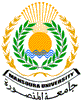
|  |  |  |  |
| --- | --- | --- | --- |
| Faculty of Computers and Information Sciences Computer Science Dept. |  |  | Mansoura University |



**Releaf**

**Application for Afforestation and Home Agriculture using Ai**

**by**

1.Amr Ahmed Elbakary 2. Shahd Ashraf Mustafa

3.Aisha Mohamed El-Sayed 4. Mohamed Hassan Saber

5.Younes Mohamed Ibrahim 6. Amr El-Sayed Tawfiq

7.Mohamed Elsayed Ali 8. Omar Mohamed Reda

9.Rawan Tarek Abd Elrazik 10. Mohammed El-Haj

**A project submitted in partial fulfillment of the requirements for the degree of Bachelor of Sciences in Information Technology**

**Supervised by**

Dr. Reem Eldeeb

Eng. Hassan Khaled

2024 - 2025

**Abstract**

Environmental degradation and urbanization have significantly reduced green spaces, contributing to climate change and air pollution. This project introduces a mobile application designed to enhance tree-planting efforts and increase greenery in Egypt. The app leverages AI to provide users with interactive planting maps, educational resources, and growth tracking features. It encourages community participation through volunteering, donations, and home gardening guidance. By integrating technology with environmental sustainability, the app aims to facilitate large-scale afforestation, improve air quality, and promote eco-friendly urban development. Ultimately, this initiative supports global sustainability goals while fostering social and economic benefits.

The project delivers a mobile application designed to:

* Display interactive maps highlighting areas in need of afforestation, whether in cities, villages, or public spaces.
* Recommend plant species that are optimal for specific environmental conditions, such as climate, soil type, and pollution levels.
* Offer step-by-step instructions on planting and maintenance practices to ensure successful cultivation.
* Record and track the growth of trees and plants, providing users with notifications and reminders for care and watering.
* Provide a built-in chatbot that offers real-time support, answering user questions about suitable tree species, planting methods, care instructions, and location-based recommendations.

 The application focuses on Egypt, addressing both urban areas that face ecological challenges. By leveraging AI and digital mapping technologies, the project establishes clear boundaries by focusing on planting-related solutions while promoting environmental awareness and engagement.

Table of Content

[Abstract 2](#_Toc435734391)

[1 Chapter 1: Introduction 7](#_Toc435734396)

[1.1 Introduction 8](#_Toc435734397)

[1.2 Problem Definition 8](#_Toc435734398)

[1.3 Project Objectives 9](#_Toc435734399)

[1.4 Contributions of This Study 10](#_Toc435734401)

[1.5 Document Organization 10](#_Toc435734403)

[2 Chapter 2: Literature Review 12](#_Toc435734404)

[2.1 Introduction 13](#_Toc435734405)

[2.2 Background 13](#_Toc435734406)

[2.3 Review of Relevant Work 17](#_Toc435734407)

[2.4 Relationship between the Relevant Work and Our Own Work 21](#_Toc435734408)

[2.5 Summary 21](#_Toc435734409)

[3 Chapter 3: System Analysis 22](#_Toc435734410)

[3.1 Introduction 23](#_Toc435734411)

[3.2 System Requirements 23](#_Toc435734413)

[3.2.1 Functional Requirements 24](#_Toc435734414)

[3.2.2 Non Functional Requirements 26](#_Toc435734415)

[3.2.3 User Requirements 27](#_Toc435734417)

[3.3 System Architecture 28](#_Toc435734418)

[3.3.1 User Interface 25](#_Toc435734414)

[3.3.2 Server Interface 26](#_Toc435734415)

[3.3.3 Administrator Interface 26](#_Toc435734417)

[3.4 Development Methodology 29](#_Toc435734419)

[3.5 Tools and Languages 30](#_Toc435734423)

3.6 Buisness Model ……………………………………………………………………………….………………31

[3.7 Summary 31](#_Toc435734424)

[4 Chapter 4: System Design 32](#_Toc435734425)

[4.1 Introduction 33](#_Toc435734426)

[4.2 System Archetucture 34](#_Toc435734427)

[4.3 Development Methods 34](#_Toc435734427)

[4.4 UseCase Diagram 35](#_Toc435734427)

[4.5 Logical and Physical Database Design 39](#_Toc435734427)

[4.6 Squence Diagram 40](#_Toc435734428)

[4.7 Activity Diagram 50](#_Toc435734428)

[4.8 Interface Design 53](#_Toc435734429)

[4.9 Summary 55](#_Toc435734430)

5 [Chapter 5: System Implementation 56](#_Toc435734425)

5[.1 Introduction 57](#_Toc435734426)

[5.2 Backend Implementation 57](#_Toc435734427)

[5.3 Plant Identification Model 60.](#_Toc435734427)

[5.4 Plant Disease Diagnosis Model 61](#_Toc435734427)

[5.5 Flutter Application Interface 62](#_Toc435734428)

[5.6 Web Interface 65](#_Toc435734428)

[6 Chapter 6: Conclusion and Future Work 67](#_Toc435734425)

[6.1 Introduction 68](#_Toc435734426)

[6.2 Overall Weaknesses 68](#_Toc435734427)

[6.3 Overall Strengths 68](#_Toc435734427)

[6.4 Future Work 68](#_Toc435734427)

[**References** 69](#_Toc435734428)

**List of Figures**

Figure 2.3 Relationship between the Relevant Work and Our Work …………..………….……21

Figure 3.6 Shows Business Model……………………………………………………..…..……31

Figure 4.2.1 Shows System Architecture………………………………………….…….………34

Figure 4.4.1 Shows Actor Symbol………………………………………………………………35

Figure 4.4.2 Shows Use Case Symbol……………………………………………………..……36

Figure 4.4.3 Shows System Boundary Symbol…………………………………………………36

Figure 4.4.4 Shows Generalization Relationship Symbol………………………………………36

Figure 4.4.5 Shows Association Relationship Symbol…………………………………….……37

Figure 4.4.6 Shows Use Case Diagram…………………………………………………………37

Figure 4.5 Shows Database Design (ERD)…………………………………..…………………39

Figure 4.6.1.1 Shows Actor in Sequence Diagram……………………………………...………40

Figure 4.6.1.2 Shows Lifeline…………………………………………………………..………41

Figure 4.6.1.3 Shows Activation………………………………………………………..………42

Figure 4.6.1.4 Shows Call Message…………………………………………………….………43

Figure 4.6.1.5 Shows Reply Message……………………………………………………..……43

Figure 4.6.1.6 Shows Asynchronous Message……………………………………….…………44

Figure 4.6.1.7 Shows Alt Fragment…………………………………………………..…………44

Figure 4.6.1.8 Shows Sequence Diagram - Image Scanning……………………………………45

Figure 4.6.1.9 Shows Sequence Diagram - Discover Trees………………………….…………46

Figure 4.6.1.10 Shows Sequence Diagram - View My Garden…………………………………47

Figure 4.6.1.11 Shows Sequence Diagram - Browse Map………………………...……………48

Figure 4.6.1.12 Shows Sequence Diagram - Campaigns……………………….………………49

Figure 4.7.1 Shows Activity Diagram act1…………………………………………...…………50

Figure 4.7.2 Shows Activity Diagram act2………………………………………...……………51

Figure 4.7.3 Shows Activity Diagram act3………………………………………...……………52

**List of Tables**

Table 3.2.1.1 outlines the functional requirements of the system………………………………25

Table 3.2.1.2 outlines the non-functional requirements of the system……………..……………26

|  |
| --- |
| **Chapter 1** |

**Chapter 1: Introduction**

**1. Introduction**

The importance of green spaces in urban and rural areas cannot be overstated. Green spaces improve air quality, reduce urban heat, and provide aesthetic and psychological benefits to communities. However, many cities face rapid urbanization, deforestation, and pollution, leading to a significant decline in green areas.  
This project addresses these pressing issues through an innovative application that integrates Artificial Intelligence (AI), chatbot assistance, and interactive digital mapping technologies. By enabling users to identify suitable locations for planting, receive personalized care recommendations for plants, interact with a chatbot for real-time guidance, and participate in community-based green initiatives, the application bridges the gap between environmental needs and public engagement.  
Designed as a comprehensive tool, the application not only facilitates planting activities but also empowers users with real-time insights and tailored support. By promoting ecological restoration, sustainable practices, and environmental awareness, this initiative contributes to global efforts toward sustainability and climate resilience while fostering a greener, more engaged society in Egypt.

**1.2 Problem Definition**

Urbanization and industrial activities have significantly reduced green spaces and exacerbated pollution levels in Egypt. These changes have led to an increase in urban heat islands, deterioration of air quality, and loss of biodiversity. Despite growing awareness of these issues, afforestation efforts remain limited due to a lack of accessible tools and resources for individuals and organizations.

**This project addresses several pressing questions:**

* How can AI and digital mapping tools facilitate afforestation in urban and rural areas?
* What role can technology play in identifying optimal planting locations and suitable plant species?
* What strategies can be implemented to make afforestation accessible to non-experts?
* How can community participation in planting initiatives be encouraged?

By tackling these questions, the study seeks to develop a user-friendly application that promotes environmental restoration and fosters a culture of sustainability.

**1.3 Project Objectives**

The proposed project aims to achieve the following objectives:

* To identify and prioritize suitable areas for planting using digital maps and environmental data.
* To study the environmental and social benefits of afforestation in reducing pollution and enhancing urban landscapes.
* To develop an interactive application that offers personalized planting recommendations based on local climate.
* To provide educational resources and tools that empower individuals and organizations to participate in afforestation efforts effectively.
* To encourage community engagement through awareness campaigns, volunteering opportunities, and accessible guides for home gardening and public planting initiatives
* To integrate a chatbot that provides real-time guidance and answers to users’ questions about planting techniques, suitable tree species, and maintenance based on their specific location and needs.

**1.4 Contributions of This Study**

This project contributes significantly to sustainable development and ecological restoration by:  
 - Providing a scalable and practical tool for individuals and organizations to engage in afforestation and environmental conservation efforts.  
 - Enhancing air quality and mitigating the effects of pollution through increased green coverage in urban and rural areas.  
 - Addressing gaps in existing afforestation initiatives by offering a unique combination of AI-driven recommendations and user-friendly mapping technologies.  
 - Promoting a culture of environmental stewardship by empowering users with the knowledge and tools necessary for sustainable planting practices.

**1.5 Document Organization**

This project consists of six chapters in addition to one appendix. These chapters are organized to reflect the scientific steps toward our main objective. A brief description of the contents of each chapter is provided below:

* Chapter 1 introduces the project objectives, the motivation behind the project,the contributions made by the project, the scope of work, and the overall project layout.
* Chapter 2 presents the literature review, offering an overview of the current systems or solutions already existing in the field related to our topic. It discusses their main features, strengths, and limitations.
* Chapter 3 focuses on system analysis, detailing all the steps undertaken to analyze the system comprehensively.
* Chapter 4 discusses the system design, highlighting all the design steps and including snapshots to illustrate the design process.
* Chapter 5 covers system implementation, describing all the implementation steps supported by relevant snapshots.
* Chapter 6 provides the conclusion and future work, summarizing the project outcomes and suggesting possible directions for future enhancements.

|  |
| --- |
| **Chapter 2** |

**Chapter 2: Literature Review**

In this chapter, we establish the context for Releaf by examining relevant theories, existing works, historical contexts, and previous studies pertinent to tree planting, AI-based plant care, and home agriculture. The chapter highlights the relationships between Releaf and other works in similar domains while emphasizing the app’s unique contributions. A comprehensive review of primary and secondary sources substantiates the theoretical, practical, and research foundations of the project.

**2.1 Background**

The Releaf application merges Artificial Intelligence, mobile technology, and community engagement tools to address modern environmental challenges. It uniquely combines features for tree planting, AI-based plant identification, and home agriculture guidance to promote sustainability and enhance green spaces. This section provides evidence supporting the theoretical and research background of the study by discussing critical topics such as:

* **Home Agriculture and Gardening Tips.**
* **AI-Powered Plant Identification.**
* **Interactive Planting Maps.**
* **Community Participation.**
* **Chatbot-Based Planting Guidance**

**2.1.1 Home Agriculture and Gardening Tips**

Releaf promotes sustainability by supporting home agriculture and afforestation efforts. The app offers personalized gardening guides for growing a wide range of plants, including trees, vegetables, flowers, and herbs, suited for various environments like balconies, rooftops, and backyards. In addition to typical gardening, Releaf encourages tree planting for afforestation, providing users with tailored recommendations for tree species that are suitable for local conditions and spaces. The app also sends reminders for tree care tasks, including watering and fertilizing. By integrating home gardening and tree planting, Releaf empowers users to contribute to greening efforts and support environmental restoration through local afforestation initiatives.

1. **Tips for Home Gardening:**
   1. Step-by-step guides for growing different types of plants, including flowers, vegetables, and herbs.
   2. Best practices for soil preparation, watering schedules, and pest control tailored for indoor and outdoor home gardens.

**2. Types of Plants:**

* + 1. A comprehensive catalog of plants categorized by their care requirements, space needs, season ,soil pH and growth conditions.
    2. Estimated plant growth timelines that inform users of how long each plant takes to germinate, mature, and reach harvest or full growth. This helps users plan their gardening activities and set realistic expectations based on the plant’s lifecycle.

**2.1.2 AI-Powered Plant Identification**

AI-powered plant identification has revolutionized how users can recognize and care for plants. This technology uses advanced algorithms and image processing techniques to identify plants accurately from user-uploaded images. AI-driven systems enable users to identify plants even in varying lighting conditions, improving the accuracy and reliability of plant recognition.

Once a plant is identified, the system can provide users with personalized care plans, including watering schedules, fertilization, and pruning advice, tailored to the plant's specific needs. Additionally,AI-powered systems can monitor plant health.

**2.1.3 Interactive Planting Maps**

Releaf features interactive planting maps that identify areas in need of afforestation based on environmental factors such as climate and pollution levels. The map highlights these regions, guiding users to areas where tree planting can make the most impact. Additionally, the app suggests tree species that are well-suited for planting in these locations, considering factors like local climate and soil conditions.

**2.1.4 Community Participation**

Releaf plays an active role in promoting environmental awareness by organizing and supporting targeted campaigns focused on afforestation and green living. The app provides users with access to ongoing tree-planting campaigns and environmental awareness events. These campaigns are designed to encourage participation, and create a sense of shared responsibility for environmental restoration.

Through in-app notifications and location-based suggestions, users are invited to take part in both local and national campaigns. By making campaign participation easy and accessible, Releaf fosters a culture of environmental action and community involvement, helping to build a greener and more sustainable future.

**2.1.5 Chatbot-Based Planting Guidance**

One of Releaf’s key innovations is its integrated chatbot, designed to offer users real-time support throughout their planting journey, context-aware responses related to tree planting and plant care.

This intelligent assistant simplifies the afforestation process by:

* Answering user questions on plant selection, soil types, watering schedules, and pest control.
* Recommending suitable tree species based on the user’s geographic location, space constraints, and environmental conditions.
* Providing reminders and care tips tailored to seasonal changes and weather patterns.
* Guiding beginners through planting steps with friendly, conversational instructions that eliminate the need for technical knowledge.

Unlike traditional static guides, the chatbot enables dynamic, two-way interaction that personalizes the user experience. This ensures that users, regardless of their expertise level, feel supported and empowered to contribute to environmental restoration.

**2.2 Related Works**

**1. Picture This**



“PictureThis” is a plant identification app that uses AI to recognize plants from pictures. It also offers care instructions and connects users to a plant community.

**Pros:**

* Fast and accurate plant identification with a large database.
* Provides plant care instructions and reminders.
* Features a plant journal to track plant growth and health.

**Cons:**

* Focuses mostly on plant identification and care, lacking environmental impact features.
* Does not prioritize privacy or transparency regarding user data.
* Lacks functionality like community-driven planting efforts or environmental conservation features.

**2. Garden Tags**



“Garden Tags” is a plant identification and gardening app that allows users to identify plants by uploading photos, share gardening tips, and track plant care.

**Pros:**

* Provides accurate plant identification through AI.
* Offers gardening advice and care tips based on plant species.
* Includes a plant care reminder feature to help users keep track of their plant needs.
* Has a community feature for sharing gardening experiences and tips.

**Cons:**

* May not cover all plant species.
* Limited in terms of integrating environmental or ecological features (like interactive maps for planting locations).
* Cannot benefit from more interactive features (e.g., AI-based recommendations for tree planting in specific areas).

**3. Plant Net**



“PlantNet” helps users identify plants through photos and community contributions, offering detailed plant information and offline functionality.

**Pros:**

* Extensive plant identification database.
* Community contributions improve the app.
* Offline access for fieldwork.

**Cons:**

* No interactive planting maps.
* Lacks volunteer and donation features.
* No challenges or rewards for users.

**4. Tree Planet**



“Tree Planet” allows users to plant virtual and real trees while tracking their growth and participating in gamified challenges.

**Pros:**

* Combines virtual and real tree planting.
* Gamification and rewards increase engagement.
* Provides educational content on trees.

**Cons:**

* No AI-powered plant identification.
* Lacks interactive maps and volunteer features.
* Missing plant care guidance.

**2.****3** **Relationship between the Relevant Work and Our Work**

In this table, we will explain some of the features of the applications we talked about. It is a simplified table through which you can understand what we were talking about in these applications:

(Figure 2.3) Relationship between the Relevant Work and Our Work  
A screenshot of a computer

AI-generated content may be incorrect.

**2.4 Summary**

This chapter has provided an overview of the foundational tools, methodologies, and related work that influenced the development of Releaf. The comparative analysis illustrates the app’s distinct features, such as integrating AI-driven plant identification, home gardening tips, and interactive planting maps, setting it apart from existing solutions

|  |
| --- |
| **Chapter 3** |

**Chapter 3: System Analysis**

**3.1 Introduction**

System Analysis represents a systematic and structured methodology used to understand, define, and implement systems that align with specific organizational or functional objectives. This chapter serves as a detailed guide to System Analysis, highlighting its fundamental elements, methodologies, and importance in the realm of system development.

The primary purpose of System Analysis is to gain an in-depth understanding of the requirements and complexities of a system, enabling the creation of effective solutions tailored to address specific challenges or needs. This process involves a meticulous examination of various aspects, including functional, non-functional, and user requirements, to ensure all perspectives are considered.

By integrating best practices and proven techniques, System Analysis lays the groundwork for developing systems that are not only efficient but also adaptable to evolving demands. This chapter aims to equip readers with the essential knowledge and tools needed to excel in this critical phase of system development.

**3.2 System Requirements**

The System Requirements outline the essential specifications and functionalities required for developing a comprehensive Plant Identification and a map that displays detailed information about the plants. These requirements serve as a framework for system architects, developers, and stakeholders, providing clear guidance to ensure the creation of a practical and efficient solution.

System requirements represent the key configurations and features necessary for the system to operate effectively and meet its intended goals. This section is organized to provide a clear understanding of these requirements, fostering alignment among all parties involved in the development process.

The subsequent subsections will delve into the functional requirements, non-functional requirements, and user requirements, each of which is crucial to shaping the system's functionality and ensuring its performance meets user needs.

**3.2.1 Functional Requirements**

Functional requirements specify the essential functions, features, and capabilities that a system, software application, or product must perform to satisfy user needs and fulfill its intended purpose. These requirements define how the system operates, its expected behavior, and the interactions users can have with it. By addressing the question, "What functions does the system need to perform?" functional requirements provide clarity and direction for system development.

In the context of a Plant Identification system that incorporates a map displaying detailed information about plants, functional requirements may include features such as accurate identification of plant species and seamless navigation of the interactive map. These features are crucial to ensuring the system's usability and effectiveness.

Table 3.2.1.1 outlines the functional requirements of the system

|  |  |
| --- | --- |
| **Functional Requirements** | **Description** |
| **Scan/Upload Image for Identification** | Allow users to scan or upload images for plant identification. |
| **Identify Plant Type** | Provide information about plants based on user input image. |
| **Explore plant diseases** | show information about plant diseases to avoid it . |
| **Upgrade for Premium** | Offer users an option to upgrade to a premium version with additional features. |
| **View My Garden** | Enable users to view and manage their personal garden space. |
| **Browse Map** | Provide a map for users to browse different locations and view information about a specific area and its potential for afforestation |
| **Register for Campaigns** | Enable users to register for environmental campaigns. |
| **Explore Events and Campaigns** | Provide information about events and Campaigns activities related to gardening. |
| **View Information About Plants** | Provide detailed information about different plants. |
| **Set Reminder** | Allow users to set reminders for various gardening tasks. |

**3.2.2 Non-Functional Requirements**

The following non-functional requirements ensure that the application performs reliably, securely, and efficiently while maintaining user satisfaction and system integrity.

Table 3.2.1.2 outlines the non-functional requirements of the system

|  |  |
| --- | --- |
| **Non-Functional**  **Requirements** | **Description** |
| **Performance** | The system shall have fast response times for user actions and load pages within a few seconds. |
| **Usability** | The application shall provide an intuitive and easy-to-use interface for all age groups. |
| **Security** | User data shall be protected through encryption, and secure authentication must be implemented. |
| **Scalability** | The system must support a growing number of users without affecting performance. |
| **Compatibility** | The application shall run on Android platforms across common versions. |
| **Reliability** | The application uptime shall be 99.9%, ensuring high availability for users. |
| **Maintainability** | The system shall allow easy updates and enhancements with minimal downtime. |
| **Availability** | The app shall be available 24/7 except for scheduled maintenance periods. |

**3.2.3 User Requirements**

The following are the primary requirements expected by users of the application, including features to identify polluted areas and provide actionable suggestions for tree planting to reduce pollution and improve air quality.

**Registration:** Users should be able to register using email or social media accounts.

**Login/Authentication:** Users should log in securely with credentials or alternative methods.

**Map Browsing:** Users should view a map displaying areas with high pollution or elevated temperatures and receive suggestions for tree planting in those regions.

**Area Selection and Suitability Analysis**: Users should be able to select or tap on a specific area on the map, after which the app provides detailed information about that location’s environmental conditions—such as pollution levels, soil quality, and climate—and evaluates its suitability for planting, along with recommended plant species.

**Plant Identification:** Users should scan or upload images of plants for identification, including detecting plant diseases.

**My Garden Management:** Users should create and manage a personal garden, add favorite plants, set reminders, and track growth progress.

**Community Features:** Users should participate in challenges, register for campaigns, and engage with a community of environmentally conscious individuals.

**Notifications:** Users should receive notifications and reminders about tree planting activities or maintenance.

**Premium Services:** Users should have the option to upgrade to premium features, such as advanced analytics or disease detection.

**Chatbot Assistance**: Users should interact with a built-in chatbot to receive real-time support for plant selection, care tips, environmental advice, and answers to general questions based on location and user preferences.

**3.3 System Architecture**

System architecture which defines the structure, behavior and more views of the system that will be explained in three views:

1. User Interface (3.6.1)
2. Server Interface (3.6.2)
3. Administrator Interface (3.6.3)

**3.3.1 User Interface**

The user signs up using a unique name, a valid email, and a secure password that includes letters, numbers, and special characters. After signing up, they can log in using their email and password. The app is available on mobile devices.

After logging in, the user can select a location on the map, use GPS to detect their current location. The app shows environmental information like air quality and pollution levels, then suggests suitable trees for planting in that area. The user can ask the chatbot for help with planting tips and care instructions. They can explore different types of plants, view details about each plant, and find out which are suitable for their space and climate. The user can also add plants to their personal garden, set reminders, and track growth. Users can join planting campaigns. The interface is simple, with clearly labeled options for login, location selection, exploring plants, getting recommendations, chatbot help, and campaign participation.

**3.3.2 Server Interface**

The server verifies user credentials during sign-up by checking:

* Password: Meets required complexity standards.
* Email: Is valid and verified via a confirmation message.

During sign-in, the server validates the email and password. If either is invalid, an error message, "Invalid email or password," is displayed.

The server processes user actions such as:

* Location Selection: Handles GPS or manual inputs.
* Air Quality Data Retrieval: Fetches pollutant data via external APIs.
* Tree Recommendation: Recommendations for suitable tree species to plant based on the pollution levels in that region
* Save or Share: Updates saved data or facilitates sharing.

It ensures secure and efficient data handling for all interactions.

**3.3.3 Administrator Interface**

When a user interacts with the application, such as submitting data or saving tree recommendations, a notification is sent to the administrator interface. The administrator can review the actions and take the following steps:

* Approve: Confirm the submission or action by clicking the "Approve" button.
* Reject: Dismiss the submission or action by clicking the "Reject" button.

This ensures the quality and accuracy of user interactions and data within the application

**3.4 Development Methodology**

In the field of software engineering and project management, a development methodology is a systematic framework designed to guide the planning, structuring, and control of the information system development process. It encompasses a comprehensive set of guidelines, practices, and procedures aimed at ensuring the efficient and effective production of high-quality software solutions.

Development methodologies serve as indispensable tools for managing complexity, fostering collaboration among team members, and ensuring that projects are delivered on schedule and within the allocated budget. They provide a clear roadmap that aligns the efforts of all stakeholders towards achieving the project objectives.

This section explores the diverse spectrum of development methodologies, shedding light on both traditional and agile approaches that have transformed the software development landscape. By examining the principles, strategies, and best practices associated with these methodologies, readers can gain valuable insights into selecting and applying the most suitable approach for their projects. This understanding is vital for navigating the challenges of modern software development effectively.

**3.5 Tools and Languages**

The following tools and technologies are used in the development of this application:

**Backend .NET:** For handling server-side logic, APIs, and database interactions.

**Flutter**: A cross-platform framework for developing mobile applications.

**AI:** For plant identification and disease detection.

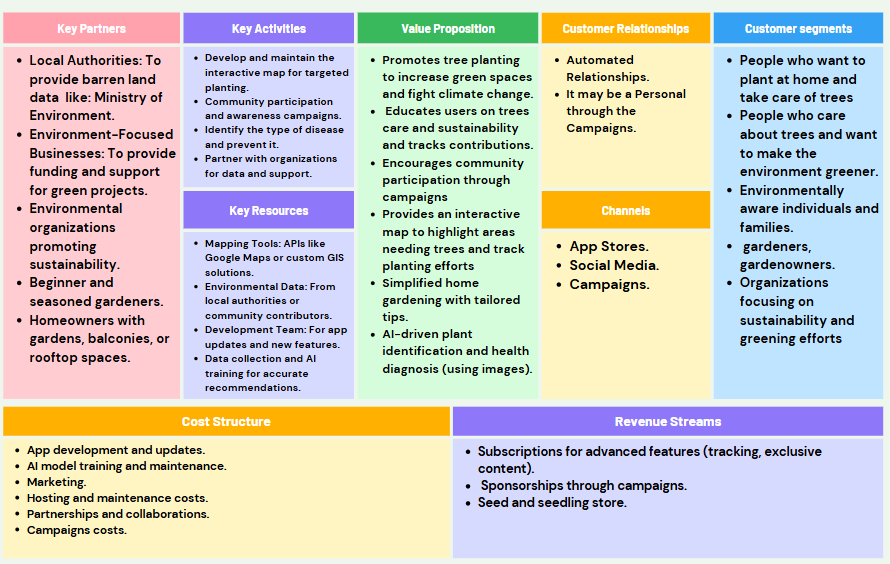
**SQL Server:** For managing and storing structured application data.

**UI/UX:** To design intuitive and user-friendly interfaces.

**Git/GitHub:** For source code management and collaboration.

[**3.**](https://docs.google.com/document/d/1YU1d2ttiDz5MN1gg5ctPL_H4OVnwszw7/edit#heading=h.mrbjoymat5aa)**6 Business Model**

(Figure 3.6) Business Model

****

**3.7 Summary**

This chapter provides a comprehensive overview of the project, starting with the purpose and scope outlined in the introduction, which emphasizes addressing environmental issues through tree planting and awareness. It describes the system’s functional and non-functional requirements, detailing user expectations and system attributes like performance, security, and scalability. The architecture and development methodology sections outline the structure of the system and the iterative approach used. Visual aids such as diagrams (Use Case, Sequence, and Activity) clarify workflows and user interactions. The chapter concludes with an overview of the tools and technologies, including Flutter, .NET, AI and SQL, used to build the system.

|  |
| --- |
| **Chapter 4** |

**Chapter 4: System Design**

The design of the investigation explains how the study was formulated to investigate each question or hypothesis and if appropriate, it identifies all variables and how they are manipulated.

**4.1 Introduction**

After determining the requirements of the system, we will discuss system design in this chapter, which is the process of defining elements of a system like modules, architecture, components, and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing, and designing systems, which satisfy the specific needs, and requirements of a business or organization. Design is the act of taking information, creating a product design to be manufactured, and developing systems to meet specific user requirements. Therefore, there are many types to achieve this purpose and meet all requirements, including:

**Architectural Design:**

focuses on designing a system structure that describes the structure, behavior, perspectives, and analysis of this system.

**Logical design:**

focuses on the abstract representation of system data, input, and output flows. This is often done by modeling from the actual system. In the context of systems, designs are included. Logical design includes relationship entity diagrams (ERD).

**Physical Design:**

focuses on actual system I / O operations. Everything related to how data is entered into the system, how to validate it, how it is processed, and how it is presented is covered. In the physical design, the following requirements of the system are defined:

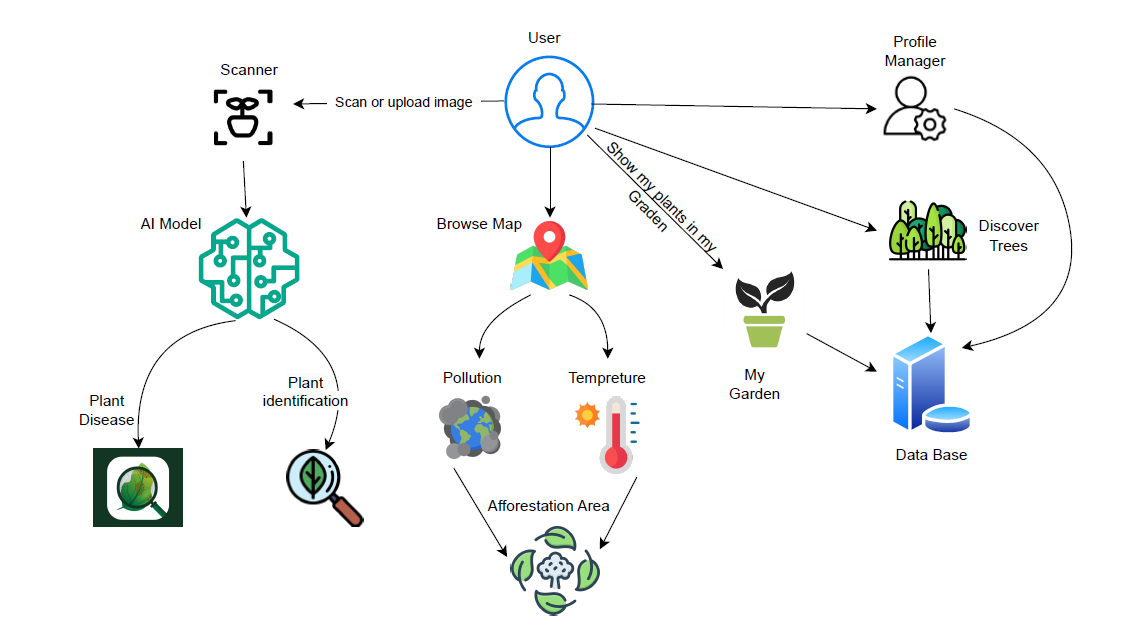
* Input condition.
* Production requirements.
* Storage requirements.
* Processing requirements.
* Control system, backup, or recovery.

In other words, the actual design part of the system can be divided into three subtasks:

* User interface design
* Data design.
* Process design

**4.2 System Architecture**

The system architecture is the conceptual model that defines a system's structure, behavior, and views. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. Systems Architecture responds to the conceptual and practical difficulties of the description and the design of complex systems. Systems Architecture helps to describe and consistently and efficiently design complex systems.

(figure 4.2.1) Shows System Architecture****

**4.3 Development Methodology**

Unified Modeling Language (UML) is a standardized modeling language enabling developers to specify, visualize, construct, and document artifacts of a software system. Thus, UML makes these artifacts scalable, secure, and robust in execution. UML is an important aspect of object-oriented software development. It uses graphic notation to create visual models of software systems.

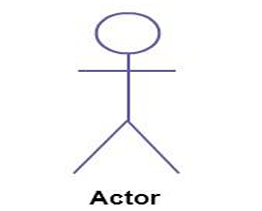
**4.4 Use Case Diagrams**

Describes functionality of a system in terms of actors, goals as use cases and dependencies among the use cases. It is a methodology used in system analysis to identify, clarify, and organize system requirements. A use case diagram contains four main components.

**Actors :**

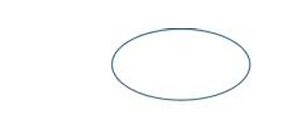
Are usually individuals involved with the system defined according to their roles? The actor can be a human or another external system.

(figure 4.4.1) Shows Actor Symbol



**Use Case :**

It is a visual representation of a distinct business Functionality in a system.

(figure 4.4.2) Shows Use Case Symbol

**System Boundary :**

(figure 4.4.3) Shows the System Boundary Symbol

**Relationships :**

**Generalization**

It is a relationship from a child use case to a parent use case, specifying how a child can specialize all behaviors and characteristics described by the parent.

(figure 4.4.4) Shows Generalization Relationship Symbol



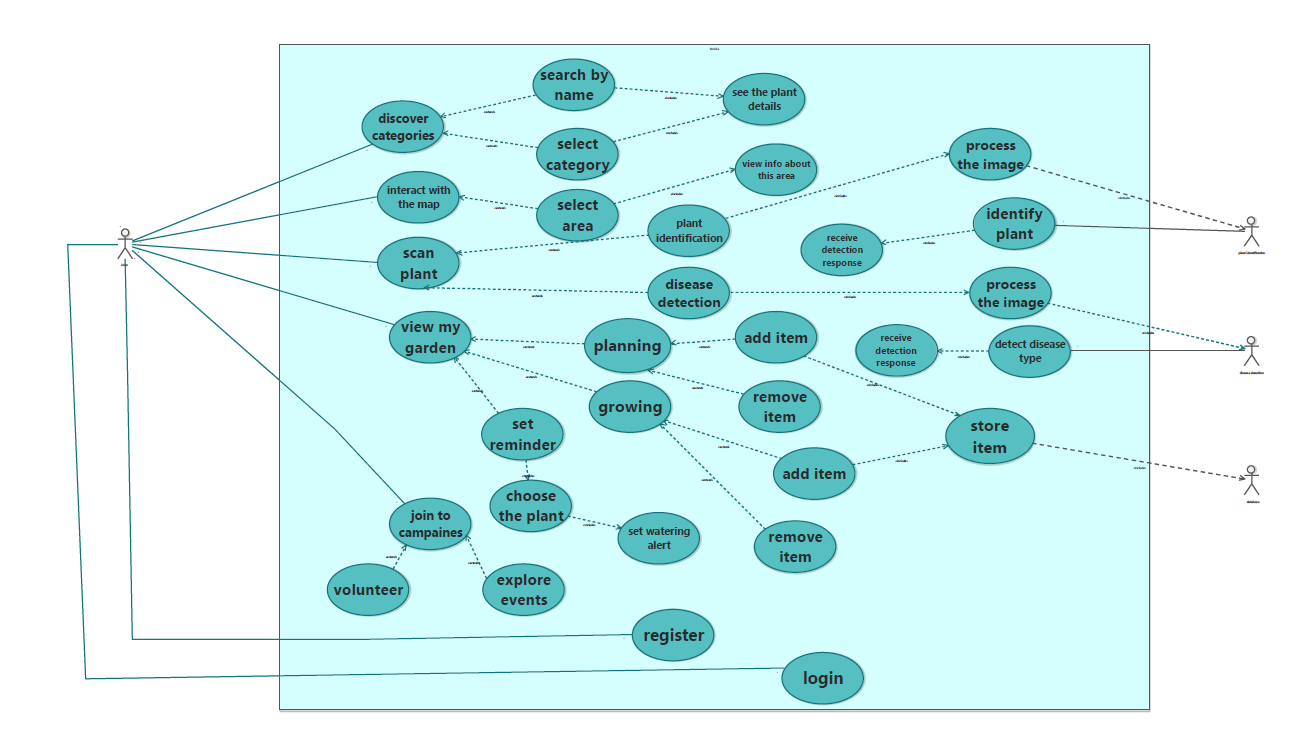
**Association**

It is the relationship between an actor and a business use case. It indicates that an actor can use a certain business system functionality.

(figure 4.4.5) Shows Association Relationship symbol



(figure 4.4.6) Shows Use Case Diagram



1. **Scan/Upload Image for Identification**: Identify plants by scanning or uploading images.

**Extends to:**

* + 1. **See Plant Type**: Display the plant type.
    2. **See Info About Plant**: Show details about the plant.

1. **My Garden**: Manage the user's virtual garden.

**Extends to:**

* + 1. **Set Reminder**: Set reminders for plant care.
    2. **Track growing**
    3. **planning to plant**
    4. **View My Garden**: View and edit the garden.

Extends to:

* + - * 1. **Edit Garden**: Make changes to the garden.

1. **Join Community/Environmental Challenges**: Participate in environmental challenges and community events.

**Extends to:**

* + 1. **Registration for Campaigns**: Register for campaigns.
    2. **Explore Events and Community**: Discover events and community activities.

1. **Discover Trees**: Explore information about different tree species.
2. **Upgrade for Premium**: Upgrade the account to access premium features

**4.5** **Logical and Physical Database Design**

The database design is a crucial component of the system, ensuring efficient data storage, retrieval, and management. This section presents the Entity-Relationship Diagram (ERD), which provides a conceptual representation of the database structure, including entities, attributes, and relationships. Following the ERD, the database schema and physical implementation details are described, outlining the design of tables, primary and foreign keys, and constraints. Together, these elements form the foundation for the system's data operations.

(figure 4.5) Shows database design

A screenshot of a computer

AI-generated content may be incorrect.

**4.6 Sequence diagram**

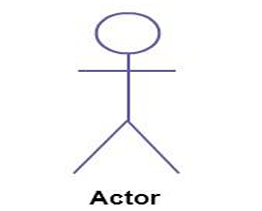
A sequence diagram illustrates how objects interact in a specific sequence over time. It captures the objects and classes involved in a scenario, along with the sequence of messages exchanged between them to execute the functionality of the scenario. These diagrams are also referred to as event diagrams or event scenarios and are valuable in understanding the dynamic behavior of the system.

**4.6.1 Component of sequence diagram**

**Actors:**

An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.

(figure 4.6.1.1) Shows Actor

****

**Lifeline:**

Represents the passage of time as it extends downward. This dashed vertical line shows the sequential events that occur to an object during the charted process. Lifelines may begin with a labeled rectangle shape or an actor symbol.

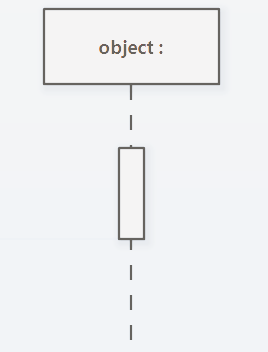
(figure 4.6.1.2) Shows lifeline



**Activation:**

A thin rectangle on a lifeline represents the period during which an element is performing an operation. The top and the bottom of the rectangle are aligned with the initiation and the completion time respectively.

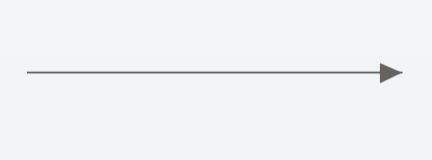
(figure 4.6.1.3) Shows lifeline activation

**

**Call Message:**

A message defines a particular communication between Lifelines of an Interaction. Call message is a kind of message that represents an invocation of operation of the target lifeline.

(figure 4.6.1.4) Shows call message

**

**Reply Message:**

A message defines a particular communication between Lifelines of an Interaction. Return message is a kind of message that represents the pass of information back to the caller of a corresponding former message.

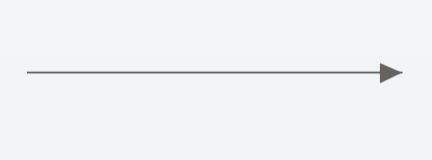
(figure 4.6.1.5) Shows reply message

**

**Asynchronous Message:**

Unlike a call message, an asynchronous message does not require a response.

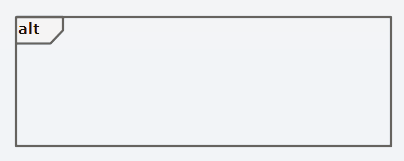
(figure 4.6.1.6) Shows asynchronous message

**

**Alt Fragment:**

An alt fragment in a sequence diagram represents alternative interactions or conditional logic within a system. It models a scenario where a decision must be made, and only one of the alternative paths is executed based on a specified condition.

(figure 4.6.1.7) Shows alt fragment

**

**Image Scanning Sequence Diagram:**

(figure 4.6.1.8) Shows sequence diagram seq1

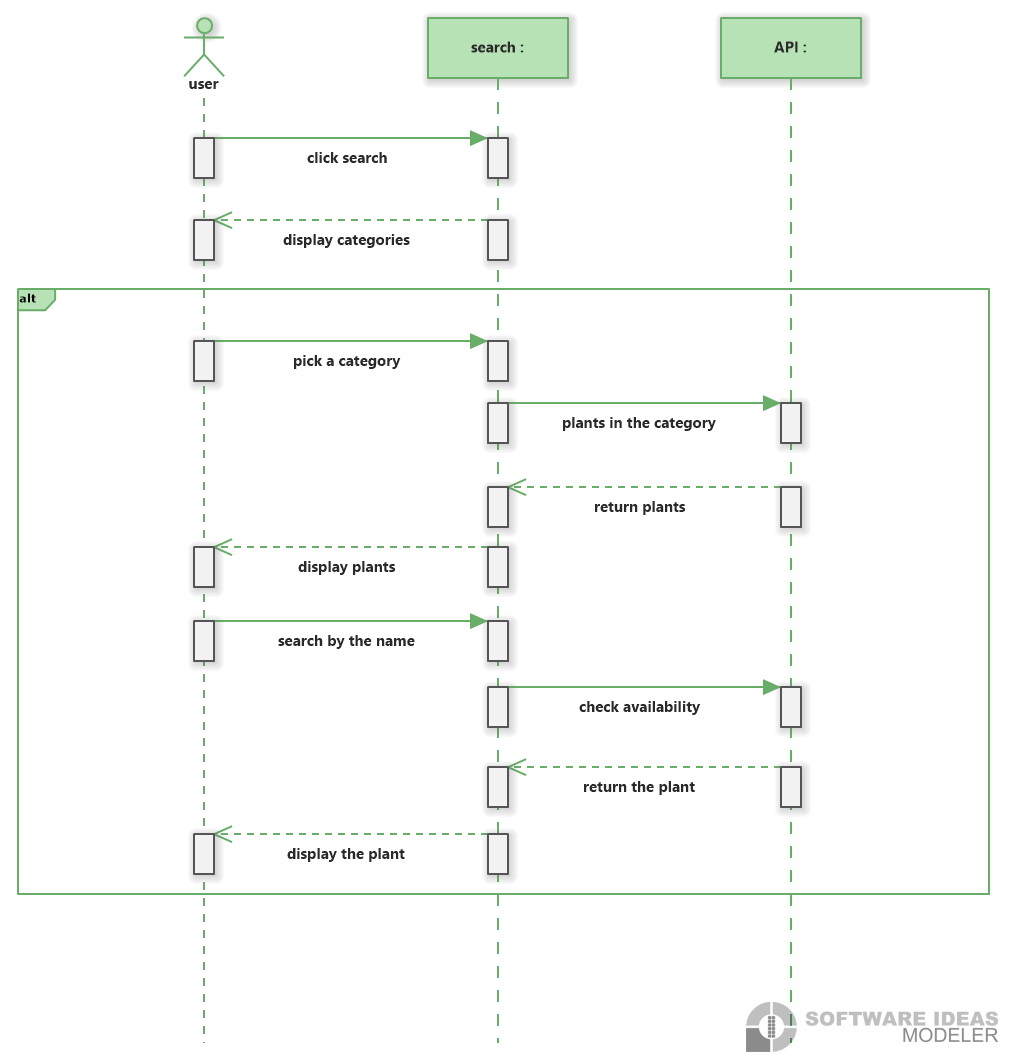
A screenshot of a computer screen

AI-generated content may be incorrect.

This diagram demonstrates the process of tree detection via image upload. Users upload an image, and the system analyzes it using machine learning. The results are displayed, including the plant's type.

**Discover Trees Sequence Diagram:**

(figure 4.6.1.9) Shows sequence diagram seq2



The "Discover Trees" sequence diagram demonstrates how users can explore different tree species within the application. Users can either select a specific category of trees or choose to view all available trees. When a category is selected, the system displays a list of trees within that category, providing detailed information about each tree. If no category is selected, the system shows a comprehensive list of all trees. This functionality allows users to gain insights into various tree species, aiding in their understanding and decision-making regarding tree planting and care.

**View My Garden Sequence Diagram**

(figure 4.6.1.10) Shows sequence diagram seq3

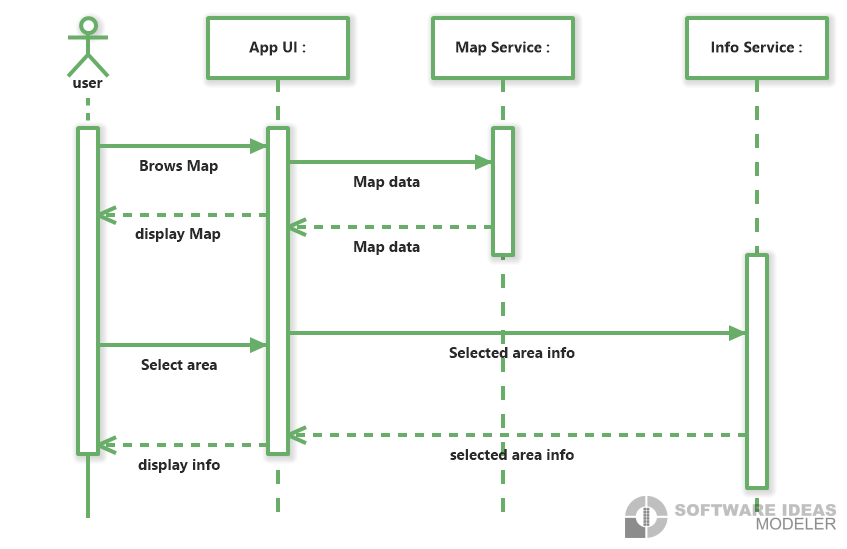
A diagram of a garden

AI-generated content may be incorrect.

The "View My Garden" sequence diagram demonstrates how users can view and manage their personal garden space within the application. The process begins with the user selecting the "My Garden" option, prompting the system to fetch and display the user's garden data from the database. Users can set reminders for gardening tasks, add to growing, add to planning and make edits to their garden layout. The system ensures that all changes are saved and reflected in the displayed garden layout, providing users with an interactive and personalized gardening experience.

**Browse Map Sequence Diagram:**

(figure 4.6.1.11) Shows sequence diagram seq4



The "Browse Map" sequence diagram demonstrates how users can explore different locations on a map and view information about plants in specific areas. The process begins with the user selecting the map browsing option in the app. The app UI requests map data from the map service, which retrieves and returns the data. The app UI then displays the map to the user. When the user selects a specific area on the map, the app UI requests detailed information about that area from the info service. The info service retrieves and returns the information, which is then displayed to the user. This functionality allows users to gain insights into the plants present in various locations, enhancing their understanding and interaction with the environment.

**Campaigns Sequence Diagram:**

(figure 4.6.1.12) Shows sequence diagram seq5

A diagram of a campaign

AI-generated content may be incorrect.

The "Campaigns" sequence diagram demonstrates how users can explore upcoming campaigns. The process begins with the user selecting the option to explore upcoming campaigns. The community system checks the database for any upcoming campaigns and displays the results to the user. If no events are found, a message is displayed indicating the absence of campaigns. Users can also see details about the campaigns. The system retrieves and displays data about campaigns. This functionality enhances user engagement by providing access to campaigns activities and interactions.

**4.7 Activity Diagrams**

An activity diagram illustrates the sequence of actions and decisions involved in a specific process. It is used to represent how the user interacts with the app and how the system responds to each action.

(figure 4.7.1) Shows Activity diagram act1

A diagram of a company

AI-generated content may be incorrect.

A diagram of a garden

AI-generated content may be incorrect.(figure 4.7.2) Shows Activity diagram act2

(figure 4.7.3) Shows Activity diagram act3

A diagram of a diagram

AI-generated content may be incorrect.

**4.8 Interface Design**

A screenshot of a cell phone

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

Screens screenshot of a phone

AI-generated content may be incorrect.

Screens screenshots of a phone

AI-generated content may be incorrect.

Screens screenshot of a mobile app

AI-generated content may be incorrect.

**4.5 Summary**

This chapter has outlined the essential components and methodologies utilized in the system design process. It delves into the system architecture, development methodologies, and diagrams such as use case, class, sequence, and context diagrams, all of which contribute to the structured visualization of the project. The chapter also introduces the graphical user interface (GUI), providing a glimpse into the system's user-centered design. These design elements collectively lay the foundation for a functional and intuitive application.

**Chapter (5)**

**Chapter (5): System implementation**

**5.1 Introduction**

In this chapter we show the implementation of Releaf Project.

**5.2 Backend Implementation**

This part handles the core logic of the system. It receives image inputs from the user, sends them to the appropriate AI model for processing, and returns the results to the user interface.

You can find code here :

<https://github.com/Releaf-Team/Backend>

The samples of code is displayed in the images below:

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a program code

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

**5.3 Plant Identification AI Model**

**This module is responsible for identifying the type of plant from the uploaded image. It uses a trained model to recognize different plant types and returns the most likely match.**

You can find code here :

<https://github.com/Releaf-Team/AI-Models/blob/main/Plant%20Identification%20Detection.ipynb>

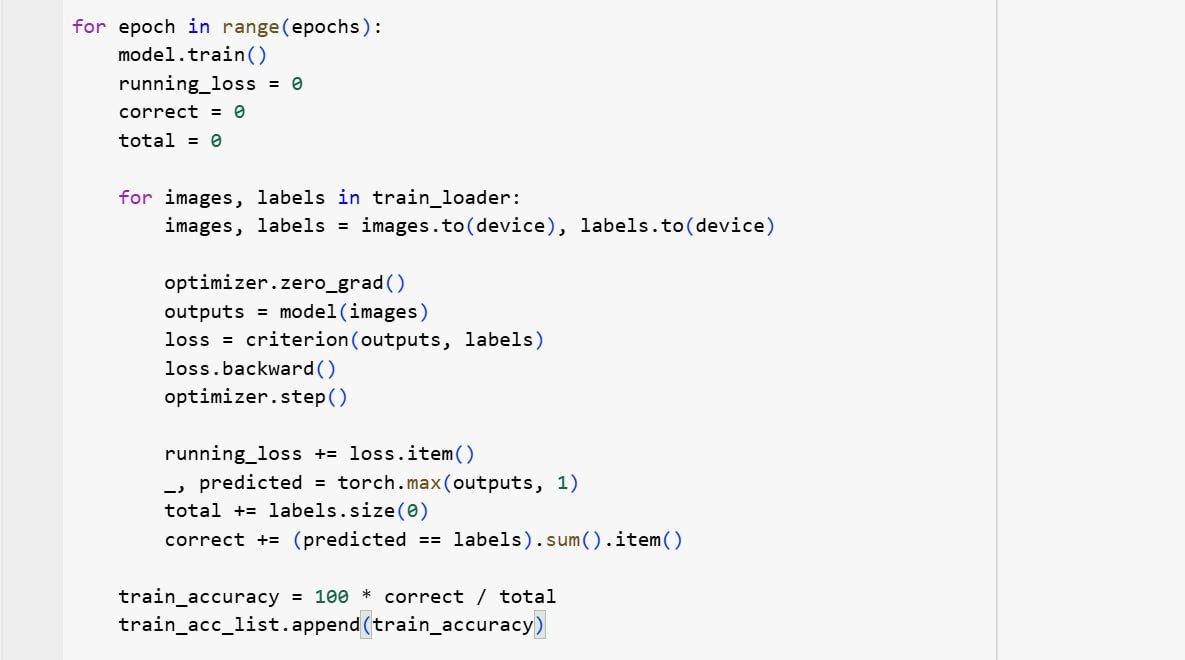
The samples of code is displayed in the images below:

A computer code with many text

AI-generated content may be incorrect.

A computer screen with text

AI-generated content may be incorrect.



**5.4 Plant Disease Diagnosis Model**

This section diagnoses the health condition of the plant. It determines whether the plant is healthy or infected, and if infected, identifies the specific disease.

You can find code here :

<https://github.com/Releaf-Team/AI-Models/blob/main/Plant%20Disease%20Detection.ipynb>

The samples of code is displayed in the images below:

A screen shot of a computer

AI-generated content may be incorrect.

**5.5 Flutter Application Interface**

The user interface was developed using Flutter. It allows users to upload images, view identification and diagnosis results, and interact with the system in a simple and clear way.

You can find code here :

<https://github.com/Releaf-Team/Flutter>

The samples of code is displayed in the images below:

Integrate the Ai model:

A screen shot of a computer program

AI-generated content may be incorrect.

Fetch data from Api:



Give the Ai model prompt:

A screenshot of a computer program

AI-generated content may be incorrect.

Fetch city coordinates for map :



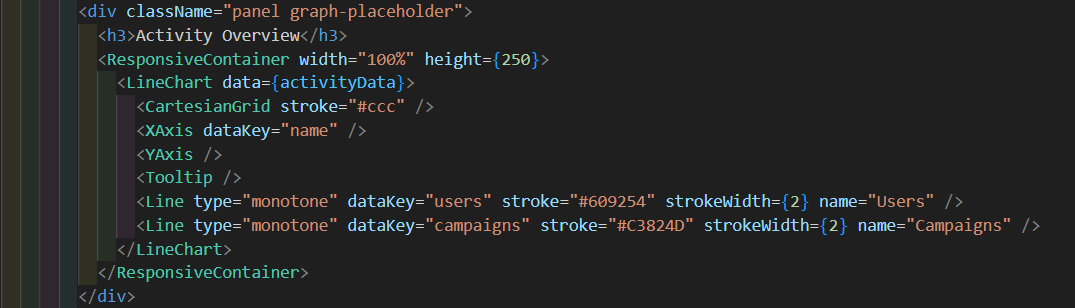
**5.6 Web Interface**

This section presents the web-based user interface, which allows users to interact with the system through a browser. The interface enables image uploading, displays the identification and diagnosis results, and ensures a smooth and user-friendly experience.

You can find code here :

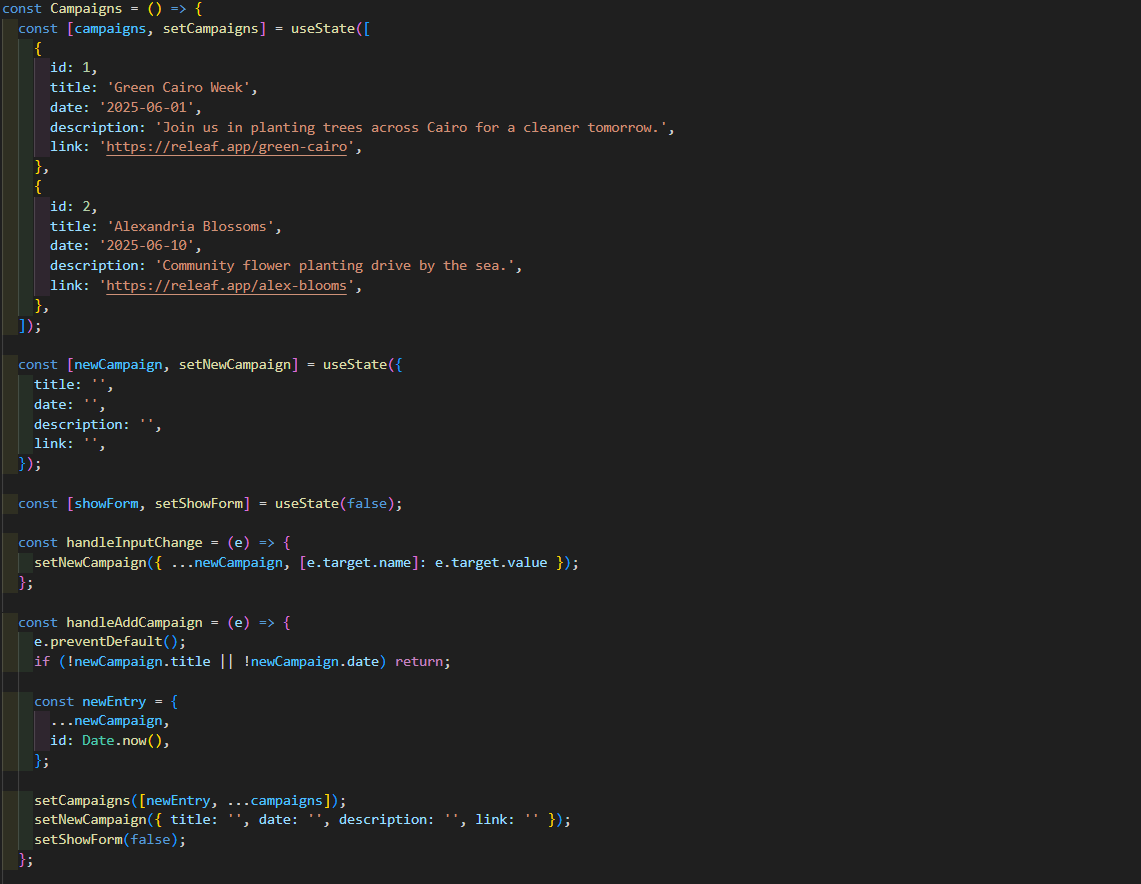
<https://github.com/therawlu/releaf-dashboard>

The samples of code is displayed in the images below:



A screen shot of a computer

AI-generated content may be incorrect.



**Chapter (6)**

**Chapter (6) :** **Conclusion and Future Work**

**6.1 Introduction**

Many users lack the knowledge or tools needed to choose the right plants, identify planting areas, or care for trees effectively. Our mobile application helps solve this problem by using AI and environmental data to suggest suitable planting locations, recommend tree species, and provide care guidance through interactive features and a built-in chatbot.

**6.2 Overall Weaknesses**

* Limited environmental or plant data for certain regions.
* Occasional inaccuracies in plant recognition or recommendations.
* Some user-uploaded images may be rejected due to quality issues.

**6.3 Overall Strengths**

* Provides accurate planting recommendations based on pollution, climate, and location.
* Identifies plants and offers care instructions using AI.
* Simple, intuitive, and accessible mobile interface.
* Encourages community participation through campaigns and personal garden tracking.
* Offers chatbot support for real-time help without requiring expert knowledge.

**6.4 Future Work**

* Expand environmental and plant species datasets for better coverage
* Recommend customized care plans and treatment suggestions
* Enhance image processing and recognition accuracy
* Improve chatbot with voice input and multi-language support
* Add plant health tracking with visual growth timelines and alerts

**References**

* <https://www.kaggle.com>
* [Wikipedia](https://www.wikipedia.org/)
* <https://www.eeaa.gov.eg/>
* <https://openaq.org/>
* <https://earthengine.google.com/>
* <https://waqi.info/>
* <https://www.gbif.org/>
* <https://api-docs.iqair.com/?version=latest>
* <https://www.iqair.com/world-air-quality>
* <https://www.mdpi.com/1424-8220/18/8/2674>
* <https://doi.org/10.1016/j.envint.2019.105190>
* <https://journals.sagepub.com/doi/10.1177/0309133308094081>
* <https://webgi.xyz/editor3>
* <https://api.flutter.dev>
* <https://pub.dev/>
* <https://docs.flutter.dev>
* <https://www.idsc.gov.eg>