenological stages.
- Various types of data analysis providing scatter plots, geographic maps, correlation plots, etc. based on the phenological parameters.
- Calculation and display of various phenological parameters as described in the objectives.
- An interface to show images illustrating the various phenological stages of development.

Tools & Techniques:

The following tools and techniques may be required for the execution of the project:

- **Data Analytics:** Developing suitable data analytics on datasets.
- **File I/O:** Importing and exporting of data files.
- **User Interface Design:** Developing appropriate application with suitable user interface.

Stakeholders:

- **Project co-supervisor:** *Mr. Philip van Huyssteen*. Technology Aided Biocontrol Group (TAB), Belgium Campus Itiversity.
- **Project supervisor:** *Mr. Dino Giovannoni*. Technology Aided Biocontrol Group (TAB), Belgium Campus Itiversity.
- **Project sponsor:** *Prof. Craig Peter*. Department of Botany (RUBOT), Rhodes University.

2. Project Sprints and Tasks

Below is a detailed List of the sprints that the team will be running during the project life cycle coupled with multiple tasks, duration of weeks and delivery dates. This is to ensure the team keeps the project on track, allowing for continuous collaboration and iteration development withing an agile software development model.

Sprint 1: Project Planning

- **Duration:** 9 weeks
- **Start Date:** 2024-05-20
- **End Date:** 2024-07-20
- **User Stories and Tasks:**
 - **Requirement Analysis** (RG, TT)
 - **Sprint Duration:** 2024-06-10 to 2024-07-12
 - **Project Plan** (RG, TT)
 - **Sprint Duration:** 2024-06-10 to 2024-07-12
 - **Data Analysis Planning Document** (RG, TT)
 - **Sprint Duration:** 2024-06-10 to 2024-07-12
 - **Technology Stack Collaboration** (RP, AN, JH)
 - **Sprint Duration:** 2024-06-10 to 2024-07-12
 - **Dataset Cleanup and Usability Planning** (RG, TT)
 - **Sprint Duration:** 2024-06-10 to 2024-07-12
 - **UML/Class Definition Diagram** (RP, AN, JH)
 - **Sprint Duration:** 2024-06-10 to 2024-07-12
 - **Design Patterns** (RP, AN, JH)
 - **Sprint Duration:** 2024-06-10 to 2024-07-12
 - **Items Review** (Reon Gottsche)
 - **Sprint Duration:** 2024-07-15 to 2024-07-18
- **Milestone:** Planning Deadline on 2024-07-21

Sprint 2: Data Collection and Preparation

- **Duration:** 4 weeks
- **Start Date:** 2024-07-21
- **End Date:** 2024-08-18
- **User Stories and Tasks:**
 - **Dataset Design** (Ajay Ramkelawan)
 - **Sprint Duration:** 2024-07-22 to 2024-08-30
 - **Data Analytics Design** (RG, RC, AR)
 - **Sprint Duration:** 2024-07-22 to 2024-08-30
 - **UX Design** (RG, AR)
 - **Sprint Duration:** 2024-06-10 to 2024-08-30

Sprint 3: Software Development

- **Duration:** 8 weeks

- **Start Date:** 2024-08-01
- **End Date:** 2024-09-30
- **User Stories and Tasks:**
 - **Backend Development** (RP, AN, JH)
 - **Sprint Duration:** 2024-09-02 to 2024-10-14
 - **Web Interface Development** (RP, AN, JH)
 - **Sprint Duration:** 2024-09-02 to 2024-10-14
 - **Design and Develop User Interface** (Software Development Team)
 - **Sprint Duration:** Continuous through Sprint 3
 - **Data Analysis Report Development** (Reon Gottsche)
 - **Sprint Duration:** 2024-09-02 to 2024-10-14
- **Milestone:** Design Deadline on 2024-09-06

Sprint 4: Testing and Validation

- **Duration:** 2 weeks
- **Start Date:** 2024-10-01
- **End Date:** 2024-10-14
- **User Stories and Tasks:**
 - **Testing the Software** (Testing Team)
 - **Validate Results with Experts** (Botanists)
 - **Items Review** (Reon Gottsche)
 - **Sprint Duration:** 2024-10-28 to 2024-10-31
 - **Solution Presentation Creation** (Thabiso Mokoena)
 - **Sprint Duration:** 2024-10-21 to 2024-10-31

Sprint 5: Deployment and Training

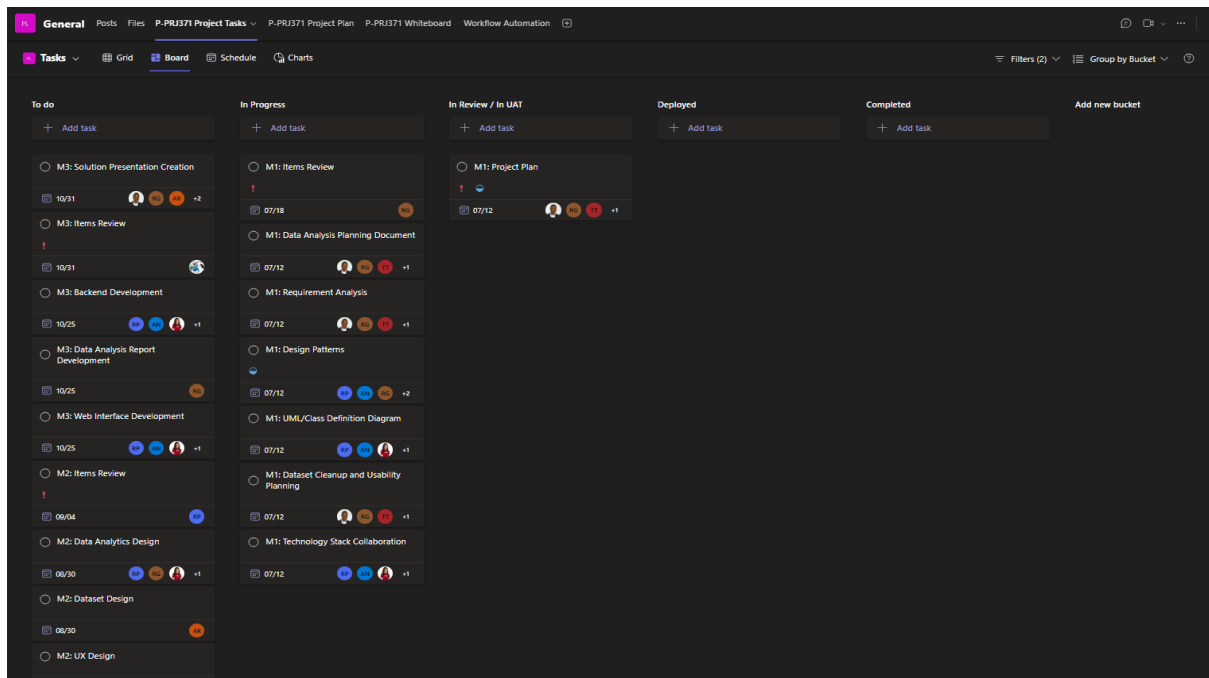
- **Duration:** 1 week
- **Start Date:** 2024-10-14
- **End Date:** 2024-10-21
- **User Stories and Tasks:**
 - **Deploy Software** (Deployment Team)
 - **Conduct Training Sessions** (Training Team)
 - **Provide Documentation and Support** (Documentation Team)

Sprint 6: Monitoring and Evaluation

- **Duration:** Ongoing
- **Start Date:** 2024-10-21
- **User Stories and Tasks:**
 - **Monitor Software Performance** (Monitoring Team)
 - **Collect User Feedback** (Feedback Team)
 - **Regular Updates and Improvements** (Development Team)
- **Milestone:** Implementation, Test & Demo Deadline on 2024-11-01

Project Task List with Assignment and Status:

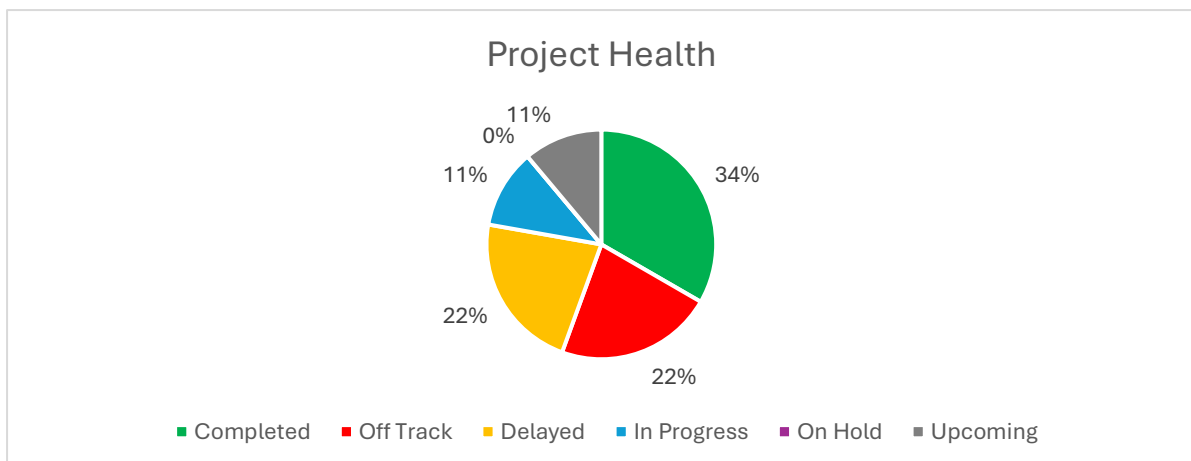
Below is a screen grab of how tasks will be tracked on Teams via the Planner Tool app.



Below is how we will track the sprints and tasks additionally as an example:

Sprint 1	Task	Duration	Status
Week 1	Requirement Analysis	7 Days	Completed
Week 2	Data Analysis Planning Document	7 Days	Completed
Week 3	Technology Stack Collaboration	7 Days	Completed
Week 4	Data set Cleanup and Usability Planning	7 Days	Completed
Week 5	UML/Class Definition Diagram	7 Days	Completed
Week 6	Design Patterns	7 Days	Completed
Week 7	Release Planning Document	7 Days	Completed
Week 8	Project Plan	7 Days	Completed
Week 9	Review	7 Days	In Progress

The below pie chart will display the Status per Item and how that would determine the Health of the Project as a whole.



3. Project Schedule & Timeline

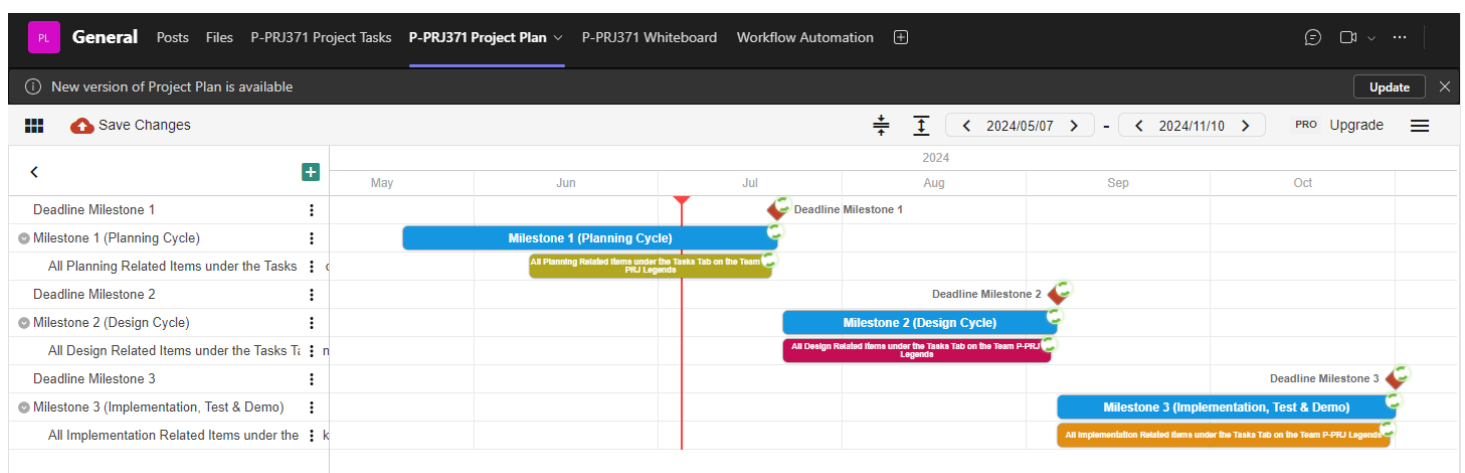
Below is the project schedule and timeline encompassing all major planned sprints with an overall deliverable to complete the solution within the project milestone dates and deadlines.

Sprint	Duration	Start Date	End Date	Milestone
Project Planning	9 weeks	2024-05-20	2024-07-20	Planning Deadline on 2024-07-21
Data Collection & Preparation	4 weeks	2024-07-21	2024-08-18	
Software Development	8 weeks	2024-08-01	2024-09-30	Design Deadline on 2024-09-06
Testing and Validation	2 weeks	2024-10-01	2024-10-14	
Deployment and Training	1 week	2024-10-14	2024-10-21	
Monitoring and Evaluation	Ongoing	2024-10-21	Ongoing	Implementation, Test & Demo Deadline on 2024-11-01

Event	Date
Kick-off Webinar	2024-05-13
Milestone 1: Planning (Cycle)	2024-05-20 to 2024-07-20
Milestone 1: Planning (Deadline)	2024-07-21
Milestone 2: Design (Cycle)	2024-07-01 to 2024-09-05
Milestone 2: Design (Deadline)	2024-09-06
Milestone 3: Implementation, Test & Demo (Cycle)	2024-09-16 to 2024-10-31
Milestone 3: Implementation, Test & Demo (Deadline)	2024-11-01
Expo Preparation	2024-11-04 to 2024-11-08
Expo: Diploma (D-PRJ361 & PRJ361)	2024-11-11 to 2024-11-15
Expo: Degree (P-PRJ371, PRJ371 & PRJ381)	2024-11-18 to 2024-11-22

Project Ghant Chart Timeline High View:

Below is the Project's life cycle Ghant chart. This enables stakeholders to visually see the timeline and progress of the project.



4. Responsibilities

Below are common terms of various team members and resources that will be partaking in delivering the project.

- **Project Manager:** Oversee project progress, manage resources, and ensure timelines are met.
 - **Data Scientists:** Compares and delivers dataset cleanup and analysis design.
 - **Software Developers:** Design and implement the user interface and software features.
 - **Botanists:** Provide expertise on phenology stages and validate the data analysis accuracy.
 - **Stakeholders:** Review progress, provide feedback, and approve deliverables.
-
- **Project Managers:** Elias, Reon, Thabiso and Tshepo
 - **Data Scientists:** Johan, Jamie, Andre and Reino
 - **Software Developers:** Jamie, Johan, Reino, Andre and Ajay
 - **Botanists:** *Prof. Craig Peter*
 - **Stakeholders:** *Dino Giovannoni and Philip van Huyssteen*
-

5. Release Planning for Software Development

This section outlines the release planning for multiple iterative versions of the solution during development in an Agile model. It includes the overall vision, release plans, iteration schedules, feature development within iterations, and tasks necessary to deliver each feature.

Vision Statement: Develop a comprehensive software solution that delivers a user-friendly interface, robust backend, and reliable data analysis capabilities. The solution should meet the needs of the end-users, align with organizational goals, and be adaptable to changing requirements.

Goals:

- Deliver a high-quality software solution that meets user needs.
- Ensure continuous improvement through iterative development.
- Incorporate feedback and adapt to changes efficiently.

Review Roadmap: Discuss the overall vision and product roadmap.

- **Review Architecture:** Evaluate the architecture and technical details.
- **Review Velocity and Iteration Schedule:** Present estimated velocity and proposed iteration schedule.
- **Establish Definition of Done:** Agree on the acceptance criteria for the release.

Velocity and Iteration Schedule:

Iteration	Duration	Start Date	End Date	Objectives
Iteration 1	2 weeks	2024-08-01	2024-08-14	Initial Backend Setup, Basic UI Framework
Iteration 2	2 weeks	2024-08-15	2024-08-28	Core Backend Functions, Initial Web Interface Elements
Iteration 3	2 weeks	2024-08-29	2024-09-11	Enhanced Backend Features, Intermediate UI Design
Iteration 4	2 weeks	2024-09-12	2024-09-25	Full Backend Integration, Advanced UI Components
Finalization	1 week	2024-09-26	2024-09-30	Complete Integration, Bug Fixing, Final Design Adjustments

Definition of Done:

- All user stories are completed.
- Code is reviewed and tested.
- Documentation is updated.
- User acceptance testing is passed.
- Product owner approves the release.

Release Calendar:

Release	Start Date	End Date	Objectives
Release 1	2024-08-01	2024-08-28	Initial working version with core backend and basic UI elements
Release 2	2024-08-29	2024-09-25	Enhanced version with advanced features and intermediate UI design
Final Release	2024-09-26	2024-09-30	Fully integrated solution with final design adjustments and bug fixes

6. Risk Management

The following risks have been identified and have proposed mitigation strategies in the event that they occur.

- **Data Quality:** Ensuring data cleanliness and accurate labelling is crucial. To mitigate risks related to data quality, rigorous data preprocessing steps should be implemented. This includes data validation, cleaning, and transformation processes to ensure the dataset is ready for model training.

- **Software Bugs:** Bugs in the software can disrupt operations. Mitigation involves thorough testing and validation. Implement unit tests, integration tests, and user acceptance tests (UAT) to catch and resolve bugs before deployment.
 - **User Adoption:** Users might find the software challenging to use. Provide comprehensive training and support to mitigate this risk. Develop user-friendly documentation, tutorials, and offer webinars to help users navigate the software efficiently.
 - **Scope Creep:** Additional features or changes requested during the project could expand scope beyond initial estimates, impacting timelines and resources. Proposed mitigation would be to maintain a change management process to evaluate and prioritize new requirements, conduct impact assessments, and renegotiate project scope if necessary.
 - **Integration Complexity:** Challenges in integrating different software components (e.g., Django backend with React.js frontend, Power BI with data sources) could delay project timelines. Proposed mitigation would be to conduct thorough testing of integrations during development sprints, ensure compatibility of APIs, and have backup plans for data migration and synchronization.
-

7. Communication Plan

Below is our planned communication plan and how the team will collaborate during the project life cycle.

- **Kick-off Webinar:** Scheduled for 2024-05-13
 - **Weekly Meetings:** Regular updates and progress reviews with the project team.
 - **Monthly Reports:** Detailed progress reports to stakeholders.
 - **Feedback Sessions:** Regular sessions with users and stakeholders for feedback and improvements.
 - **Documentation:** Maintain up-to-date documentation for all aspects of the project.
-

8. Budget and Resources

Below are the current proposed perspectives on the budget and resource allocations. This can deviate as the project endures.

- **Personnel:** Salaries for data scientists, software developers, and project manager.
- **Software:** Tools for data preprocessing, model training, and software development.
- **Hardware:** Computers and servers for model training and software deployment.
- **Miscellaneous:** Training materials, documentation, and contingency funds.

Resource Capacity planning diagram:

The below table illustrates how we will accomplish resource allocation over time, showing how resources are distributed across different phases or tasks of the project.

Team Matrix					
Team Member	Programmers Team	Designers Team	Planners Team	Project Managers Team	DBs Dev/Admins Team
Elias	Reserves	Reserves	Main Team	Main Team	Reserves
Reon	Reserves	Main Team	Main Team	Main Team	Reserves
Johan	Main Team	Reserves	Reserves	Reserves	Main Team
Thabiso	Reserves	Main Team	Main Team	Main Team	Reserves
Ajay	Reserves	Main Team	Main Team	Reserves	Reserves
Jamie	Main Team	Reserves	Reserves	Reserves	Main Team
Tshepo	Reserves	Main Team	Reserves	Main Team	Reserves
Andre	Main Team	Reserves	Reserves	Reserves	Main Team
Reino	Main Team	Reserves	Reserves	Reserves	Main Team

9. Recommended Software Tools

These tools encompass the comprehensive infrastructure for data collection, management, software development, presentation, reporting, and project management, enabling a robust framework for the project's execution.

- **Data Collection:**
 - **Google Forms:** For gathering initial data.
 - **Microsoft Excel:** For organizing and managing collected data.
- **Data Management:**
 - **Power BI:** For data visualization and analysis.
 - **Microsoft SQL Server:** For storing and querying large datasets.
- **Software Development:**
 - **Visual Studio Code:** For coding and developing the software application.
 - **PHP:** For API integration, reading the dataset and backend development.
 - **JavaScript, CSS & HTML:** For Coding the front end of the web application.
- **Presentation and Reporting:**
 - **Microsoft PowerPoint:** For creating presentations.
 - **Microsoft Word:** For documentation and reporting.
- **Project Management:**
 - **Microsoft Teams:** For managing Tasks/ Milestones/Collaboration and Project as a whole.

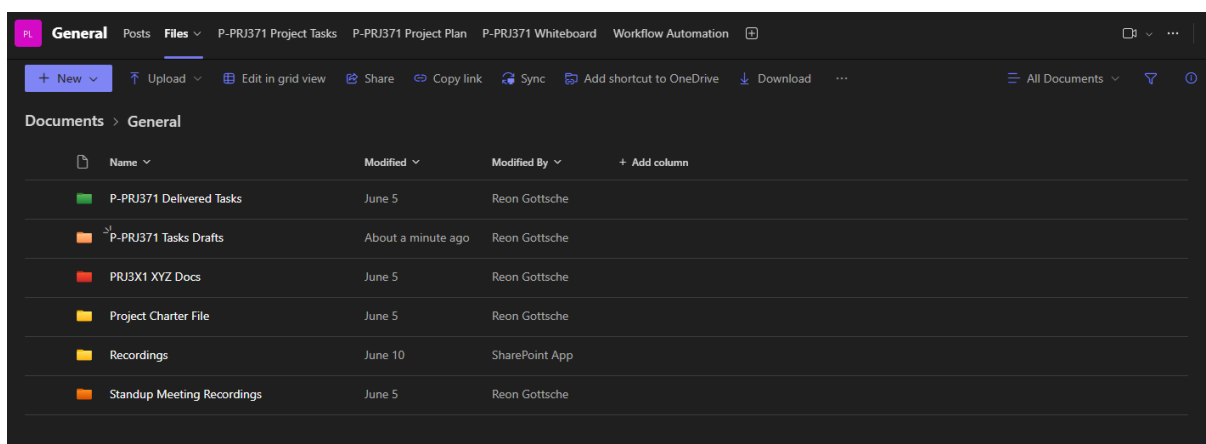
10. Success Criteria

These goals highlight the project's focus on achieving high user satisfaction through intuitive usability, ensuring timely delivery within budget constraints, and fostering continuous improvement through iterative enhancements and user feedback.

- **User Satisfaction:** Positive feedback from users on the software's ease of use and functionality.
- **Timely Delivery:** Completing the project within the defined timeline and budget.
- **Continuous Improvement:** Ongoing updates and improvements based on user feedback and new data.

11. Documentation/FSD Prep:

All Documentation will be stored in a central repository for the ongoing project on the Teams platform to enable knowledge sharing and to ensure if any resources are re-allocated or assigned that the team experiences minimal disruption and are enabled to provide seamless handovers from and between the various teams.



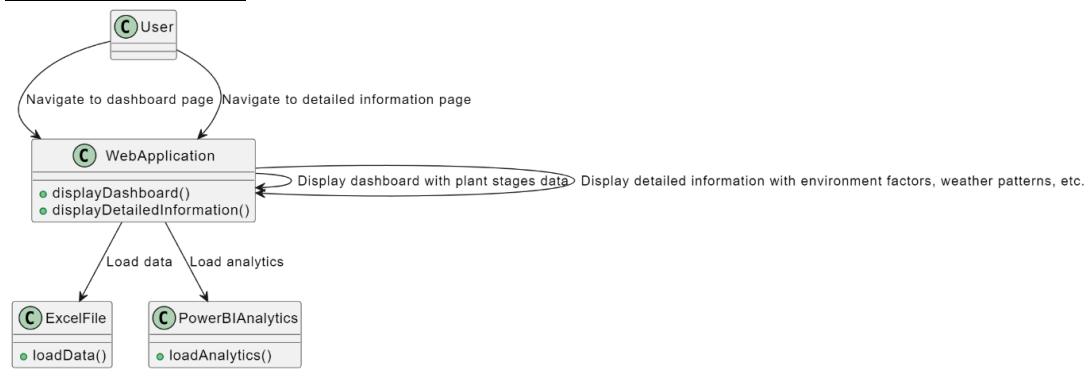
12. Class Diagram and High-level overview of user flow of the Proposed Solution

Below we have three high level diagrams to provide a comprehensive view of the application. Together these diagrams will give a clear understanding of the system architecture, the user experience, and the underlying objects of the system.

- **The Architectural Flow Diagram** clarifies the how the interaction will take place between different components of the system.
- **The User flow diagram** maps out the journey and interaction from the user perspective.
- **The Class Diagram** defines the applications building blocks.

These diagrams will help in improving future iterations of development as well as document the starting point of the development process.

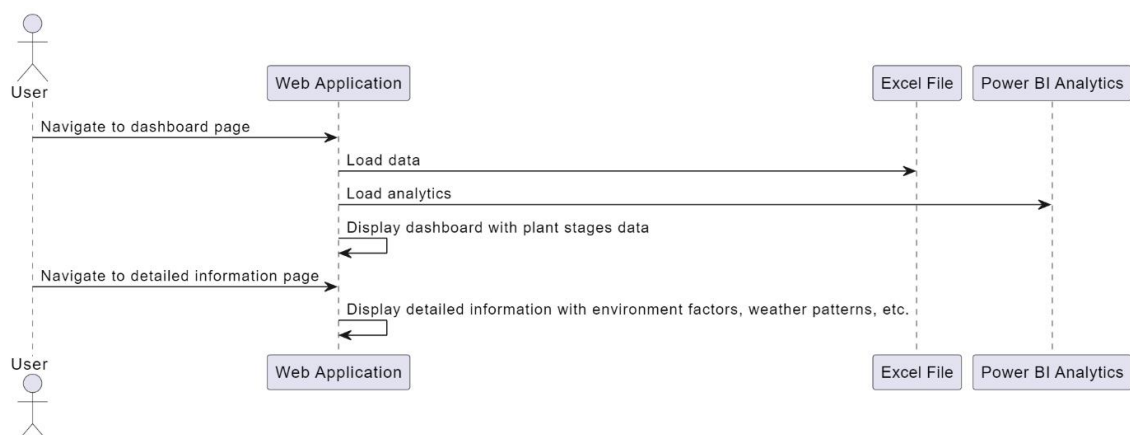
Architectural Flow



Class Diagram



User Flow Diagram



13. Proposed Technical Direction & Details:

Below is the planned Technology Stack the team will utilize to develop the solution. Additionally, Each Technology is explained for its specific use case in terms of the current project.

1. Technology Stack:

	Python	.NET (C#)	JavaScript and PHP
Reading Excel	Pandas	The team can use ClosedXML, which must then be mapped to a model.	Use PHP to read Excel or convert to DB.
Data manipulation	Pandas	Once the model is created, the data manipulation is easy.	Use models to manipulate data.
Creating graphs	Matplotlib	Chart.js	Chart.js
Creating a map with data	Geopandas	Leaflet or Google Maps	Leaflet or Google Maps
Consuming API	Django (hard to learn)	Swagger	JavaScript and PHP
Creating web application	Django	<ul style="list-style-type: none"> • Razor pages • HTML, CSS, JavaScript • Angular • React 	HTML and CSS with Bootstrap
Upskilling required	A lot of upskilling	The team already knows .NET, JavaScript, HTML and CSS, so there are a few things to learn.	Some upskilling with PHP, but the team know everything else.

2.Explanation of Stack:

The team considered a few options for the technology stack, as seen in the table above. This is further explained below:

- 1. Reading Excel:** We plan to use PHP to read the Excel dataset.
- 2. Data Manipulation:** Combination of JavaScript and PHP will be used to manipulate the data in a coding format.
- 3. Creating Graphs:** Chart.js is a library available with prewritten code that the team can utilize to easier develop the solution.
- 4. Creating a Map with Data:** To visualize geographical data we will utilize Leaflet or Google Maps
- 5. Consuming APIs:** We will develop API integration with JavaScript or PHP
- 6. Creating Web Applications:** We will develop the web application front end with HTML, CSS, and Bootstrap
- 7. Upskilling Required:** Some team members need to upskill in PHP, but they are proficient with all other required technologies.

In the end, we decided to use PHP, JavaScript, HTML and CSS as our technology stack due to available skillset and familiarity.

Please refer to the green section of the Table.

14. Project Lifecycle Details:

Outlined below are the high level steps the project team will undertake during the project life cycle.

1. Dataset in Excel

1. Dataset Cleanup.
2. Usability Enhancements.
3. Data Validation.
4. Quality Assurance.

2. Using the Dataset in Power BI

1. Import the Excel File.
2. Data Preparation.
3. Data Visualization.
4. Dashboards and Reports.

3. Design of front end and Wireframing

1. Design low-fidelity with wireframing to show flow state.
2. Design high-fidelity with prototyping.
3. Iteration of designs as required by and from the Development team.

4. Development of the Web Application

1. Create all necessary PHP classes with their properties and methods.
2. Create HTML for views.
3. Create JAVASCRIPT files for interaction between front and backend.
4. Add CSS to the HTML such as classes etc. for styling.
5. Add read operations in PHP for data display from sources.

5. Integrate with weather API

1. Create a JavaScript file to connect to a weather API (e.g., OpenWeatherMap).
2. Sign up for a free API key from OpenWeatherMap.
3. Add code to fetch the relevant weather data from the API.

6. Testing of Solution

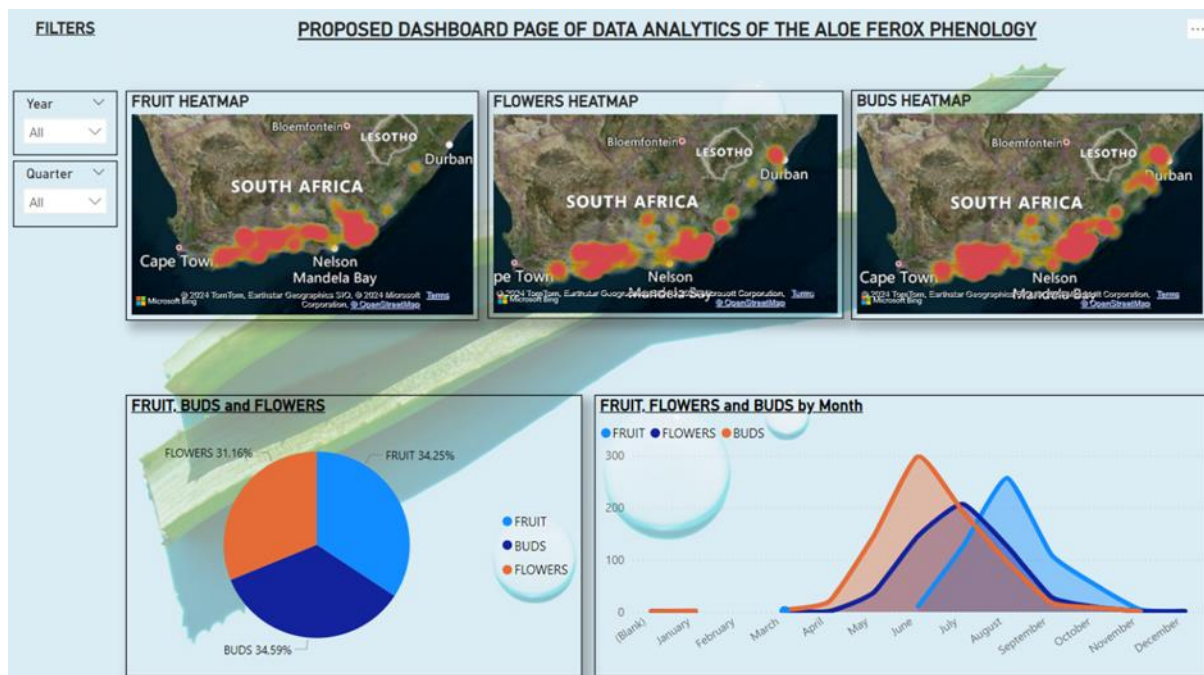
1. Ensuring that components of the software work as expected, such as the accuracy of the calculation and data transformation.
2. Confirm that different modules and services within the application work together seamlessly.

3. Validate the entire end-to-end system's functionality against the specified requirements, ensuring the system meets performance, usability, and security standards.
4. Validate that the system meets the needs and expectations of the end-users and stakeholders.
5. Assess the system's performance under various conditions and loads.

7. Stake Holder Demo

1. How we are working with the dataset
2. Treating the data on Power BI and making it presentable
3. Creating the necessary links to correlated data
4. Representing in a manner that will be of substance to the user by providing the necessary views
5. Getting those views onto a platform (website for the client's satisfaction)

Below is an example of these visualizations.



8. Deployment of Solution

1. Preparing code for deployment, including all final changes from design and stakeholder requirements
2. Configuring and transferring the code to a hosted server
3. Launch the solution and start system stress testing for feedback
4. Monitor the systems performance and implement technical changes as needed
5. Implement final rollout of codebase and go live

9. Training and User Acceptance

1. Document user guides detailing manuals that cover all features and functionalities.
2. Providing step by step instructions containing screenshots and examples.
3. Conduct online training using interactive tutorials and video explaining key functionalities.
4. Conduct live training sessions such as webinars and workshops for interactive learning.
5. Organize hands-on sessions demonstrating real-time usage of the platform/system.
6. Address user questions and queries through Q&A segments.

10. Monitoring/Maintenance Feature Creation.

1. Monitoring and identifying errors, bugs, and security issues
2. Prepare maintenance for the application and implement bug fixes and patches for the system
3. Enhance the system with newer technology over time and implement new features
4. Scale the application depending on the demand and business requirements
5. Additionally implement more training to users if there are more updates to the system

15. Additional Considerations for The Project Life Cycle:

1. **Validation of Assumptions:** Ensure all assumptions made in the project plan, such as data availability, resource capacities, and technology stack compatibility, are regularly validated and updated as needed.
2. **Contingency Planning:** While risks are identified, consider adding specific contingency plans for critical risks that could significantly impact project timelines or deliverables.
3. **User Feedback Integration:** Enhance the iterative development approach by explicitly outlining how user feedback will be collected, analysed, and integrated into the development cycles.
4. **Evaluation Metrics:** Define clear metrics for evaluating the success of each sprint and milestone, beyond just completion of tasks, to measure impact and progress towards project goals.
5. **Monitoring and Evaluation:** Expand on how you will monitor project performance and evaluate outcomes beyond the initial deployment phase to ensure continuous improvement.