A black and white logo

AI-generated content may be incorrect.A black and red sign with white text

AI-generated content may be incorrect. Assessment Cover Sheet

This Assessment Cover Sheet is only to be attached to hard copy submission of assessments.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ASSESSMENT DETAILS** | | | | |
| Unit title | Computing Technology Innovation Project | Tutorial /Lab Group | Group 8 | Office use only |
| Unit code | COS30049 | Due date | 26/9/25 |  |
| Name of lecturer/tutor | Ts. Dr. Lee Sue Han | | |  |
| Assignment title | System Design Proposal | | | Faculty or school date stamp |
| **STUDENT(S) DETAILS** | | | | |

Student Name(s) Student ID Number(s)

|  |  |
| --- | --- |
| You Wee LIEW | 102786467 |
| Natalie ROBERT | 102787350 |
| Esther Ee Qian LING | 104381903 |
| Li Ying YEO | 102789314 |
| Elisa Bui Sian PUI | 104388991 |
| Gibson Chee Yii CHONG | 102779580 |

(1)

(2)

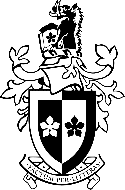
(3)

(4)

(5)

(6)

|  |  |  |
| --- | --- | --- |
| **DECLARATION AND STATEMENT OF AUTHORSHIP** | | |
| 1. I/we have not impersonated, or allowed myself/ourselves to be impersonated by any person for the purposes of this assessment. 2. This assessment is my/our original work and no part of it has been copied from any other source except where due acknowledgement is made. 3. No part of this assessment has been written for me/us by any other person except where such collaboration has been authorised by the lecturer/tutor concerned. 4. I/we have not previously submitted this work for this or any other course/unit. 5. I/we give permission for my/our assessment response to be reproduced, communicated, compared and archived for plagiarism detection, benchmarking or educational purposes.   I/we understand that:   1. Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to exclusion from the University. Plagiarised material can be drawn from, and presented in, written, graphic and visual form, including electronic data and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited. | | |
| **Student signature/s**  I/we declare that I/we have read and understood the declaration and statement of authorship. | | |
| (1) |  | (4) |
| (2) |  | (5) |
| (3) |  | (6) |

 Assessment Cover Sheet

This Assessment Cover Sheet is only to be attached to hard copy submission of assessments.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ASSESSMENT DETAILS** | | | | |
| Unit title | Computing Technology Innovation Project | Tutorial /Lab Group | Group 8 | Office use only |
| Unit code | COS30049 | Due date | 26/9/25 |  |
| Name of lecturer/tutor | Ts. Dr. Lee Sue Han | | |  |
| Assignment title | System Design Proposal | | | Faculty or school date stamp |
| **STUDENT(S) DETAILS** | | | | |

Student Name(s) Student ID Number(s)

|  |  |
| --- | --- |
| Daniel Meng Yeu TIONG | 102777801 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

(7)

(8)

(9)

(10)

(11)

(12)

|  |  |  |
| --- | --- | --- |
| **DECLARATION AND STATEMENT OF AUTHORSHIP** | | |
| 1. I/we have not impersonated, or allowed myself/ourselves to be impersonated by any person for the purposes of this assessment. 2. This assessment is my/our original work and no part of it has been copied from any other source except where due acknowledgement is made. 3. No part of this assessment has been written for me/us by any other person except where such collaboration has been authorised by the lecturer/tutor concerned. 4. I/we have not previously submitted this work for this or any other course/unit. 5. I/we give permission for my/our assessment response to be reproduced, communicated, compared and archived for plagiarism detection, benchmarking or educational purposes.   I/we understand that:   1. Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to exclusion from the University. Plagiarised material can be drawn from, and presented in, written, graphic and visual form, including electronic data and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited. | | |
| **Student signature/s**  I/we declare that I/we have read and understood the declaration and statement of authorship. | | |
| (7) |  | (10) |
| (8) |  | (11) |

**A black and red sign with white text

AI-generated content may be incorrect.**

**System Design Project Proposal**

**Unit Code:** COS30049

**Unit Name:** Computing Technology Innovation Project

**Tutor:** Ts.Dr. Lee Sue Han, Dr. Kelvin Yong, Dr. Mark Tee, Dr. Fu Swee Tee

**Team:** Group 8

**Members:**

You Wee LIEW

Natalie ROBERT

Esther Ee Qian LING

Li Ying YEO

Elisa Bui Sian PUI

Gibson Chee Yii CHONG

Daniel Meng Yeu TIONG

**Table of Content**

**Project Proposal:** SmartPlant Sarawak

**Part 1: Project Description**

**1.0 Background**

Sarawak is well known for its rich biodiversity and is home to thousands of plants species, including many that are rare and endangered plant. Despite this, identifying plants in the region remains a challenge because it usually requires expert knowledge or access to herbarium collections. The general public often relies on global plant identification applications, but these tools are not designed for Sarawak and do not always produce accurate results. In addition, sharing sensitive information such as the exact location of endangered plants can create risks if it is not properly protected. With the growth of artificial intelligence (AI), Internet of Things (IoT) technologies, cybersecurity and secure software practices, these is now an opportunity to create a platform that make plant identification easier, improves conservation efforts, and protects biodiversity information from misuse.

**1.1 Introduction**

Plant identification is an important activity for conservation, research, and education. Applications such as Pl@ntNet and iNaturalist have shown that artificial intelligence can help the public and researchers recognize plant species more easily. However, these systems are global in scope and are not focused on the plants of Sarawak. The current methods for identifying plants are fragmented, often hard for the public to access, and lack a way for people to collaborate on verifying findings. This leads to gaps and inaccuracies in biodiversity data. To tackle these issues, this project proposes designing SmartPlant Sarawak, a secure mobile application powered by AI and integrated with IoT monitoring. The system aims to help the public, researchers, and conservationists identify plant species accurately, map their distribution and protect endangered plants through community engagement and technology-driven solutions. This proposal explains how the system design will support biodiversity conservation while encouraging greater awareness and participation in protecting Sarawak’s unique ecological assets.

**1.2 Problem Statement**

Currently, Sarawak lacks a unified digital platform that can conveniently assist the public and researchers in identifying, mapping, and monitoring plant species. Plant identification still relies primarily on expert experience or traditional literature, which is slow, fragmented, and inaccessible to most people. Existing identification tools have limited accuracy for local plants, and there is also a lack of an expert review mechanism to verify questionable results. Furthermore, Sarawak's exceptionally high plant diversity makes continuously updating species records a significant challenge, resulting in widespread lags and incompleteness in existing data (Wong & Neo 2019, p. 463). Furthermore, there is currently a lack of a systematic approach that combines plant identification with real-time IoT-based monitoring of endangered species. This passive management approach not only limits researchers' ability to effectively protect vulnerable species but also hinders broader public participation in biodiversity conservation efforts. Therefore, building a comprehensive platform integrating plant identification, mapping, and real-time monitoring is urgently needed to support Sarawak's efforts in proactive biodiversity conservation.

**1.3 Solution**

The solution aims to develop a mobile-first platform that enables Sarawak’s plant biodiversity to be identified, mapped and protected by researchers and environmentalists. Accurate species identification, real-time biodiversity tracking and safe storage of data will be provided by platform in order to enhance conservation efforts by utilizing AI technology, community involvement and IoT monitoring.

**1.3.1 Existing Solutions**

There are several existing solutions for plant identification that can be used as a reference. Pl@ntNet is a well-known application where users contribute images through AI analysis (Pl@ntNet 2025), but its accuracy decreases when applied to species that are unique to Sarawak. iNaturalist is another platform that encourages community participation in biodiversity recording, but it does not provide adequate safeguards to protect sensitive data such as the locations of endangered species. Conventional methods, such as manual identification through field surveys and herbarium records, remain accurate but are time-consuming, resource-intensive, and not easily accessible to the public. While these solutions contribute to raising biodiversity awareness, none of them address the combined need for local accuracy, verification, monitoring, and secure data management.

**1.3.2 Proposed Solution**

The proposed solution is the development of a secure mobile application that supports biodiversity identification and monitoring in Sarawak. The application will use an AI model to identify plant based on photographs taken by users and will display the result with a confidence score to indicate reliability. If the prediction is uncertain, users will be able to flag the observation, which will then be reviewed by administrators or experts. An interactive map will show plant distribution, but sensitive locations of endangered plants will be masked or restricted to authorized administrators. IoT sensors will also be integrated to monitor environmental conditions around endangered plants and to send alerts when unusual changes occur. To ensure that the system is safe and trustworthy, strong security measures such as encryption, role-based access control, and multi-factor authentication will be implemented to protect user privacy and data. Last, Secure Software Development Life Cycle practices will be applied throughout the development process.

**1.4 Motivation**

The motivation behind this project comes from both ecological and social needs. Sarawak’s biodiversity is globally significant and protecting it is important for environmental sustainability and cultural heritage. At present, the absence of an effective and unified platform for plant identification and conservation creates challenges for both researchers and the public. The SmartPlant project is motivated by the need to bridge the gap with an innovative and accessible solution that support biodiversity protection in a secure and reliable way.

A key motivation is to promote plant biodiversity awareness and encourage public participation. By offering an AI-powered mobile application, the project makes plant identification simple, interactive and educational. Through features like visual maps and conservation status indicators are designed to inspire communities to engage with nature and contribute actively to protecting Sarawak’s unique flora. In this way, the project not only supports conservation but also promotes environmental awareness and learning among the public.

Another motivation is the opportunity integrates AI and IoT technologies to improve biodiversity monitoring and response. Artificial intelligence enables quick and accurate species identification, while IoT sensors provide real-time environmental data and generate alerts about potential threats such as habitat disturbance or poaching. By combining these technologies, the project ensures better data accuracy and enables researchers and conservationists to respond more effectively to protect endangered species.

**2.0 Objectives**

The main purpose of this project is to design and develop a secure mobile application that uses artificial intelligence (AI) and Internet of Things (IoT) technologies to support plant identification and biodiversity conservation in Sarawak. The application will not only provide plant identification for the public but also allow experts to verify the result, display interactive biodiversity maps and monitor endangered plants using IoT sensors. Security will be a core part of the system to protect sensitive data, especially the locations of rare species.

**2.1 Project Objectives**

The objective of this project is to design and deploy an AI-driven mobile application that engages the public in plant identification, mapping and biodiversity monitoring. This project helps to enhance public engagement, raise the accuracy of biodiversity data and improve conservation management in Sarawak.

Our project aims to:

* + Develop an AI-driven plant identification system that can identify plant images and provide user with the plant status along with a confidence score.
  + Develop an interactive mapping feature to display plant sighting and track endangered plant species while protecting sensitive data.
  + Integrate IoT-based monitoring systems by using sensors to provide real-time protection and continuous monitoring endangered plants.
  + Ensure strong cybersecurity and data privacy by securing user information and sensitive biodiversity data through encryption, role-based access controls and multi-factor authentication.
  + Enable collaborative validation mechanisms that allow admins and users to review, verify and improve plant species records.

**2.2 Scope**

The scope of this project is focused on building a working prototype of the biodiversity application. The application will be developed as mobile app that integrates artificial intelligence, mapping, and IoT concepts to support plant identification and biodiversity conservation. Given the extensive biodiversity across Sarawak, the project will lower the scope by focusing on a specific region rather than attempting to cover the entire state. This approach ensures that the prototype remains realistic and manageable within the academic timeline, while still demonstrating how AI-assisted plant identification, geospatial visualization, and simulated IoT monitoring can be integrated into a functional system. Supporting documentation such as system design diagrams, testing strategies, risk management, and security considerations will also be produced to provide a complete and academically rigorous design.

Out of scope for this project are large-scale biodiversity data collection across Sarawak, deployment of physical IoT hardware in the field, and the public release of the mobile application. Instead, the system will use a smaller, representative dataset from the selected region to showcase its core features. Regulatory compliance and conservation policies will also be discussed at a conceptual level only, rather than implemented. By narrowing the geographical scope and excluding large-scale deployment tasks, the project remains focused on delivering a well-defined system design and prototype that highlight the potential of AI, IoT, and community-driven features to support plant conservation in a practical and targeted way

**2.3 Limitations**

This project is limited by several factors that must be considered during development. The timeline is restricted to one academic semester, which means that only core features can be implemented within the available time. Some advanced features, such as full IoT integration and large-scale field trials, may not be fully realized during the initial phase and could require future enhancements. Another key limitation is the availability of IoT hardware. Deploying environmental sensors across conservation sites demands significant resources, so monitoring may initially be limited to selected areas. This reduces the platform's ability to provide complete real-time coverage of endangered species habitats. In addition, the AI model's performance depends heavily on the amount and quality of training data. Rare species may be harder to recognise until more verified images are collected. Similarly, while cybersecurity measures are included, safeguarding sensitive biodiversity data may need continuous refinement beyond the project's first release. These limitations set the boundaries of what the project can achieve in its first version.

**2.4 Strength**

Our team possesses several strengths that place us in a good position to deliver the project. These strengths are outlined as follows:

First, one of the key strengths of our team is we are equipped with a wide range of expertise, covering nearly all the professional areas required for the project. Our team is made up of students from almost all Computer Science majors, including Artificial Intelligence, Cybersecurity, Data Science, and Software Development. Although our team lacks a member from the Internet of Things (IoT) major, we have all taken IoT-related courses in the past semesters and gained basic understanding of it, which enables us to pick up IoT-related tasks more easily during the project. The diversity of our majors allows us to approach the project from multiple perspectives and ensures that the wide scope of technical requirements can be fulfilled effectively.

Another strength lies in our ability to train a custom AI model based on established architectures for the plant identification. This means we have greater flexibility and specificity precisely for this project. By fine-tuning and adapting the model to Sarawak plant datasets, the team can achieve greater accuracy and ensure the solution is more suitable for Sarawak’s biodiversity.

Finally, our team has a high level of motivation to complete this project successfully. The final aim of the project is to implement a secure, AI-powered mobile application that enables plant species identification, collaborative verification, and mapping, with IoT-based monitoring to protect endangered species. Since our team members are Malaysians and currently studying in Sarawak, this project is directly relevant to our environment and community. The biodiversity challenges close to our life gives us greater motivation to work actively and to ensure that the final product is both practical and impactful. This not only deepens our passion for our field of study but also gives us a sense of achievement, knowing that our work can contribute to biodiversity protection in Sarawak.

**3.0 Stakeholders and Persona**

The success of this project depends on understanding the different stakeholders who will use the system and the personas that represent their needs. By identifying these groups, the project can be designed to ensure the features are relevant, secure and practical for both users and experts.

**3.1 Stakeholders**

Identifying and prioritizing key for a project, followed by establishing and maintaining productive work relationships, is important for achieving project success. To identify all parties who may be involved in this project, either directly or indirectly, the following table present a list of Stakeholders along with their respective contributions in the project.

|  |  |  |
| --- | --- | --- |
| Stakeholders | Roles | Contributions |
| Scrum Master | Facilitator and guide | * + Ensure the team follows Agile principle and Scrum practices.   + Provide guidance to improve productivity and teamwork.   + Use sprint burndown charts to monitor progress and make sure features are delivered on time. |
| Product Owner | Decision Maker and Backlog Manager | * + Provide the vision of the project and ensure it aligns with the objectives.   + Manage and prioritize the product backlog.   + Communicate project requirements with the client and provide feedback to the team. |
| Scrum Member | Designers, developers and testers | * + Design, develop and test the features of the system architecture.   + Ensure that core functions are implemented and they meet requirements.   + Conduct code reviews and make sure that best practices and SSDLC are applied. |
| User | Data Contributor | * Use the mobile app to take pictures of plants and upload them. * Flag uncertain identifications and wait for admin evaluation. * See interactive maps with plant observation locations, images and species information. |
| Admin | Conservation Manager | * Manage user roles and access permissions. * Protect sensitive plant locations by masking endangered species data. * Monitor real-time IoT sensor data for environmental conditions and threats. |
| Plant Expert | Validator and Data Collector | * Validate plant species when the average of AI accuracy is lower than 70%. * Refer to the plant species database, especially for endangered plants. |

**3.2 Persona**

To ensure that the system meets the needs of different users, the project considers several user personas. These personas represent typical users and their expectations from the system.

|  |  |
| --- | --- |
| **Persona** | **Expectations** |
| Student | Enjoys exploring nature and uses the app to take photos of plants and instantly learn their name |
| Officer | Uses the application to receive IoT alerts about endangered plants and monitors changes in their environment |
| Researcher | Relying on verified data collected from the system to support biodiversity studies and conservation planning |

Table 1: Understand the expectations of different users

From our understanding from the table 1 above, these users have different ways of thinking and interacting with the system. Students feel curious and excited when they are able to discover the names of plants instantly. They see the application as a learning tool and often say and do things such as sharing their discoveries with friends or classmates.

Conservation officers think and feel responsible for protecting endangered plants. They see the IoT alerts as helpful tools that allow them to respond quickly to changes in the environment, and they often say and do things related to fieldworks and conservation activities.

Researchers think and feel motivated by the availability of reliable data. They see the system as a trusted source of verified information, and they say and do things that involve using the data for analysis, research paper, and planning conservation strategies.

Conclusion, all of these users hear about biodiversity either from their peers, communities or institutions, which shapes their expectations and encourage them to use the system.

**Part 2: Project Planning**

**4.0 Project Analysis**

Project planning provides a clear framework for how the system will be developed and managed within the given time and resources. It defines the analysis of the solution, the method that will be applied, and the factors that influence the project. Planning is also important to identify risks, constrains, and opportunities early, so the team can take action to reduce challenges and improve success. The solution for this project is to design and build a mobile application called SmartPlant that integrates artificial intelligence, IoT monitoring, and cybersecurity into one system. The solution analysis helps to understand the tools, practices, and strategies needed to make this application both effective and secure.

**4.1 SSDLC [Esther]**

In this project, we will use the Secure Software Development Life Cycle (SSDLC) methodology to make sure that security is implemented into every phase of development. By following to the SSDLC, we can find a balance between security, usability and functionality to satisfy the needs of all stakeholders involved.

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Activities | Security Considerations | Output |
| Planning and Security Requirement | Define project scope, identify stakeholders and research user needs | List out security requirement | Project scope and security requirements identified and documented |
| Secure Design and Prototyping | Create system architecture and app prototype | Apply secure design principles | System design document and prototypes created |
| Secure Development | Develop core features | Make use of secure coding practices | Secure source code and functional modules |
| Security and Vulnerability Testing | Conduct unit tests, vulnerability scanning and penetration tests. | Test authentication | Test results |
| Secure Deployment | Deploy mobile app, backend services and apply access policies in a secure cloud environment. | Enforce HTTPS and secure configuration | Deployed system and security configuration |
| Maintenance and Monitoring | Monitor IoT data, app usage and system logs. Gather feedback. | Maintain security with patches and monitoring | Maintenance logs |

At first, in the Planning and Security Requirement phase, we define the project scope and identify the stakeholders which are important in this project. We also focus on security requirements including protecting endangered plant records, role-based access control for different user categories and encrypting sensitive data. The next phase is Secure Design and Prototyping phase, where the system architecture is developed with security design principles. Wireframes and prototypes are developed to provide a draft overview of the system. Next, for the Secure Development phase, we implement the application core features like plant identification, mapping functions, IoT-enabled monitoring and security controls. We make sure that both usability and security are taken into consideration while creating features like dashboards, reporting tools and image uploads. Security testing tools and code reviews are also integrated into the workflow to identify vulnerabilities early. After implementation, the project moves into the Security and Vulnerability Testing Phase, which involves functional testing, usability checks and security testing. Security mechanisms such as multi-factor authentication, data encryption and role-based access control are validated. To ensure that risks have been reduced prior to deployment, any vulnerabilities found are noted, addressed and retested. In the Secure Deployment phase, the system is launched in a secure cloud environment. The deployment procedure makes sure that all communications occur over secure protocols, like HTTPS. Finally, the last phase is Maintenance and Monitoring phase. Regular updates and patches are applied to identify emerging activity and monitoring is conducted to detect suspicious activities. User feedback is incorporated into improvements. By following to these SSDLC phases, the project ensures that security is integrated throughout the lifecycle, lowering risks and producing a system that is reliable, strong, and in line with objectives for biodiversity conservation.

**4.2 Solution Analysis**

**4.2.1 Software Track**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Problem Domain:  There is a shortage of biodiversity applications that combine modern software technologies into a single system. Existing tools rarely include AI services, cloud databases, IoT simulations, and administrative dashboards in a cross-platform mobile app. The primary software problem is to create a safe, scalable, and user-friendly system that connects mobile users, backend services, and cloud storage successfully. | | | | |
| Potential solution | Project Force | SWOT | KoST | Final Decision |
| Cross-platform mobile app with a cloud backend | Create a single cross-platform mobile app using React Native, a Node.js backend, and Firebase for authentication, database, and hosting. | **Strengths:**  A single codebase for Android and iOS allows for quicker development, and Firebase provides a ready-made secure backend.  **Weakness:**  Performance is slightly lower than native programs.  **Opportunities:**  Scalable, with AI and IoT connection, and easy to add features.  **Threats:**  Vendor lock-in with Firebase causes React Native updates to break. | **Knowledge Gap:**  A thorough understanding of the React Native ecosystem and Firebase rules are required.  **Skill Gap:**  Cross-platform UI/UX design with a Node.js backend and cloud integration.  **Technology Gap:**  Relies on third-party libraries (React Native, Firebase SDK), which can change quickly. | Selected |
| Native Mobile Application | Create two independent native apps for Android and iOS that have backend integration and database functionality. | **Strengths:**  High performance, improved access to device APIs, and optimized UI/UX for each platform.  **Weakness:**  Using two independent codebases leads to increased maintenance work and cost.  **Opportunities:**  Can integrate deeper with native features like camera, GPS, sensors for future biodiversity expansion.  **Threats:**  Slow development cycle cause by inconsistency in features if Android and iOS teams progress independently. | **Knowledge Gap:**  Lack skills in both Swift and Kotlin.  **Skill Gap:**  Experienced in maintaining two independent codebases as well as backend integration.  **Technology Gap:**  Using limited cross-platform frameworks and no standard code increases the danger of duplication. | Rejected |
| Progressive Web Application with cloud backend | Create a web application using ReactJS, hosted on a cloud provider, with backend API and database integration. | **Strengths:**  Single app accessible via web, no app store approval, and simple upgrades.  **Weakness:**  Limited offline functionality, with less access to native device features such as the camera and GPS.  **Opportunities:**  Simple scaling across devices like desktop and mobile with little time required for installation.  **Threats:**  Different iOS and Android versions have different progressive web application compatibility, which could lead to uneven user experiences and restricted feature availability on some devices. | **Knowledge Gap:**  Understands progressive web application standards and browser compatibility.  **Skill Gap:**  Web performance enhancement, flexible UI design, and API integration.  **Technology Gap:**  Less support for complex native smartphone functions compared to React Native or native apps | Rejected |

Justification:

A cross-platform mobile app with a cloud backend that incorporates React Native, Node.js, and Firebase is accepted since it is the most practical and scalable solution for the project. Unlike native mobile applications, which require two different native codebases, React Native allows for cross-platform development using a single codebase, minimizing complexity and time. When compared to a Progressive Web App with a cloud backend, the PWA solution does not have sufficient access to device features like the camera and GPS, which are required for AI-driven plant recognition.

A cross-platform mobile app with a cloud backend combines React Native's cross-platform capability with Firebase's secure cloud services and Node.js backend flexibility to deliver an efficient, maintainable, and future-ready software solution that is matched with the project's scope and academic timeframe.

**4.2.1 AI Track and DS Track**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Problem Domain:  The plant classification system may misclassify rare or unseen plant species due to limited dataset size, class imbalance, and environmental variations, reducing model accuracy and user trust | | | | |
| Potential solution | Project Force | SWOT | KoST | Final Decision |
| Transfer Learning + Data Augmentation + Periodic Model Retraining | Time: Medium  Cost: Not Applicable Scope: On-scope | **Strength:** Improves accuracy and generalization for rare species; adapts to new data  **Weakness:** Requires careful validation of retraining  **Opportunities:** Can handle class imbalance and evolving datasets  **Threats:** Uses storage and computation for retraining | **Knowledge Gap:** Limited experience in retraining pipelines and advanced augmentation  **Skill Gap:** No team member has implemented retraining in ML projects  **Technology Gap:** None major; frameworks like TensorFlow and PyTorch support these methods | Selected |
| Classical ML on Extracted Features (SVM, Random Forest) | Time: Low  Cost: Not Applicable  Scope: On-scope but limited | **Strength:** Easier to implement on small datasets  **Weakness:** Low accuracy for rare or diverse plant images  **Opportunities:** Can experiment quickly with feature engineering  **Threats:** May fail to generalize; misses complex visual features | **Knowledge Gap:** Few team members have experience with feature extraction for images  **Skill Gap:** Limited expertise in applying classical ML to image data  **Technology Gap:** None major | Rejected |
| Federated Learning (User-Device Training) | Time: High  Cost: Not applicable Scope: Out-of-scope | **Strength:** Adapts to diverse user data; privacy-preserving  **Weakness:** Very complex to implement and maintain  **Opportunities:** Real-time model improvement across users  **Threats:** Requires advanced infrastructure and coordination | **Knowledge Gap:** Federated learning, secure gradient aggregation  **Skill Gap:** No experience with distributed training  **Technology Gap:** Requires specialized frameworks and server orchestration | Rejected |

Justification: The selected solution directly addresses the problem domain by improving generalization, balancing classes and keeping the model sustainable. Classical ML is easier to implement but manual feature extraction cannot handle complex images feature, so it is less appropriate. Federated Learning could be effective but is too complex in restricted timeline and resources of this school project.

**4.2.2 Cybersecurity Track**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Problem Domain: SmartPlant Sarawak gathers user data and sensitive biodiversity data, locations of rare and endangered plants, as well as user information. Without strong cybersecurity, this data may be exploited for illegal activities such as poaching, logging, or breaches of privacy. | | | | |
| Potential Solution | Project Force | SWOT | KoST | Final Decision |
| Role based access control (RBAC) + multi-factor authentication (MFA) + data encryption for plant data | Aim: Protect user and administrator access, encrypt sensitive species data, and secure logins  Time: Moderate Cost: Not applicable  Scope: On scope | **Strengths:** Strong protection of both user and biodiversity data **Weaknesses:** Extra setup and login steps for users **Opportunities:** Builds trust and ensures data compliance **Threats:** MFA or encryption can fail if poorly managed | **Knowledge Gap:** Limited understanding of encryption (AES, RSA), RBAC policies, and MFA procedures  **Skill Gap:** No experience in implementing encryption or configuring MFA in mobile application **Technology Gap:** None major – Firebase supports encryption and MFA | Selected |
| IoT device-level security like sensor encryption and anomaly alerts | Aim: Protects IoT monitoring devices from tampering or hacking  Time: High Cost: Not applicable  Scope: On scope but limited | **Strengths:** Protects live monitoring data **Weaknesses:** Expensive and difficult to implement **Opportunities:** Enhance long-term IoT security. **Threats:** IoT hardware remains vulnerable. | **Knowledge Gap:** Limited knowledge of IoT-specific cybersecurity threats  **Skill Gap:** No experience in securing IoT hardware or anomaly detection  **Technology Gap:** Requires additional IoT security tools and device-level encryption beyond current scope | Rejected |
| Advanced Intrusion Detection System (IDS) Integration | Aim: Monitors system traffic and alerts for suspicious activity  Time: High Cost: Not applicable  Scope: Out of scope | **Strengths:** Helps detect cyberattacks early. **Weaknesses:** Need continuous monitoring and analysis of packet traces **Opportunities:** Stronger cybersecurity monitoring. **Threats:** Attackers may use advanced evasion techniques to bypass IDS detection | **Knowledge Gap:** Limited of IDS concepts and log monitoring  **Skill Gap:** Only one team member has experience in deploying or managing IDS **Technology Gap:** Requires third-party tools like Snort | Rejected |

Justification: We choose RBAC with MFA and data encryption as our selected solution because it provides a balanced and smart way to secure both user and plant data. It can protect user personal information and preventing unauthorised access to sensitive biodiversity data. Users may have to take more procedures like when user wanted to access a login protected by a CAPTCHA, user must complete the human verification test presented by the CAPTCHA, but the advantages of improved security, compliance to data protection standards and increased user trust exceed this inconvenience. Hence, it ensures that only authorized users can access important information and keeps endangered plant locations encrypted to lower the risk of poaching or misuse.

The IoT device-level security solution was rejected because although protecting real-time monitoring data, it is costly, technically challenging and beyond the team's existing capabilities. Additionally, the Advanced IDS integration was rejected since it is inappropriate for the current project scope and necessitates ongoing monitoring, specialised tools and skills that the team does not have. As a result, the chosen solution offers the best possible balance between project resources, efficacy, and practicality.

**4.2.3 IoT Track**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Problem Domain: The current biodiversity monitoring process lacks real-time IoT-enabled detection and alerting systems. Without IoT integration, it is difficult to track endangered species movements, transmit data to cloud storage, and provide immediate alerts to users or administrators. This results in delayed responses and limits the effectiveness of conservation measures. | | | | |
| Potential solution | Project Force | SWOT | KoST | Final Decision |
| Node MCU microcontroller with HC-SR501 motion sensor, buzzer alert, and Firebase database integration | Aim: Detect motion, trigger alerts, and send data to the cloud for real-time monitoring | **Strengths**: Low-cost, easy to implement, supports real-time data logging and alerts; scalable with other sensors.  **Weaknesses:** Dependent on stable internet connection, limited range of PIR motion sensor.  Opportunities: Can expand to more IoT devices such as cameras, GPS, or environmental sensors.  **Threats:** Device tampering or sensor malfunction may lead to false alerts. | **Knowledge Gap:** Limited understanding of optimizing MQTT/HTTP protocols for reliable data transfer.  **Skill Gap:** Team lack experience in configuring NodeMCU Firebase integration  **Technology Gap:** No prior use of real-time IoT clouds synchronization in projects. | Selected |
| Rasberry Pi with camera-based monitoring system | Aim: Use a higher-powered device with image capture and object detection | **Strengths:** Can detect species more accurately with vision-based input.  **Weaknesses:** Higher cost and power consumption, more complex software integration.  **Opportunities:** Stronger AI integration for image classification.  **Threats:** Not energy efficient for remote biodiversity monitoring. | **Knowledge Gap:** Lack of expertise in integrating Raspberry Pi camera with cloud backend.  **Skill Gap:** Team inexperienced in building real-time image capture pipelines.  **Technology Gap:** Requires higher computational hardware not available in scope. | Rejected |
| LoRaWAN-based IoT network for large-area coverage | Aim: Use long-range, low-power wireless communication to cover wide monitoring areas | **Strengths:** Very wide coverage, low power consumption, suitable for large biodiversity zones.  **Weaknesses:** Requires LoRa gateways and setup, not easily available in urban project context.  **Opportunities:** Scales well for future biodiversity IoT projects.  **Threats:** Complex to configure and outside the skillset of the current team. | **Knowledge Gap:** No team knowledge of LoRaWAN protocols and development.  **Skill Gap:** Lack of experience in configuring and managing LoRaWAN gateways.  **Technology Gap:** Additional hardware infrastructure outside current resources. | Rejected |

Justification:

The selected solution is the NodeMCU microcontroller integrated with the HC-SR501 motion sensor, buzzer, and Firebase database, as it provides a practical, affordable, and scalable IoT setup. This configuration directly supports motion detection, triggers immediate alerts through the buzzer, and ensures data is stored in Firebase for real-time monitoring by both users and administrators. It fits well within the project’s scope, timeline, and available resources.

The Raspberry Pi with a camera-based system was rejected because while it offers higher accuracy, it is too resource-heavy, costly, and complex to implement within the project’s constraints. Similarly, the LoRaWAN IoT network was dismissed since it requires additional infrastructure and advanced setup beyond the scope of the project.

By choosing NodeMCU with motion sensing, the team ensures a low-cost, real-time, and cloud-connected biodiversity monitoring system that aligns with the project objectives while leaving room for future expansion.

**4.3 Risk Management [Natalie]**

A structured approach to risk management is essential for anticipating and addressing potential challenges, optimizing the use of human resources, and enhancing workflow efficiency. Organized risk assessments are particularly critical, as unresolved risks can significantly disrupt project development. Our team employs systematic risk classification using Confluence templates, which enables timely identification, reduction, and resolution of risks to mitigate these problems. This proactive strategy not only ensures smoother project execution and consistent progress but also fosters adaptability and resilience in problem-solving throughout the development lifecycle.

Identified risks must be addressed systematically to minimize their impact on the project’s success. Risk treatment involves developing and applying mitigation strategies that either reduce the likelihood of risks occurring or lessen their potential severity. For this project, the team adopts proactive measures such as structured testing, modular design, version control, and early deadline planning to ensure risks are managed effectively. By anticipating potential obstacles and preparing fallback options, the project can maintain steady progress, safeguard core deliverables, and improve overall resilience. The following tables outline the specific risks, their likelihood, severity, and corresponding mitigation strategies.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Likelihood** | **ACCEPTABLE**  *Little to no effect on event* | **TOLERABLE**  *Effects are felt, but not critical to outcome* | **UNDESIRABLE***Serious impact to the course of action and outcome* | **INTOLERABLE***Could result in disaster* |
| **IMPROBABLE** *Risk is unlikely to occur* | Minor formatting errors in documentation | Slight delays in submitting intermediate drafts | Minor inaccuracies in diagrams or charts | Loss of project files due to system crash without backup |
| **POSSIBLE** *Risk will likely occur* | Small coding bugs identified during testing | Difficulty integrating AI model with mobile app prototype | Incomplete testing leads to reduced system reliability | Critical component (e.g., AI image recognition) fails to function |
| **PROBABLE** *Risk will occur* | Minor version control conflicts resolved quickly | Limited dataset reduces AI training accuracy | Poor system performance during demonstrations | Project cannot meet core requirements within timeline |

|  |  |
| --- | --- |
| Risk | Mitigation Strategy |
| Minor formatting errors in documentation | Assign a member to perform final proofreading and use standardized templates |
| Slight delays in submitting intermediate drafts | Set internal deadlines earlier than official submission dates to allow buffer time. |
| Minor inaccuracies in diagrams or charts | Cross-check diagrams against requirements definition and validate with lecturer feedback. |
| Loss of project files due to system crash without backup | Use version control (e.g., GitHub) and regular cloud backups to secure all files. |
| Small coding bugs identified during testing | Conduct unit testing and peer code reviews frequently to catch and resolve bugs early |
| Difficulty integrating AI model with mobile app prototype | Use modular design, define clear APIs, and test components incrementally before integration |
| Incomplete testing leads to reduced system reliability | Develop a structured testing plan including unit, integration, and system tests |
| Critical component (e.g., AI image recognition) fails to function | Prepare fallback strategies, such as simplified prototype demo. |
| Minor version control conflicts resolved quickly | Establish clear Git workflow (branching, pull requests, merge approvals) |
| Limited dataset reduces AI training accuracy | Augment dataset with public sources and clearly state limitations |
| Poor system performance during demonstrations | Optimize code, reduce image size/processing load, and pre-load sample inputs for smoother demos |
| Project cannot meet core requirements within timeline | Prioritize must-have features, track progress weekly, and reallocate tasks if delays occur. |

Ongoing risk monitoring and review are essential to keeping risk management strategies both effective and adaptable. This continuous process ensures that emerging risks are identified promptly and that existing strategies are refined to remain relevant throughout the project lifecycle. By conducting regular assessments, the team can obtain timely insights that enhance decision-making. Routine reviews not only support compliance with project requirements but also allow strategies to be fine-tuned in order to address the specific context of the project. Ultimately, this continuous oversight can ensure the project remains prepared to respond effectively to evolving risks

**4.4 Constrains**

The project is affected by several constrains that set the boundaries of what can realistically be achieved. These constrain include time, budget, technical limitations, and human resources. By identifying this constraint clearly, the project team can plan realistically and prioritize the most important features.

|  |  |  |
| --- | --- | --- |
| **Constraint** | **Description** | **Impact on Project** |
| Time | The project must be completed within one academic semester | Limited the development of advanced features and reduces time for extensive testing and refinement |
| Budget | Resources for IoT devices, cloud services and other tools are limited | Restricts the number of devices and services that can be deployed during the prototype phase |
| Available Data | Local plant datasets are small compared to global datasets | May reduce the accuracy of the Ai model in identifying rare or unique species |
| Human Resources | The project team is small and has limited technique expertise across all areas | Slow development progress and requires careful task distribution among team members |

The table above shows the main constrain that influence the project and how they affect development. By recognizing these constraints, the project focuses on achievable goals and core features. While some advanced functions may not be possible in this version, the system can still deliver strong results and lay the foundation for future improvements.

**Part 3: Deliverables and Schedule**

**5.0 Deliverables and Schedule**

The project deliverables and schedule provide a clear plan of what the team will produce, when tasks will be completed, how progress will be managed. The project is developed using Agile approach, which allows the team to deliver features in iterations, test them early, and improve them based on feedback. This ensures that the core functions are delivered on time while still allowing flexibility for refinement.

**5.1 Activities, Tasks and Key Deliverables**

The project included several activities and tasks that are required to deliver the final product. These tasks are broken into smaller deliverables to make progress measurable and easier to track.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Tasks** | **Key Deliverables** |
| Project Planning | Define scope, objectives and risks | Project proposal and planning documents |
| System Design | Create architecture, database schema, and security model | System design document with diagrams |
| UI/UX Design | Develop wireframes, layout design and navigation flow | Approved UI design and prototype interface |
| AI Model Development | Collect dataset, train model and test accuracy | Functional AI plant identification service |
| Mobile Application Development | Build user interface, integrate AI service, implement mapping | Mobile app prototype with plant ID and map features |
| Database Development | Design schema, configure secure storage, enable queries | Database for plants, user, and IoT data |
| IoT Integration | Configure sensors, connect to backend, set up alert | Working IoT system with real-time monitoring |
| Security Implementation | Apply SSDLC, add encryption, MFA | Secure authentication and protected data storage |
| Testing | Conduct functional, usability, and security tests | Test report with result and recommendations |
| Documentation | Write user guide, technical docs and final report | Completed documentation and project report |
| Final Delivery | Prepare presentation and submission | Completed SmartPlant system |

**5.2 Milestones**

Milestones represent important achievements that show clear progress and help team measure if it on track in the project. Each milestone is tied to a deliverable or a major task, and together they mark the key stages of development from planning to delivery.

|  |  |  |
| --- | --- | --- |
| **Milestone** | | **Expected Completion** |
| 1 | Project Proposal Completed | Week 2 |
| 2 | System Architecture Finalized | Week 3 |
| 3 | UI/UX Designed | Week 3 |
| 4 | Implemented Design to Mobile Application | Week 5 |
| 5 | Database Implemented | Week 5 |
| 6 | AI Model Trained & Tested | Week 7 |
| 7 | Admin Verification & Map Features Completed | Week 9 |
| 8 | Mobile App Prototype Ready | Week 9 |
| 9 | IoT Integration Completed | Week 10 |
| 10 | Security Features & Hosting Implemented | Week 10 |
| 11 | System Testing Conducted | Week 11 |
| 12 | Final Report & Presentation | Week 12 |

**5.3 Schedule Overview**

The schedule provides a timeline for the project from the start of the final delivery. It shows when the main activities will be carried out and ensures that the project stays on track. Each activity is arranged week by week across the semester so the team can measure progress and adjust when needed.

|  |  |
| --- | --- |
| Week | Main Activities |
| 1 – 2 | Project planning and proposal preparation |
| 3 | System architecture finalized and UI/UX design completed |
| 4 – 5 | Mobile application development begins, database implemented |
| 6 – 7 | AI model training, testing, and integration |
| 8 - 9 | Admin verification features and biodiversity map completed; mobile prototype ready |
| 10 | IoT integration completed; security features and hosting implemented |
| 11 | Functional, usability, and security testing conducted |
| 12 | Final report and presentation prepared; SmartPlant system delivered |

**5.4 Delivery Phases**

The project is delivered in phases that follow an Agile approach. Each phase focuses on a specific part of the system and ends with a working deliverable that can be tested and improved. This iterative method allows the team to deliver features step by step while collecting feedback to refine the system.

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Main Tasks | Expected Outcome | Weeks |
| Phase 1 – Planning and Design | Define scope, objectives, risks; finalize system architecture and UI/UX design | Requirements analysis, dataset prep, AI prototype and UI design | 1 - 3 |
| Phase 2 – Core Development | Implement mobile application features, set up database, begin AI model training | Functional database, early version of mobile app, initial AI model | 4 – 6 |
| Phase 3 – Integration | Complete AI model testing, add admin verification, develop biodiversity map | Working plant identification system with verification and mapping | 7 - 9 |
| Phase 4 – IoT and Security | Integrate IoT sensors, add hosting, apply encryption, MFA, and RBAC | Secure and integrated system with real-time IoT monitoring | 10 |
| Phase 5 – Testing and Delivery | Conduct system testing, prepare documentation, finalize report and presentation | Completed SmartPlant system with documentation and demo | 11 - 12 |

**5.5 Initial Release Schedule**

**5.5.1 Product Backlogs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product Backlog ID | Product Backlog Item | (Product Backlog ID) | Business Value (1 least – 10 most) | Release Schedule (Sprint #1 | 2) |
| 1 | Define system scope and objectives | – | 9 | Sprint 1 |
| 2 | Design UI/UX wireframes | 1 | 8 | Sprint 1 |
| 3 | Develop mobile app basic interface (React Native) | 2 | 9 | Sprint 1 |
| 4 | Implement navigation and task flow | 3 | 8 | Sprint 1 |
| 5 | Set up Firebase Firestore database | 1 | 10 | Sprint 1 |
| 6 | Configure Firebase authentication | 5 | 10 | Sprint 1 |
| 7 | Implement AI dataset collection (local + global images) | 1 | 9 | Sprint 1 |
| 8 | Train initial AI model for plant identification | 7 | 10 | Sprint 1 |
| 9 | Optimize AI model with image augmentation | 8 | 8 | Sprint 1 |
| 10 | Connect AI model to backend (Node.js API) | 8 | 9 | Sprint 2 |
| 11 | Implement image upload function in mobile app | 3, 10 | 10 | Sprint 2 |
| 12 | Display plant identification results with confidence level | 11 | 10 | Sprint 2 |
| 13 | Add admin dashboard for plant verification | 5 | 9 | Sprint 2 |
| 14 | Implement biodiversity map with plant markers | 5 | 8 | Sprint 2 |
| 15 | Add plant search and filter functionality | 14 | 8 | Sprint 2 |
| 16 | Configure IoT sensors for environment data (temperature, humidity, GPS) | 5 | 9 | Sprint 1 |
| 17 | Calibrate IoT sensors for accuracy | 16 | 8 | Sprint 1 |
| 18 | Integrate IoT data into backend | 16 | 9 | Sprint 2 |
| 19 | Enable real-time alerts for endangered plants | 18 | 10 | Sprint 2 |
| 20 | Apply role-based access control (RBAC) for admin/users | 6 | 10 | Sprint 2 |
| 21 | Add multi-factor authentication (MFA) | 6 | 9 | Sprint 2 |
| 22 | Apply SSL/TLS for secure communication | 6 | 10 | Sprint 2 |
| 23 | Implement secure Firestore access rules | 5 | 10 | Sprint 1 |
| 24 | Conduct functional testing of all modules | ALL | 10 | Sprint 2 |
| 25 | Conduct usability testing with sample users | 24 | 9 | Sprint 2 |
| 26 | Conduct penetration testing for security validation | 22, 23 | 9 | Sprint 2 |
| 27 | Prepare user guide (installation + app usage) | 24 | 8 | Sprint 2 |
| 28 | Prepare technical documentation (API, DB schema) | 24 | 8 | Sprint 2 |
| 29 | Conduct system integration testing | 10, 19 | 10 | Sprint 2 |
| 30 | Deliver final report and presentation | 27, 28 | 9 | Sprint 2 |

**5.5.2 Product Timelines**

**A screenshot of a computer

AI-generated content may be incorrect.Sprint 1**

**Sprint 2**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Part 4: System Architecture**

**6.0 System Architecture**

The system architecture explains how the SmartPlant application is designed and how its components work together. It shows the overall framework of the system, the main features, and the technical design that connects hardware, software, and security. A clear architecture ensures that the system is reliable, scalable, and aligned with the project objectives.

**6.1 Overall Framework**

The SmartPlant system adopts a layered framework to ensure modularity and scalability. It is structured into four main layers, the user interface (UI), application services, IoT monitoring and data management, with a security layer applied across all levels.

**6.1.1 UI Design**

The user interface (UI) design was developed to ensure that the software interface provides a clear, consistent, and intuitive interaction flow for both general users and administrators. The design emphasizes accessibility, ease of navigation, and support for the system’s core functions.

Screens screenshot of a login screen

AI-generated content may be incorrect.

**Authentication Screens**

* **Login Screen**: Provides input fields for email and password, a “Remember Me” option for persistent login, options for social login (Google/Facebook), password recovery, and navigation to the Register screen.
* **Register Screen**: Allows new users to enter full name, email, password, confirm password, and accept terms before creating an account.

A screenshot of a computer

AI-generated content may be incorrect.

**User Screens**

* **User Dashboard**: Displays greeting, user information, search bar, recent identifications, posts, and quick access to other modules via the bottom navigation bar.
* **Map Screen**: Shows plant sightings on a map with markers, a search bar, and filter controls. Selecting a marker opens plant details.
* **Identify Screen**: Provides live camera capture for plant identification, with options for single or multiple recognition. User can also upload images from gallery.
* **Identification Result Screen**: Displays the captured image along with AI results (plant name, accuracy score), and a “Done” button.
* **Plant Information Screen**: Presents detailed records, including species information, sighting details, comments, and user actions.

A screenshot of a dashboard

AI-generated content may be incorrect.

**Admin Screens**

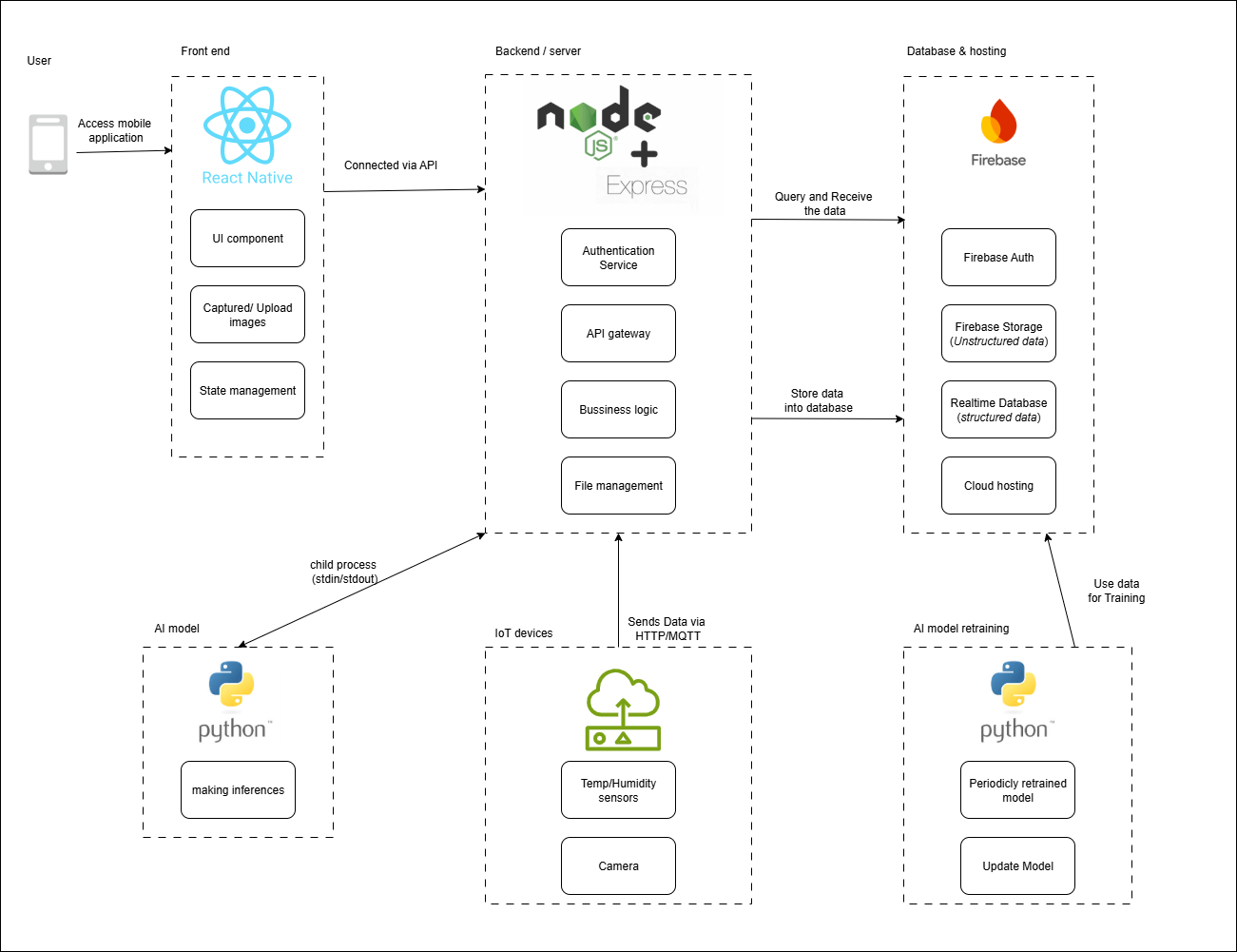
* **Admin Dashboard**: Offers modules for account management, mail management, warnings, and current biodiversity statistics.
* **IoT Dashboard**: Displays live monitoring and detection history data, with filtering options, date range selection, and a paginated data table for browsing historical records.

**Navigation Flow**

Navigation between screens follows a structured and role-based flow. Users begin at the onboarding and login/register screens, after which they are directed to the main dashboard. From the dashboard, users can access map visualizations, capture identifications, detailed species information, or view the community posts posted by other users.

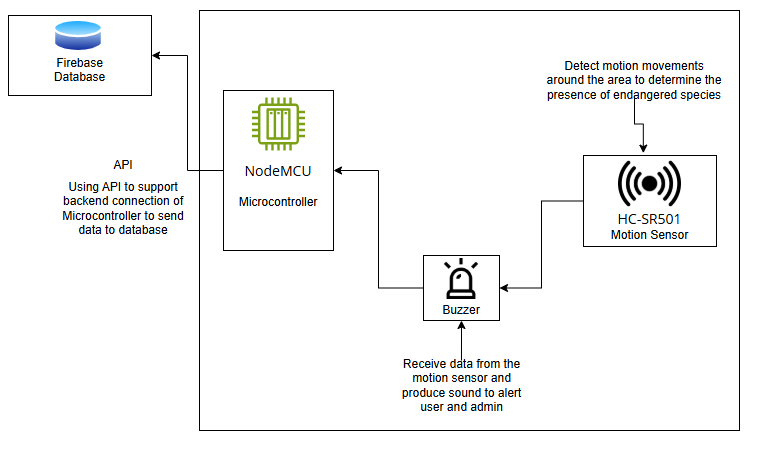
Administrators follow a parallel path but are directed to the Admin Dashboard, from which they can manage accounts, process mails, review warnings, and monitor IoT data through the IoT Dashboard.

**6.1.2 System Diagram**



This software architecture powers a mobile plant classification system using a three-tier design. The React Native frontend provides the user interface for capturing plant images and managing state. Requests are routed through the API Gateway, which directs them to the right backend services. The Node.js backend acts as the application layer: the Authentication & Authorization module (via both Firebase Auth and custom Node.js auth) manages custom authentication and role-based authorization for normal users, plant expert and IT administrators. It applies business logic and serves as the central hub for AI model inference, classifying plant species using trained deep learning models. IoT devices (temperature, humidity sensors, and cameras) send real-time data to the Node.js backend via HTTP or MQTT. The backend interacts with Firebase services, including Realtime Database for structured data, Storage for plant images. A dedicated AI Training Service runs independently, periodically retrieving data from Firebase to retrain the plant classification model using TensorFlow/PyTorch. The newly trained model is then redeployed to the inference module in the backend, ensuring continuous improvement without disrupting live services.

**6.1.3 Hardware Diagram [Gibson] (Draft)**



The system is designed to detect motion in an area and provide both immediate and remote alerts to support the monitoring of endangered species. At the core of the system is the NodeMCU microcontroller, which serves as the central processing unit. It receives input from the HC-SR501 motion sensor, a low-cost PIR (Passive Infrared) sensor that detects movements in the environment. When motion is detected, the NodeMCU triggers a buzzer to produce a sound that provides a local alert for users and administrators. At the same time, the NodeMCU connects to a Firebase database via an API to send motion detection data to the cloud. This enables the storage of structured information that can be accessed remotely for monitoring and analysis. Through Firebase, researchers and administrators can review historical data, track patterns of movement, and confirm the presence of endangered species in the monitored area. Overall, the system combines real-time alerts with cloud-based data storage, making it both a practical and scalable solution for wildlife monitoring.

**6.1.4 Use Case Diagram**

The planned Plant Information & Monitoring System will include various features that can be implemented, but not all of them will be available to all users. To maintain effective access control, the system's users have been divided into three categories: admin, user, and plant expert. Each category will be allocated specific use cases or functionalities, while certain aspects may overlap and be shared by various user roles. The Use Case Diagram shows these interactions in a simpler manner, allowing stakeholders to better understand the scope of the system and the unique responsibilities of each user group.

A diagram of a company

AI-generated content may be incorrect.

**6.1.5 Activity Diagram**

A diagram of a software system

AI-generated content may be incorrect.The activity diagram explains the workflow for the main process of the plant identification.

**6.2 System Features**

The Smart plant system provides several main features

|  |  |
| --- | --- |
| **System Features** | **Description** |
| AI Plant Identification | User upload images to get identification with confidence levels by the AI |
| Biodiversity Map | Displays plant distribution and conservation status, with sensitive data protection |
| Verification Workflow | Low-confidence identifications are reviewed by administrators |
| IoT Monitoring | Sensors track environmental conditions and alert officers of potential threats |
| Security Features | Encryption, MFA and RBAC protect data and system access |

**6.2.1 System Purpose [Esther] (drafttttt)**

The purpose of our system is to provide a safe, reliable and community-driven platform that supports the identification, mapping and preservation of plant biodiversity in Sarawak. Traditional methods of plant identification often require expert knowledge or access to herbarium collections, which are not easily available to the public. Other plant identification applications are designed for a worldwide audience and are not concentrate on the Sarawak plants, especially endangered species. Hence, our system aims to provide a straightforward and solution to bridge that gap.

Users can take images of plants and upload them to the system, enabling the community to get involved in biodiversity conservation. An artificial intelligence model is built to identify plant species. Expert validation is used to guarantee the accuracy of the plant, especially in cases where the AI's confidence level is lower than 70%. The user and researchers can both benefit from the trustworthy and large biodiversity database which is produced by combining crowdsourcing, automation, and expert review.

In addition to identifying plants, the system monitors plant conditions in real time using Internet of Things (IoT) devices, such as temperature and humidity sensors. This data helps track the health of plants and detect environmental changes that may threaten local ecosystems. AI, IoT and cloud hosting make the system a comprehensive application for protecting plants and their environments.

Finally, the goal of the SmartPlant Sarawak system is to raise awareness and encourage more people to participate in conservation efforts, in addition to safeguarding endangered plants. The system serves as a learning platform and conservation tool since researchers and educators can use the data to promote scientific research and instructional programs.

**6.2.2 Processes, Procedures, Standards**

The development of the SmartPlant system follows structured processes, clearly defined procedures, and internationally recognized standards to ensure the final product is reliable, secure, and user-friendly. The project adopts an Agile methodology, where development is divided into short iterations that allow for continuous feedback and refinement. This process ensures that features are delivered incrementally and tested early, reducing risks of failure and aligning the system with user needs. In parallel, the Secure Software Development Life Cycle (SSDLC) is applied across all phases, embedding security considerations into planning, design, coding, testing, and deployment.

To maintain consistency and quality, specific procedures are followed throughout development. All source code is managed using version control (GitHub), enabling collaboration, issue tracking, and rollback when needed. A code review process ensures that each feature is checked for quality, efficiency, and adherence to best practices. Testing procedures are systematic, covering functional, usability, and security aspects, while data handling follows strict guidelines to protect sensitive information such as the GPS locations of endangered plants.

The system also conforms to established standards. Usability is guided by ISO 9241-210, ensuring the application provides an intuitive, accessible, and user-centered experience. Coding practices are aligned with widely accepted programming standards that emphasize modularity, documentation, and maintainability. Security measures follow OWASP recommendations, with SSL/TLS encryption, role-based access control (RBAC), and multi-factor authentication (MFA) applied to safeguard users and sensitive biodiversity data. Together, these processes, procedures, and standards create a development environment that promotes quality, security, and sustainability in the SmartPlant system.

**7.0 Quality Management**

Quality management ensures that the SmartPlant system meets its functional, usability, and security requirements while also delivering a reliable and user-friendly experience. The approach to quality in this project is guided by the principle of “built-in quality,” meaning that quality assurance is applied at every stage of development rather than only at the end. By combining Agile practices with Secure Software Development Life Cycle (SSDLC) principles, the project ensures that testing, validation, and review are ongoing processes that strengthen the final product.

To define the boundaries of quality, the project team applies a Definition of Done (DoD) for each deliverable. A feature is only considered complete when it has been implemented, reviewed, tested, and documented. This prevents incomplete or unstable components from being accepted into the system. Quality attributes are also prioritized, including accuracy of plant identification, security of user and biodiversity data, system reliability, and ease of use. Each attribute is measurable and directly contributes to the success of the project.

The quality management process also integrates continuous testing. Functional tests confirm that the AI model, mobile app, and IoT integration operate as intended. Usability tests ensure that the interface is intuitive for both new and experienced users. Security testing identifies vulnerabilities such as unauthorized access or data leakage, allowing the team to apply fixes promptly. Performance tests are carried out to check system responsiveness, particularly in AI identification speed and IoT data synchronization.

Quality control is further maintained through documentation and team procedures. Every change in code is reviewed and tracked in GitHub, ensuring traceability and accountability. Bugs and issues are recorded, categorized by severity, and resolved before release. Regular team meetings provide opportunities to review progress, evaluate risks, and adjust strategies. By combining these practices, the SmartPlant project maintains a high level of quality and ensures that the final system not only meets technical requirements but also provides real value to its users.

During Sprint 1 and Sprint 2, the implementation of the SmartPlant system will focus on developing the core plant identification features, mobile application interface, database setup, and IoT integration. Quality will be ensured by applying the Definition of Done (DoD) for each backlog item, making sure that every function is implemented, tested, reviewed, and documented before being accepted.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product Backlog No.** | **Product Backlog Items** | **Main Characteristic** | **Sub-Characteristic** | **Definition of Done (DoD)** |
| 1 | Allow user to upload plant photo for AI identification | Functional Suitability | Functional Correctness | AI model correctly identifies plant with ≥70% accuracy in test cases |
| Performance Efficiency | Time Behaviour | Identification results displayed within 5 seconds |
| 2 | Display biodiversity map with plant locations | Usability | User Interface Aesthetics | Map loads without errors and shows plant markers with clear labels |
| Reliability | Fault Tolerance | Map continues to function even if one data source is unavailable |
| 3 | Enable admin to verify uncertain AI results | Functional Suitability | Functional Completeness | Verification workflow stores corrected data in the database |
| Security | Integrity | Only authorized admin can access verification dashboard |
| 4 | IoT sensors send environmental data to system | Reliability | Maturity | Sensor data updates received at regular intervals without disruption |
| Security | Confidentiality | IoT data is encrypted in transmission (SSL/TLS) |
| 5 | Secure user login with MFA | Security | Authentication | Users must pass MFA before accessing the system |
| Usability | Learnability | Login process is intuitive and tested with sample users |

**8.0 Resources**

This section outlines the hardware, software, human and non-human resources that will be utilised to successfully complete the SmartPlant Sarawak project. Proper allocation and management of these resources are critical to ensure that the project is delivered on time and meets its objectives.

**Software Resources**

The project will leverage a range of modern software tools and platforms to facilitate development, collaboration and deployment.

|  |  |  |
| --- | --- | --- |
| Category | Tool/Technology | Purpose in project |
| Development environment | Visual Studio Code | Primary code editor for frontend, backend and AI development. |
| Mobile app framework | React Native | To build a cross-platform mobile application for both Android and iOS from a single codebase. |
| Backend framework | Node.js with Express.js | To develop a scalable and secure backend API for handling data, authentication and AI service requests. |
| Database and cloud services | Google Firebase | Used for user authentication (Firebase Auth), a real-time NoSQL database (Firestore) and cloud storage (Firebase Storage) for images. |
| AI and machine learning | Python with TensorFlow/PyTorch | To train, test and deploy the custom AI model for plant species identification. |
| Version control | Git and GitHub | For source code management, collaboration, issue tracking and maintaining version history. |
| Project management | Confluence, Jira | To manage project documentation, track risks and organise development tasks within an Agile framework. |
| UI/UX design | Figma | To create wireframes, mockups and interactive prototypes for the mobile application interface. |
| API testing | Postman | To test and validate the backend API endpoints, ensuring reliability and correctness. |
| Security testing | OWASP ZAP, Burp Suite | For conducting vulnerability scanning and penetration testing to identify and mitigate security risks. |

Table 2: Software tools or technologies and its purpose in project

**Hardware Resources**

The project requires specific hardware for development, testing and implementing the IoT monitoring features.

|  |  |  |
| --- | --- | --- |
| Category | Item | Purpose in project |
| Development | Developer laptops/PCs | Team members' personal computers for software development, coding and running simulations. |
| Mobile testing | Android and iOS smartphones | For deploying and testing the mobile application on physical devices to ensure functionality, performance and usability. |
| IoT components | Arduino UNO microcontroller | The central processing unit for the IoT device, connecting sensors to the network. |
| PIR motion sensor | To detect motion in the monitored environment, triggering alerts for potential threats. |
| Buzzer, wires, breadboard | To create a physical alert system and assemble the IoT prototype. |

Table 3: Hardware items and its purpose in project

**Human Resources**

The project team is composed of members with diverse technical skills, supported by academic supervisors.

|  |  |
| --- | --- |
| Role | Expertise/Contribution |
| Project team | The team consists of seven members with expertise across Software Development and Artificial Intelligence. They are responsible for the entire project lifecycle, from planning and design to development, testing and final delivery. |
| Team leader | Oversees project planning, task coordination and ensures the team stays on track with milestones and deliverables. |
| Project tutors | The project is supervised by Ts. Dr. Lee Sue Han, Dr. Kelvin Yong, Dr. Mark Tee and Dr. Fu Swee Tee, who provide academic guidance, technical expertise and mentorship throughout the project. |

Table 4: Roles and their expertise/contribution

**Non-Human Resources**

These resources are essential for research, training the AI model and ensuring the project is built on a solid foundation of knowledge.

|  |  |  |
| --- | --- | --- |
| Category | Description | Purpose in project |
| Datasets | Publicly available plant image datasets (e.g., from Kaggle, GBIF) and curated local plant images from Sarawak. | To provide the necessary training and validation data for developing an accurate AI plant identification model. |
| Academic resources | University library, online academic journals (e.g., IEEE Xplore, ACM Digital Library) and research papers. | To conduct background research on biodiversity, AI in conservation, IoT monitoring and cybersecurity best practices. |
| Technical documentation | Official documentation for React Native, Firebase, Node.js and other software frameworks being used. | To serve as a primary reference for implementation, troubleshooting and adhering to best practices. |

Table 5: Non-human resources and its purpose in project

**9.0 Approval Signatures:**

**9.1 Project Team (签名等要交之前才签 怕画了会被移掉)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Name of student** | **Student ID** | **Signature** | **Roles** |
| 1 | You Wee LIEW | 102786467 |  | Team Leader |
| 2 | Natalie ROBERT | 102787350 |  | Team Member |
| 3 | Esther Ee Qian LING | 104381903 |  | Team Member |
| 4 | Li Ying YEO | 102789314 |  | Team Member |
| 5 | Elisa Bui Sian PUI | 104388991 |  | Team Member |
| 6 | Gibson Chee Yii CHONG | 102779580 |  | Team Member |
| 7 | Daniel Meng Yeu TIONG | 102777801 |  | Team Member |

(Scrum Master, Product Owner, Developer)

**9.2 Project Sponsor [Your Tutor]**

|  |  |
| --- | --- |
| Tutor’s name (on behalf of the client)  xxx | Signature: |
|  |  |

**References**

iNaturalist Australia n.d., “Making sure your observations are shared to GBIF”, iNaturalist Australia, viewed 8 September 2025, <https://inaturalist.ala.org.au/projects/flowering-plants-of-turkey/journal/29105>.

PlantNet Case Study | Citizen Science Observatory 2024, “PlantNet Case Study | Citizen Science Observatory”, ArangoDB, viewed 7 September 2025, <https://arangodb.com/solutions/case-studies/meet-plantnet-one-of-the-largest-citizen-science-observatories-in-the-world/>.

Pl@ntNet 2025, “Read more - Pl@ntNet,” Pl@ntNet, viewed 7 September 2025, <https://plantnet.org/en/about/>.

Wong, KM & Neo, L 2019, “Species richness, lineages, geography, and the forest matrix: Borneo's 'Middle Sarawak' phenomenon, Gardens’ Bulletin Singapore”, *ResearchGate*, vol. 71, pp. 463–496.