Project Report

Chanutda Manosorn s5313405

Luke Edwards s5341152

Reem Aljorani s5341074

Satyam Sharma s5330583

Peer Review Scores

		From				Average
		Chanutda	Luke	Reem	Satyam	
To	Chanutda	7	10	10	10	9.25
	Luke	7	8	8.5	10	8.375
	Reem	8	9	8.5	10	8.875
	Satyam	7	9	8.5	9	8.375

Table of Contents

INDEX OF TABLES AND FIGURES

1.0 Executive Summary

1.1 Overview

The Forest Health project is dedicated to developing a comprehensive citizen science application aimed at monitoring and evaluating the impacts of bushfires on native forests and their subsequent recovery. This initiative is focused on creating a series of prototypes that will demonstrate the core functionalities of the application, ensuring it meets the diverse needs of its stakeholders.

The project's success hinges on a thorough requirements analysis, conducted by engaging with key stakeholders—including end users, research scientists, data analysts, and environmental agency representatives. By interviewing these stakeholders, essential information was gathered using User Stories and the FURPS framework to ensure that both functional and non-functional requirements are accurately captured and prioritized. This foundational work will guide the creation of diagrams, including WBS, PDM, Gantt chart, UML and prototypes for the Citizen Scientist Application that meet all specified requirements.

1.2 Stakeholders

In the Forest Health project, various stakeholders play critical roles in the successful development and implementation of the citizen science app. These stakeholders include both the internal project team, responsible for delivering the app, and external stakeholders, who either use or benefit from the app's functionalities. Below is an overview of the key stakeholders and their respective roles in the project.

1.2.1 Project Team

Project Manager: Chanutda Manosorn

Responsible for overseeing the entire project, ensuring that all milestones are met on time, and coordinating with different stakeholders. Chanutda works closely with all team members to ensure that process flow, time management, task management, and quality assurance align with project requirements.

Requirements Analyst**:** Reem Aljorani

Tasked with gathering and documenting project requirements, ensuring that the development team has clear guidance on what needs to be built. Reem also collaborates with Developers and the Project Manager to ensure that all deliverables meet the required quality standards through rigorous testing and validation processes.

Developer (Front End UI/UX)**:** Satyam Sharma

Responsible for developing the app's front-end UI/UX, ensuring that the final product meets functional requirements and provides a satisfying user experience.

Developer (Backend Systems)**:** Luke Edwards

Responsible for developing the app's backend systems, ensuring that the final product meets functional requirements and supports the overall application architecture.

1.2.2 External Stakeholders

End Users: Citizen scientists and hobbyists who collect data on flora and fauna diversity. They are the primary users of the app, responsible for uploading their observations and data.

Research Scientists: Individuals or teams who analyse the data collected by the end users for research purposes, particularly in the context of bushfire recovery.

Data Analysts: Professionals who focus on categorizing and compiling the data into specific groups, aiding research scientists without being directly involved in writing reports or drawing conclusions.

Environmental Agency Representatives: Responsible for ensuring that the data collection methods and app functionalities comply with legal standards in Australia and other regions where the app might be used.

1.3 Project Objectives

The primary objective of this project is to develop prototypes of the Citizen Science Application that will:

- Allow the public to collect and upload environmental data.
- Interact with the backend API to store and retrieve data.
- Provide a user-friendly interface for data entry and visualization.
- Demonstrate scalability and performance under expected load conditions.

1.4 Milestones and Time Estimation

The milestones and time estimations are based on the three sprints planned for this project. Each sprint is divided into specific weekly ranges as follows.

- ***Week 1-3***: Project planning and requirements analysis.
- ***Week 4-6***: Prototype development and initial testing.
- ***Week 7-9***: Finalize prototypes, complete testing, and prepare the final report.

1.5 Deliverables

The Forest Health project will produce the following key deliverables which each plays a crucial role in achieving the project's objectives.

- ***Citizen Science App Prototype***: A working prototype of the app that lets users upload and explore environmental data related to bushfires. The prototype will be available on both mobile devices (Android and iOS) and web browsers. It will include an API backend and a database, with administrative access provided via SQL for necessary tasks.
- ***Project Plan***: A detailed project plan that includes a Work Breakdown Structure (WBS), Precedence Diagramming Method (PDM), and Gantt chart. This plan will help guide the project's progress and ensure that key milestones are met.

- ***Requirements Analysis*:** Documentation of the project's requirements, supported by UML diagrams. This analysis will clarify what the system needs to do and ensure it aligns with what stakeholders expect.
- ***Prototypes*:** UI/UX designs and functional prototypes that show the main features of the app. These will be used to gather feedback and make improvements before final development.
- ***Test Plan*:** A test plan outlining the cases that will be used to check the functionality, performance, and scalability of the app. This plan will help ensure the app works as expected and can handle the required load.
- ***Software Metrics*:** Metrics will help measure how well the app meets its goals and identify any areas that might need improvement.
- ***Final Report*:** A report that pulls together all the project deliverables and includes a reflection on the project's outcomes. This report will document what was achieved and any lessons learned during the project.

2.0 Project Plan

2.1 Work Breakdown Structure

The following Work Breakdown Structure (WBS) outlines the key tasks and milestones for the development of the Citizen Scientist Application within the Forest Health project. The project is divided into 3 main sprints (a, b, and c), each focused on different aspects of the application development process, from initial (1) setup and requirement analysis to (2) frontend and (3) backend development, and finally, (4) full-stack integration and testing. The chart below visually represents tasks, making it easier to track progress and manage resources effectively throughout the project lifecycle.

The main structure is composed of 4 parts:

1. ***OS Setup and Project Planning*****:** Establishes the project's foundation by defining the scope, identifying and analysing requirements, and preparing the initial project plan.
2. ***Frontend Development*****:** Focuses on the design and development of the user interface and user experience (UI/UX), including wireframes, prototypes, and functional implementations.
3. ***Backend Development*****:** Covers the database design and implementation, integration with the frontend, and ensuring the functionality of backend systems.
4. ***Full Stack Integration, Testing, and Finalization*****:** Combines all elements from the previous sprints, involving comprehensive testing and the preparation of the final report.

The 3 sprints are organized as follows:

1. Sprint 1 (a) focuses on the planning and foundational elements.
2. Sprint 2 (b) is dedicated to frontend and backend development.

Sprint 3 (c) involves the integration, testing, and finalization of the project.

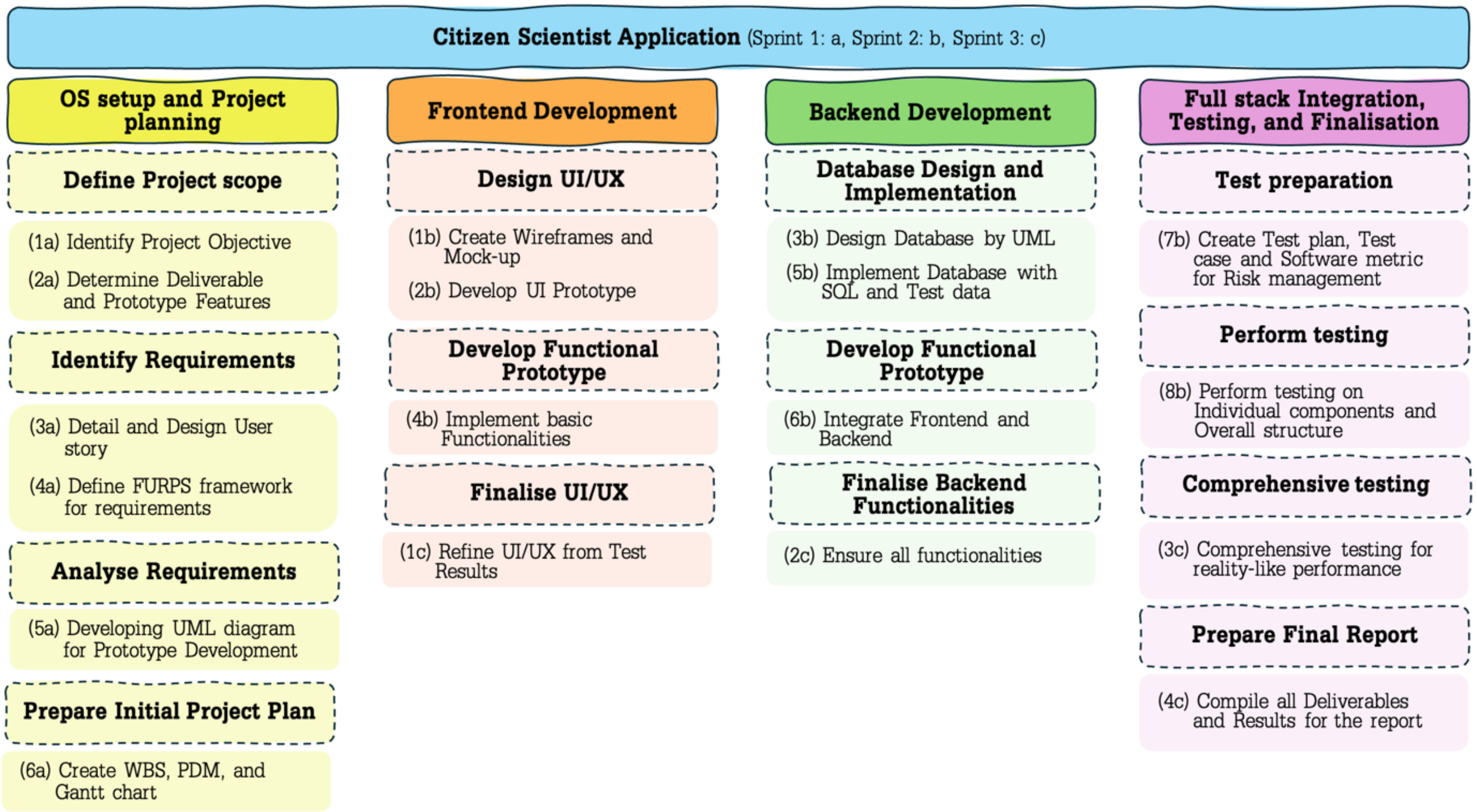

Figure 1 Work Breakdown Structure (WBS) for Citizen Scientist Application

Figure 1 presents a detailed Work Breakdown Structure (WBS) for the Citizen Scientist Application project. The project is divided into 3 main sprints, each focusing on different aspects of the development processes and activities.

2.2 Activity Definition & Estimation

2.2.1 Activity definition and estimation

Activity definition and estimation will be defined in each sprint to show scope, parallel task, dependency task, and due date for each task. The estimation is also given information to visualise Precedence Diagram Method (PDM) to analyse critical path analysis to focus.

Table 1 Activity definition and estimation for Citizen Scientist Application project.

Activity	Scope	Parallel task	Dependency task	Duration/Due date	Note
Sprint 1: Project Planning and Requirement analysis (a)					
1) Identify Project Objective	OS	-	-	4D/2nd August	-
2) Determine Deliverable Features	OS	3a	1a	2D/3rd August	-
3) Detail and Design User story	Full stack	2a	1a	4D/7th August	-
4) Define FURPS framework for requirements	Full stack	-	2a, 3a	4D/10th August	-
5) Developing UML Diagram for Prototype Development	Full stack	-	4a	7D/16th August	Split into 2 phases (1) Developing UML diagram including *Use case/Activity/ State* diagrams (2) Refine UML after requirements check
6) Create WBS, PDM, and Gantt chart	OS, Full stack	-	4a	7D/18th August	Steps (1) WBS (2) PDM (3) Gantt chart
Sprint 2: Prototype development and Initial testing (b)					
1) Create Wireframes and Mock-up	Frontend	-	5a	5D/20rd August	-
2) Develop UI prototype	Frontend	3b	1b	4D/23rd August	-
3) Design Database by UML	Backend	2b	1b	4D/23rd August	-
4) Implement basic Functionalities	Frontend	5b	2b	6D/29th August	-
5) Implement Database with SQL and Test data	Backend	4b	3b	6D/29th August	-
6) Integrate Frontend and Backend	Full stack	-	4b, 5b	5D/3rd September	-

Activity	Scope	Parallel task	Dependency task	Duration/Due date	Note
7) Create Test plan, Test case and Software metric for Risk management	OS, Full stack	2b, 3b	1b	10D/28th August	Quality control gets along all stages
8) Perform testing on Individual components and Overall structure	OS, Full stack	-	6b	4D/6th September	-
Sprint 3: Finalise Prototype, complete testing, and prepare the final report (c)					
1) Refine UI/UX from Test Results	Frontend	2c	8b	5D/10th September	-
2) Ensure all Functionalities	Full stack	1c	8b	4D/10th September	-
3) Comprehensive testing for reality-like performance	OS, Full stack	-	2c	5D/14th September	Performance and scalability check
4) Compile all Deliverables and Results for the report	OS, Full stack	-	3c	5D/18th September	Working on application and document

2.2.2 Critical Path Analysis

Critical Path Analysis will be conducted for the entire project from 1a to 4c, without separating the analysis by sprints. This approach ensures that the dependencies and durations of all activities are considered holistically, focusing on the overall project timeline rather than individual sprint timelines. The analysis will identify the longest path from start to finish, ensuring that any delays along this path are addressed promptly to keep the project on schedule. The interconnected stages, transitions between activities, and the completion of key tasks according to the planned milestones will be emphasized to ensure the timely delivery of the project.

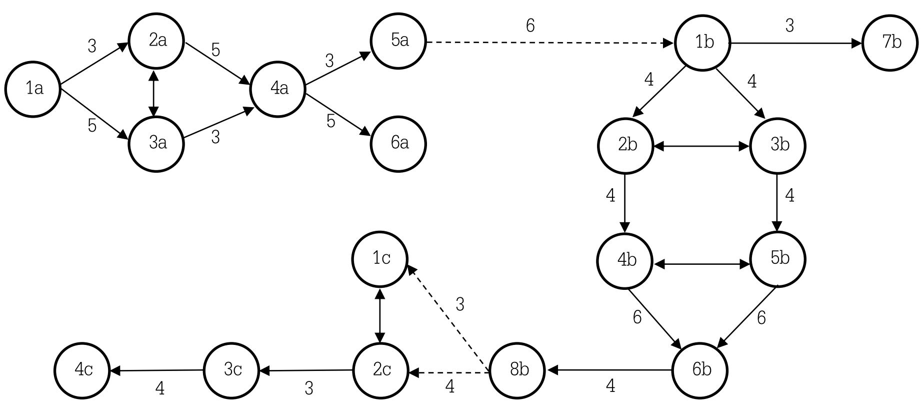

Figure 2 Precedence Diagram Method (PDM) shows the dependencies of activities and their durations, composed of 3 sprints (a, b, c), starting from 1a and finishing at 4c in a network diagram.

There are 16 distinct paths to determine the longest path from the starting point (1a, Sprint 1) to the final task (4c, Sprint 3). These paths represent the various sequences of tasks that must be completed across all sprints, from defining project objectives, requirements gathering, frontend and backend development, to full-stack integration and testing. By identifying the longest path, ensuring that critical tasks are managed to avoid project delays and timely completion of the project as shown in Table 2.

Table 2 Longest Paths and Durations in the Forest Health Project for developing Citizen Scientist Application prototype.

	Paths	**Duration (days)**
1	1a – 2a – 4a – 5a – 1b – 2b – 4b – 6b – 8b – 2c – 3c – 4c	3+5+3+6+4+4+6+4+4+3+4= 46
2	1a – 3a – 4a – 5a – 1b – 2b – 4b – 6b – 8b – 2c – 3c – 4c	5+3+3+6+4+4+6+4+4+3+4= 46
3	1a – 2a – 4a – 5a – 1b – 3b – 5b – 6b – 8b – 2c – 3c – 4c	3+5+3+6+4+4+6+4+4+3+4= 46
4	1a – 3a – 4a – 5a – 1b – 3b – 5b – 6b – 8b – 2c – 3c – 4c	5+3+3+6+4+4+6+4+4+3+4= 46
5	1a – 2a – 4a – 5a – 1b – 2b – 4b – 6b – 8b – 1c – 2c – 3c – 4c	3+5+3+6+4+4+6+4+3+0+3+4= 45
6	1a – 3a – 4a – 5a – 1b – 2b – 4b – 6b – 8b – 1c – 2c – 3c – 4c	5+3+3+6+4+4+6+4+3+0+3+4= 45
7	1a – 2a – 4a – 5a – 1b – 3b – 5b – 6b – 8b – 1c – 2c – 3c – 4c	3+5+3+6+4+4+6+4+3+0+3+4= 45
8	1a – 3a – 4a – 5a – 1b – 3b – 5b – 6b – 8b – 1c – 2c – 3c – 4c	5+3+3+6+4+4+6+4+3+0+3+4= 45
9	1a – 2a – 3a – 4a – 5a – 1b – 2b – 4b – 6b – 8b – 2c – 3c – 4c	3+0+3+3+6+4+4+6+4+4+3+4= 44
10	1a – 3a – 2a – 4a – 5a – 1b – 2b – 4b – 6b – 8b – 2c – 3c – 4c	**5+0+5+3+6+4+4+6+4+4+3+4= 48**
11	1a – 2a – 3a – 4a – 5a – 1b – 3b – 5b – 6b – 8b – 2c – 3c – 4c	3+0+3+3+6+4+4+6+4+4+3+4= 44
12	1a – 3a – 2a – 4a – 5a – 1b – 3b – 5b – 6b – 8b – 2c – 3c – 4c	**5+0+5+3+6+4+4+6+4+4+3+4= 48**
13	1a – 2a – 3a – 4a – 5a – 1b – 2b – 4b – 6b – 8b – 1c – 2c – 3c – 4c	3+0+3+3+6+4+4+6+4+3+0+3+4= 43
14	1a – 3a – 2a – 4a – 5a – 1b – 2b – 4b – 6b – 8b – 1c – 2c – 3c – 4c	5+0+5+3+6+4+4+6+4+3+0+3+4= 47
15	1a – 2a – 3a – 4a – 5a – 1b – 3b – 5b – 6b – 8b – 1c – 2c – 3c – 4c	3+0+3+3+6+4+4+6+4+3+0+3+4= 43
16	1a – 3a – 2a – 4a – 5a – 1b – 3b – 5b – 6b – 8b – 1c – 2c – 3c – 4c	5+0+5+3+6+4+4+6+4+3+0+3+4= 47

In conclusion, Paths 10 and 12 are identified as the Critical Paths because they have the longest duration of 48 days. This means that any delay in the tasks along Paths 10 and 12 will delay the completion of the entire project. However, it is crucial to account for the working duration of each activity and monitor transitions between sprints closely. Ensuring that milestones and due dates are met within each sprint is essential, as delays in milestones could significantly impact the overall project timeline. To mitigate these risks, continuous tracking and proactive adjustments will be employed throughout the project lifecycle.

2.2 Gantt Chart

The Gantt chart in figure 3 for the Citizen Scientist Application project is structured into three distinct sprints, each representing a critical phase of the project lifecycle:

1. ***Sprint 1***: Project Planning and Requirement Analysis
2. ***Sprint 2***: Prototype Development and Initial Testing
3. ***Sprint 3***: Finalize Prototypes, Complete Testing, and Prepare the Final Report

Each sprint is further broken down into smaller stages, which are color-coded to differentiate between various tasks within each sprint. The Gantt chart clearly delineates task dependencies, which have been meticulously derived from the Precedence Diagram Method (PDM). Additionally, the chart incorporates supplemental activities, such as meeting setups, which have been integrated based on the insights gathered from the Activity Definition and Estimation phase.

Overall, this Gantt chart provides a comprehensive visual representation of the project's workflow, supporting effective project management and ensuring that all key activities are completed within the scheduled timeframe.

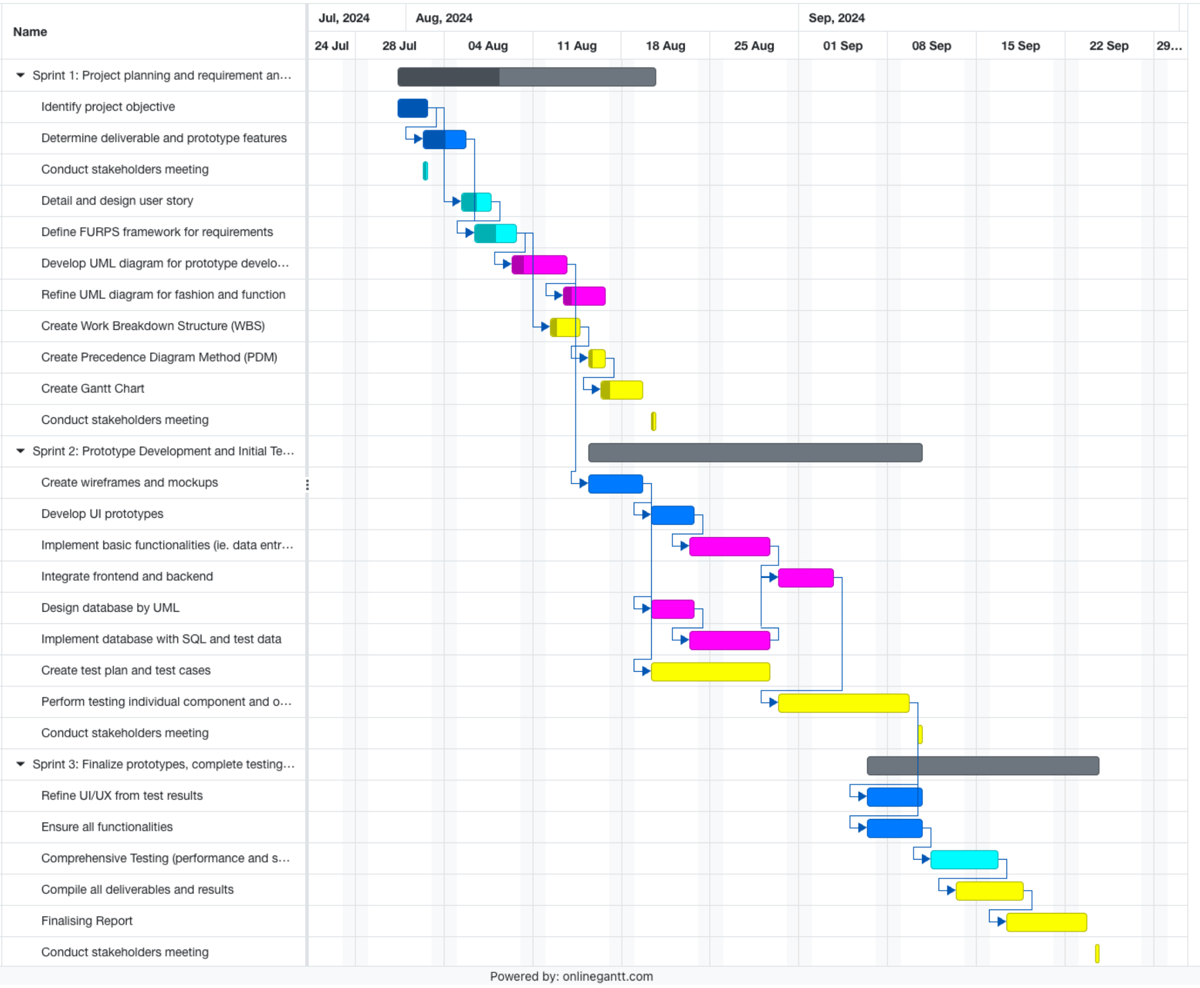

Figure 3 Gantt chart for Citizen Scientist Application project

Downloadable version: https://1drv.ms/u/s!AqAw8dgYqOqbgohTr-qiAs63aA_-Cw?e=f7deXY
Open file.gantt via: https://www.onlinegantt.com/#/gantt

3.0 Requirement Analysis

3.1 Requirements Specification

The requirements of the Forest Health's citizen science project can be analysed by investigating the key stakeholders and their needs whilst using the platform. Each stakeholder should have different requirements and thus all bring unique ideas we should implement whether it is the features of the mobile platform or different analysis access via the API. To accurately capture these requirements, different stakeholders will be interviewed with a quick brief of the project, which we will then undertake necessary feedback of their expectations with the final product. By engaging with our peers over the duration of the project, a clear prototype focusing on both functional and non-functional requirements will hasten final design completion, where thorough planning at the beginning will lessen more work later in the timeline. Separating the requirements will be conducted via User stories and categorising in the FURPS framework, ensuring all feedback collected can be addressed and easily identifiable in an organised manner.

3.1.1 Different Stakeholders Identified

The key stakeholders of the Forest Health project can be identified as the following, with brief descriptions of their role, which is important in our future development.

- ***End Users*****:** Citizen Scientists/hobbyists who aim to collect data on flora and fauna diversity in a detailed manner and upload their findings for future research. The main stakeholder in project.
- ***Research Scientists*****:** Individuals or a team who aim to analyse the collected data for specific research in responding to claims or finding solutions for efficient bush fire recovery.
- ***Data Analysts*****:** Separate to research scientists, focus on analysing the data collected and compiling into specific groups without extending on writing reports and gathering conclusions.
- ***Development Team*****:** Responsible for app and database development, utilizing requirements to provide final products that perform well and fit Forest Health's standards.
- ***Environmental Agency Representatives*****:** Individuals aiming for data collected and the methods used for collection to be compliant with Australia's legal standards, and other countries and/or states the app may broaden to.

3.1.2 Requirements of the Stakeholders

The requirements of the stakeholders presented prior in Section 3.1.2 were investigated through interviews with peers as end users and other individuals around us who may fit in different categories when interacting with the program. These categories further separate into developers, those responsible for developing the application and database, legal environmental enthusiasts requesting the program to be legally legible and thus be able to lead to feasible research conclusions, individuals who aim to use the database for analysis as well as others potentially using analysis to support research, and lastly the end users who will collate the data and thus mainly interact with the application. All requirements will be shortened into a brief description to show the difference in our project's different stakeholders, and their full individual responses gathered in the interviews will be listed and arranged in Section 3.1.3.

- ***End Users***: End users responded that they require the application to offer an efficient interface with clear navigation and user-friendly features such as San Serif fonts and non-clashing colour schemes; also desiring a range of data types collected to be easily upload-able. Visual and audible aid should be included in the user-friendly features, allowing those with possible impairments or additional care to be supported in the program. Mobile service shouldn't be essential for the app to function as not all geographical locations will have service, data collected shouldn't be lost and sync automatically when device is reconnected.
- ***Research Scientists***: The research team requires many tools within the database to aid in efficient analysis of data collected by the end users, such as in-app graph construction. May include advanced search and filtering options and ability to export data to their own devices and/or additional apps. Also, researchers should be able to track specific data such as a certain geographical location or tree species that may aid in evaluating bush fire effects and thus aid in forming strategies for renewal based on these findings. Researchers working in groups also require access to each other's data, if given access to help collate their findings. Online forums could also be implemented as our testers mentioned having a place to ask questions and interact with other researchers, which can also apply to the end users.
- ***Data Analysts***: Data analysts require similar database features to the research team, such as filtering options based on location, time, etc, as well as the exporting to those requiring the analysts' findings. Additionally, the system should provide SQL queries and provide easy access to the database and allow analysts to select data and move to in-app folder locations in a reliable manner where app movement syncs.
- ***Development Team***: The development team require both the database and application architecture to be scalable and high quality, allowing for future enhancements and integration to not be complex and complacent with Forest Health's existing systems and needs. The project should adhere to over 500 data requests per minute, as well as limit data loss when app users lose service; ensuring app can accommodate to future growth in increased user traffic and data volume, thus being scalable. The software metrics should be clean and have little to no bugs or gateways to cyber-attacks, adhering to industry and legal standards. Finally, there is a main requirement for the project prototypes to be regularly tested and updated based on findings that require integration to enhance the project.
- ***Environmental Agency Representatives***: Environmental representatives require the project for Forest Health to only include features that lead to data and research findings that can be accurately and reliably used before submission. All legal standards of the law should be adhered to and there should be no motivation for citizen scientists to break any laws such as trespassing or damaging and/or transporting flora and fauna, to collect any data. Also, the safety of users while collecting data should be encouraged, and end users should have a warning to not touch any plants while collecting data.

3.1.3 User Stories and FURPS Framework

To effectively gather and organise the requirements of stakeholders involved in the Forest Health application, the cooperative use of both User Stories and FURPS framework were conducted. User stories were developed based on the needs and expectations of our tested interviewers, who helped specify which stakeholders were involved in the program and the requirements the final product should address.

Each user story, which was organised in Table 3, represents each stakeholder’s perspective when answering the question “What is your interaction with Forest Health’s citizen scientist project, and what features/ design are you expecting?” For those unaware of what a citizen scientist is, they were briefed prior to gathering their response. Further, the table includes an ‘F’ positioned after certain user stories that were seen as a functional requirement, which is a requirement that developers will need to implement into the product features or functions. This is important as many of the stakeholders are interacting directly with the project and if the app isn’t suiting their needs or is missing needed features, the project usage levels will diminish significantly. All other user stories missing an ‘F’ are the non-functional requirements, judging the operation of the project rather than specific behaviours.

The FURPS framework consists of different groups used to categorise the gathered requirements, being functionality, usability, reliability, performance, and supportability. This organisation tool ensures the functional and non-functional requirements can be separated, and that all critical needs are identified, prioritized, and systematically addressed in prototype development process. Laying out these needs in Table 4 helps display the organised requirements for the developers so that the vital data collected by our interviewed stakeholders are not missed. A unique ID for each requirement is given and can be used further in the report when discussing certain project implementations and which requirements were considered.

Table 3 User Stories for corresponding Stakeholder groups

As a/an <type of stakeholder>	**I want to/the application <goal/objective>**	**So that <benefit/result>**
End Users	to be user-friendly and easy to navigate	I can efficiently collect and upload different types of data. (F)
	to be functionable offline and sync data automatically when there is service connection restored	I can use it in remote locations. (F)
	use as a guest without logging in	I can quickly access its features when I don't want to upload anything.
	to handle a growing number of data applications without a decrease in performance	I can keep using the app smoothly and not run out of storage.
	to not have too many flashy components as a user in the older generation	I am able to easily navigate through the app without help.
Research Scientists	to export analysed data and use advanced search and filtering	I can conduct detailed research to generate reports based on findings. (F)
	o track specific data and collaborate with other researchers	I can collect a range of flora and fauna data used to evaluate bushfire effects and develop strategies for recovery in specific areas. (F)
Data Analysts	system to offer various data visualization tools and filtering options	I can interpret trends and insights from the collected data. (F)
	system to provide SQL query capabilities and easy access to data for analysis	I can well as store various analysed data groups safely. (F)
	data I've analysed and grouped to be secure	I can keep continuing to update and/or utilise my findings.
Development Team	application and database architecture to be scalable and modular.	future enhancements and integrations are manageable such as extending number of requests per minute and catering to a larger audience of potential citizen scientists. (F)
	codebase to be well-written with minimal bugs as well as adhering to cyber-security industry standards	application is secure and maintainable over time.
	To be able to implement updates smoothly	future adaptations do not affect the program in a way that reduces uptime performance.
Environmental Agency Representatives	to ensure data used in research is accurate and reliability by meeting legal standards	data stored in database to be used by research scientists and data analysts is valid and feasible and can't lead to legal trouble.
	include features that prevent illegal activities and promote user safety	data collection process is ethical, safe, and secure.

Table 4 FURPS framework separating User Stories' requirements

FURPS	**Unique requirement ID**	**Requirements**
Functionality	F1	The application must function offline, with automatic and reliable data synchronization once the connection is restored, without data loss or corruption.
	F2	The application should allow data export, advanced search, and filtering for detailed research and report generation.
	F3	The system must provide advanced SQL query capabilities with efficient indexing and secure storage of both raw and analysed data.
	F4	The application should include features to prevent illegal activities and ensure ethical and secure collection of data.
Usability	U1	The application should have an intuitive UI/UX design that facilitates easy navigation and quick access to all features for all user demographics.
	U2	The system should provide data visualization tools and filtering options for trend interpretation.
	U3	Users should be able to access the app as guests without logging in.
Reliability	R1	Analysed and grouped data should be stored securely, with regular backups and version control to prevent data loss or unauthorized access.
	R2	Updates should be implementable without negatively affecting program performance or uptime.
	R3	The application must ensure data accuracy and integrity, complying with relevant legal and ethical standards, and offering traceability of data sources.
Performance	P1	The software should be modular, allowing easy integration of new features and efficient handling of increased user traffic and data requests.
	P2	The application must be capable of scaling to handle increasing amounts of data, ensuring it remains responsive and performant under high loads.
Supportability	S1	The system should enable tracking and monitoring of specific datasets over time, with features supporting real-time collaboration between multiple researchers.
	S2	The codebase should be well-written, bug free, secure, and maintainable, adhering to industry standards as well as utilising algorithms to support fast searching and sorting of data.

3.1.4 Conclusion of Section

In summary, a requirement analysis of the Forest Health’s citizen scientist project has provided a detailed understanding of the various stakeholders involved and their needs and expectations which will be implemented in ongoing project prototypes. By conducting interviews, these requirements were recorded and organised in the analysis tools User Stories and FURPS framework, with a brief description seen in Section 3.1.2 concisely displaying our findings. These tools ensured both functional and non-functional requirements can be defined and prioritized as both are highly needed in developing a successful and user-friendly final product. Overall, a solid foundation has been made for the next step of development and future enhancements of the application, which will be able to support efficient data collection, analysis, and research whilst adhering to legal and ethical standards of all regions the app is accessible in.

3.2 Use Case Diagram

In this section, the significant requirements of the Forest Health citizen science platform discussed prior will be illustrated using a Use Case Diagram. This diagram is used to highlight the many different communications between the system and its stakeholders/actors, including citizen scientists (end users), research scientists, data analysts, environmental agency representatives, etc. Each stakeholder plays a unique role in contributing to or benefiting from the platform's functionality, such as collecting environmental data, analysing post-bushfire recovery, or ensuring compliance with legal standards. The diagram which will be seen in Figure 4, helps to visualize the relationships between the actors (stakeholders) and the key system requirements discusses in Section 3.1, that are required to meet stakeholder expectations in this project. This may be the pathway of a data analyst, ensuring that one of their requirements such as filtering data is included in database via a separate analysis section. Further, it helps show which entry conditions are required to enter a different function as well as the requirements, such as authenticating after login before accessing the database or data collection services. The diagram will be updated accordingly per sprint stage to ensure use case corresponds to the new prototypes developed until the final project.

After the Use Case Model has been uploaded, we will describe one of the use cases in detail, by explaining the specific interaction between that user and the system, as well as the expected outcome for a particular scenario, such as collecting data or accessing the database.

3.2.1 Use Case Diagram

The Use Case Diagram is represented below in Figure 4 where the functionalities and pathways of four main stakeholders/actors is shown.

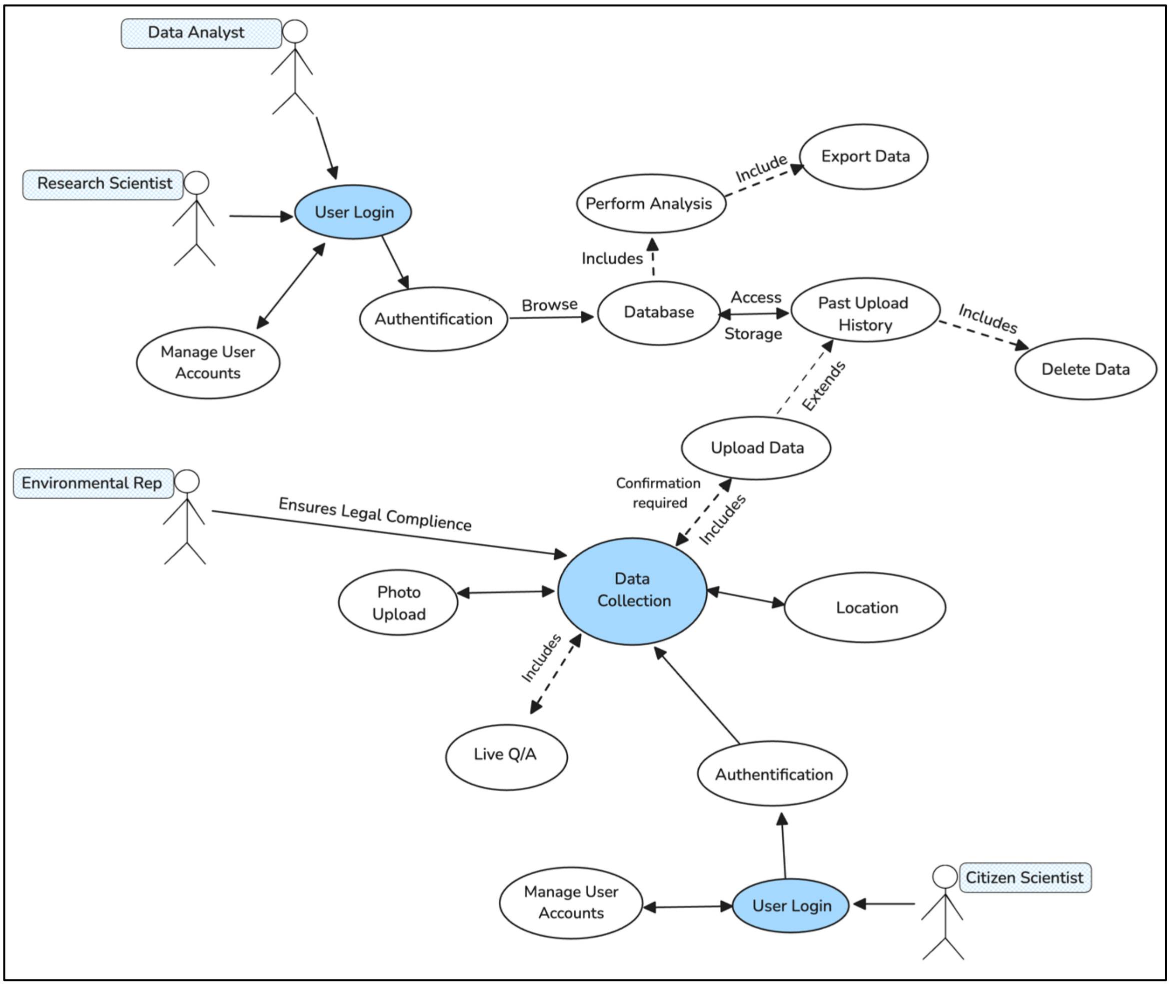

Figure 4 Use Case Diagram

The diagram describes the main Forest Health platform implementations for the first UI/UX prototype whilst illustrating the main interactions between the different stakeholders. The main stakeholders involved, known as actors in this diagram, are Citizen Scientists, Research Scientists, Data Analysts, and Environmental Representatives. By analysing the requirements of the actors via User Stories and FURPS model in Section 3.1, we were able to gather the various system features necessary to ensure the final product meets user expectations. The pathways of each main stakeholder involved in interacting with the platform will be discussed, as well as the main features that the project plans to implement based on their certain requirements.

- ***Citizen Scientist***: These are the End Users, whom are primarily interacting with the system database via the API and are the main stakeholder. Initially, they must access the system by logging in through the "User Login" functionality, which includes an authentication stage immediately after. The method of authentication may be a code sent to the user's registered email or phone number which are also self-manageable. Once approved, these users will be able to begin collecting and uploading data essential for bushfire renewal studies which can be done under "Data Collection", performing features such as uploading photos, submitting location, and submitting other key finds such as vegetation type, landscape, and more. They will also be able to participate in Q&A sessions via a live forum with other citizen scientists in the community, and after uploading data, access their "Past Upload History" and have ability to delete any. These features are highlighted in the Use Case Model; however, it is important to note that backend requirements such as data sync and handling high amount of amount will be needed also.

- ***Research Scientist***: These users need to interact with the database via SQL queries and will be conducted after the primary "User Login" and "Authentication" stages. They will be able to browse the database using SQL requests, accessing all the stored data collected by the citizen scientist's successful uploads. By accessing the "Database", these users also have access to the "Perform Analysis" function which is a necessary requirement to perform various tools to gather the data. These tools may be and are not limited to; creating graphs, sorting by factors such as location, and more. Finally, the last key feature shown in the Use Case Model is ability to exporting data if needed.
- ***Data Analyst***: These users have very similar features to the research scientists, also being able to access the database after logging into the system and successful authentication. Whilst the research scientists can also analyse data, this stakeholder is expected to utilize this feature more and export data at larger amounts for further use, a backend feature mandatory will be security on their analysed data and ability to update it.
- ***Environmental Representative***: This is the final user shown in the model, with a key goal of ensuring collected data complies with legal standards of the environment being analysed. Spoken in Section 3.1, they involve overseeing the data collection process is safe and that the public citizen scientists are not breaking any laws when collecting data. This helps authenticate the data that will be used by research scientists further on, to ensure all is feasible and valid.

Overall, the system's central functionality revolves around "Data Collection" which ties up the multiple processes of each different stakeholder. By doing this, it is easy to display how each stakeholder will interact with the system and use similar functionalities to reach their app goals. Further, scalability was kept in mind so that when the future prototypes are developing, we can easily implement the new functions and features into the model accordingly to enhance our overall knowledge of the project being created.

3.2.2 Detailed Use Case

Detailed Use Case ID: DUC-1 – data collection sample

Actor(s): Citizen Scientist (End User), Environmental Representatives

The "Data Collection" use case involves the process of our main app end users (Citizen Scientists) collecting and uploading environmental data (such as photos, location, etc) to the Forest Health platform database. The data collected should be first reviewed by the environmental representatives' requirements, using a background system that automatically flags selected data not fitting legal standards, as well as a pop-up message before collection explaining that all scientists should follow safe and legal processes. This procedure uses the Use Case Model discussed prior and any pathways not applicable will be seen as an "error condition".

Pre-conditions:

- The citizen scientist has access to a mobile device with an applicable system so that they can upload/view data and user other key features.

- The citizen scientist has successfully logged into the system using the “User Login” use case and been authenticated accordingly.
- The user can access data collection and upload forms without bugs interrupting the process.

Post-conditions:

- The data has successfully passed environmental legal standards review.
- All data collected per entry including images, location, and other environmental details, is successfully uploaded onto the system back-end database and user upload history.
- The data is accessible to its key users such as research scientists and data analysts via the database.

Main Success Scenario/Flow of Events:

1. The Citizen Scientist (End User) logs into the platform on their device through the **User Login** feature and is successfully authenticated.
2. On the main screen, they are able to navigate the easily found **Data Collection** section.
3. The user selects an option to upload a photo supporting their data collection and input it along with the corresponding location and other details to fulfill minimum requirements.
4. The system prompts user for confirmation before finalizing the submission.
5. If submitted successfully, system stores the data in the database and the user’s personal data collection history and logs the upload.
6. An automatic notification is sent to the **Environmental Representative’s legal standards section**, indicating the new data is available for a computer-based review.
7. The **computer** accesses the data, reviews the uploaded information based on given set standards, and confirms data collected meets required legal and compliance requirements.
8. Once approved and saved, data is now made accessible to **Research Scientists** and **Data Analysts** via the database for any further analysis.

Error Conditions and Alternative Pathways:

1. Failed upload due to unfulfilled data requirements:
    - If citizen scientist fails to provide all minimum data requirements necessary for a full upload such as no location, date, etc, the system will prompt user to complete missing fields.
    - Once data is now entered, the user can then successfully re-upload.
2. Data doesn’t meet legal standards:
    - If the environmental representative’s legal standards are not met in an uploaded data submission, it will be flagged for further review and a summary of why it doesn’t comply with the necessary successful requirements.
    - System notifies the citizen scientist via email or phone to modify the submission so it fits standards or delete the data if it cannot be recovered.
    - These can also be seen under the “Past Upload History” section however user’s still need to be notified so are able to review it fast.
    - All invalid data will be available for modification up to 2 weeks after notification, otherwise system automatically deletes to reduce traffic flow and unnecessary storage.

This “Data Collection” detailed Use Case is a key part of the Forest Health application, as the main end users “Citizen Scientists” are the key collaborators of the system. Their collected data play a key part in allowing further analysis and research by other relevant stakeholders regarding bushfire recovery. By ensuring legal compliance is met prior to allowing these stakeholders to analyse data, it ensures they can immediately use it in their reports without being held back by integrity and reliability issues. Ensuring pre and post conditions are met, allowing alternative pathways if an error was to occur during data submission allows the platform to be fully usable over-time as well as involve scalability where developers will be able to continuously work on updates to keep providing bug-free, successful routes. Conclusively, although there are many Use Cases that can be analysed, this is the main one that forces the platform to be fully complicit for all stakeholders involved.

3.3 Activity Diagram

A UML activity diagram is a visual representation on how the application works and progresses, capturing the flow of data which makes it simpler for the stakeholders to understand about the working of application. Further, it helps developers visualize about the working of the app itself, which makes it more efficient to spot inefficiencies errors, or any areas of improvements. This helps the developer to has a clear understanding of what are the expectations regarding the app itself and its working. Doing these checks early in planning stage avoids additional work, costs, and resources later when needing to fix an issue involving a lot of code and work. Additionally, by planning out the flow pathways of the system and discovering certain end points as well as the different forks, the app can be built as scalable earlier on, with an advantage of discovering the best system locations for updates and refinement based on the later testing stage.

The major stakeholders chosen from the requirements analysis stage and included in this UML are:

1. ***System users***: Citizen Scientist or citizens that may interact with the app itself.
2. ***Product manager***: The people who have holdings for the app from financial POV.
3. ***Developer***: The software design and production of the app.
4. ***Testers:*** The testing the app itself after the development.

The activity diagram shown below displays a management and flow of operation within the system, focusing majorly on the data collection, here's a breakdown of the diagram shown:

1. ***Start:*** The operation starts when one of the users Interact with the app by opening the application.
2. ***User Login***: A user login is asked upon opening the application, that includes a unique user id and password.
3. ***Authenticator***: The screen then is transferred to an authenticator and would require an authentication software such as authenticator or ping id.
4. ***Decision***:
    a. Authentications succeed: If auth is success then it takes to the next step that is uploading data.
    b. Authentications denied: If the auth is denied, user can't get the access and needs to try again with the authenticator, or a recovery key provided when first time authenticator was set up.

5. ***Upload Data***: The data is uploaded by the user which may include different kind of information regarding different land masses or flora and fauna.

6. ***Data validation:*** The system checks and validates if the data is uploaded correctly to the database or no, of yest it proceeds to the next step, if not it goes the previous step that is, upload again.

7. ***Store Data***: The data is stored either locally or is *uploaded to the database* and provided to Citizen scientist.

8. ***Perform analysis***: This data uploaded is then analysed by the citizen scientist for their specific field of work.

9. ***Exit***: The software or the application then can be excited by the user and citizen Scientist.

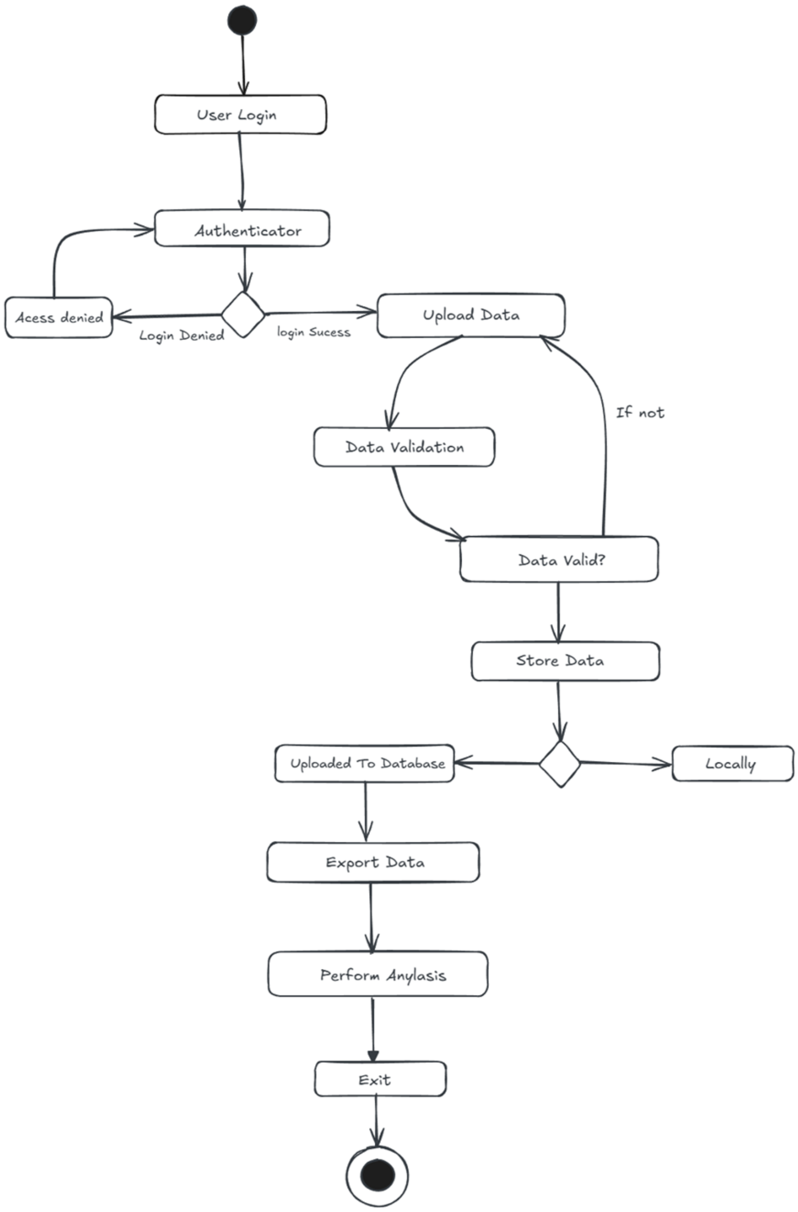

Figure 5 UML activity diagram

3.4 State Diagram

The state diagram shown below displays the flow of states different types of users have and how they would interact with the system the types of users shown here are:

1. ***End user***: this is the public users which are allowed to upload the data.
2. ***Environmental representative***: this user can approve the data and ensure that the data is relevant and is what the program is looking for.
3. ***Research scientist***: the research scientists are allowed to access the database to analyse the data and generate their reports.
4. ***Data analyst***: data analysis can access the database to categorize it and make the databases look like good for the analysis.

For each of these users the diagram shows the states the user goes through when doing a certain activity. A breakdown of the diagram is shown below.

1. ***User login***: first the users will login in the login and the system will check for a successful login.
2. ***Fork:*** One the user logins it then goes off into four different paths depending on which type of user you are.
3. ***End user path***:
    a. Data collection view: The end users will be directed to the data collection view where they can begin the data entry.
    b. Data upload: this is where the data is uploaded by the end user the program will then start the data verification process.
    c. Data verification: the data will be verified and once it is the upload will be confirmed.
    d. Data base update: once the upload is confirmed the database will then be updated.
4. ***Environmental representative***:
    a. Compliance moderating view: the environment representative will first see the view for compliance moderating where they can retrieve the data collection.
    b. Ensure compliance: the environmental representative looks at the data to ensure the data complies with the requirements and the users will then approve the data collection.
    c. Compliance confirmation: The environmental rep will then confirm the data complies and the system will the confirm the compliance.
5. ***Research scientist***:
    a. Data analysis view: First the research scientist will be shown the data analysis view where they can view the data.
    b. Data retrieval: the research scientist can then retrieve the data they want so they can analyse it.
    c. Generates reports: the reports will then be generated with the analysed data and then the system will be notified that the analysis is complete.
6. ***Data analyst***:
    a. Data categorisation view: first the data analyst will see the data categorisation view where they will then be able to access the database.
    b. Categorise data: once they access it, they can then categorise the data. When they are done, they will the submit the categorised data.
    c. Database update: once it is categorised the analyst will be notified that it is updated, and the database will be updated.
7. ***End***: once everything is done the user will then log out and the system will end.

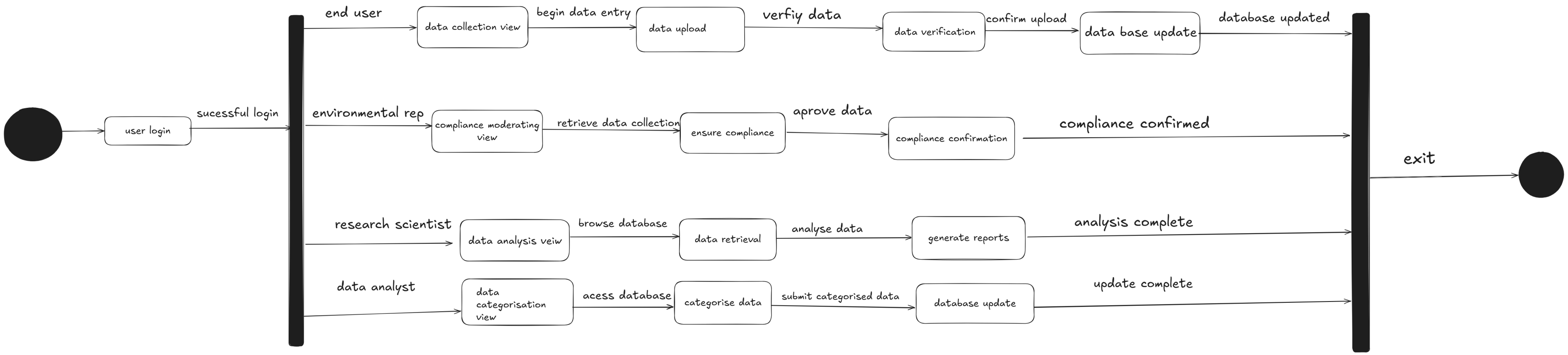

Figure 6 UML state diagram

4.0 Prototype Description

Provide a detailed description of your prototypes. The prototypes should be consistent with the scope description (WBS and deliverables) and requirement specification (covering all the relevant requirements). The prototypes should be testable and useful in generating insights to the whole project.

- *If you are developing a prototype of the UI/UX, you can use prototyping tools such as Marvelapp https://marvelapp.com/*
- *If you are implementing certain functionalities, you can use any programming language of your choice. Make sure your code is well-commented, executable, and testable.*
- *If you are developing the database, you can design your database using UML diagrams and convert your design into table structures. Also, you need to implement your database with SQL and populate your database with some synthesized data. You should also test the data retrieval of your database via SQL queries.*

4.0 Prototype Testing

4.1 Test Plan

This section should provide a test plan that contains the overall test strategy, including the test environment, test inputs, and expected test results. It can involve unit testing, integration testing, performance testing, customer acceptance testing, or any mixture of them. The choice of tests should be reasonable and feasible, and the test choices should be justified.

4.2 Test Results

This section should report the results of the testing. The test results should be documented and analysed, providing insights to the whole project. Limitations of the test results should be discussed (what they show and cannot show). You may consider including additional images or short video clips showing the test results.

5.0 Software Metrics

This section should present five corresponding metrics based on five project goals (of your choice), by applying the Goal/Question/Metrics paradigm. Justify the validity and reliability of your metrics related to the goals, and explain how to collect the required data.

6.0 Project Reflection

This section should present your reflection on how the project was managed, what went well and what could be improved. Also, discuss what you have learned during this project, and how they will benefit your future projects.