PROJECT TITLE

SMART GREEN GARDENING

Background

Urbanization and population growth have led to increased pressure on land resources and ecosystems, particularly in urban areas. As cities expand, green spaces and natural habitats are being replaced by buildings, roads, and infrastructure, resulting in environmental degradation, loss of biodiversity, and reduced quality of life for residents. In response to these challenges, there is a growing interest in promoting sustainable urban development practices, including the integration of green infrastructure into city planning and design. Green infrastructure refers to the network of natural and semi- natural features such as parks, gardens, wetlands, green roofs, and urban forests that provide multiple environmental, social, and economic benefits to urban communities. These green spaces help mitigate the urban heat island effect, improve air and water quality, reduce stormwater runoff, and enhance biodiversity, while also providing recreational opportunities, promoting mental well-being, and supporting local economies.

Nevertheless, there are a few obstacles to overcome before green infrastructure can be widely adopted and implemented in metropolitan areas. The aforementioned factors encompass restricted financial support and assets, conflicting land use preferences, legal impediments, and insufficient cognizance among legislators, developers, and residents on the significance and worth of green infrastructure in metropolitan settings.

Root Cause of the Problem:

The root cause of the problem lies in historical emphasis on traditional urban development practices that prioritize economic growth and infrastructure expansion at the expense of environmental sustainability and human well-being. Rapid urbanization, population growth, and land development have led to the loss of green spaces and natural habitats, resulting in negative impacts on ecosystem health, public health, and quality of life in cities. This problem is

particularly important to consider due to its wide-ranging implications for environmental sustainability, public health, and community resilience. Many people face challenges in maintaining healthy and thriving gardens due to factors such as lack of expertise, limited time, and varying environmental conditions. Traditional gardening methods often rely on manual monitoring and intervention, which can be inefficient and ineffective in ensuring optimal plant growth and maintenance. There is a need for a smart gardening solution that leverages technology to provide personalized guidance, automated monitoring, and intelligent recommendations to support gardeners in achieving successful and sustainable gardening outcomes.

Green infrastructure plays a critical role in addressing pressing urban challenges such as climate change, air and water pollution, heat-related illnesses, and social inequities. By integrating green infrastructure into urban planning and design, cities can create healthier, more resilient, and more livable environments for current and future generations.

Solution to the Problem

Investigating the use of IoT-enabled irrigation systems that monitor soil moisture levels, weather forecasts, and plant water requirements to deliver water precisely where and when it's needed. Exploring technologies such as drip irrigation, soil moisture sensors, and weather- based controllers to reduce water waste and promote water efficiency in urban gardens.

Investigating the role of urban gardening in supporting biodiversity, pollinator habitat, and ecological resilience in urban environments. Discussing strategies for monitoring and enhancing urban ecosystems through green infrastructure, native plantings, and wildlife-friendly gardening practices. Partner with government agencies to provide support services and technical assistance to community gardeners, including soil testing, pest management advice, and horticultural training. Leverage government extension services, agricultural experts, and Master Gardener programs to offer personalized guidance and support to gardeners using the Smart Gardening Assistant platform.

Addressing policy and planning considerations related to smart green gardening initiatives, including zoning regulations, land use policies, and incentives for sustainable landscaping practices. Discussing the role of local governments, urban planners, and community organizations in supporting and promoting green infrastructure development in urban areas.

The target group of users:

The target group includes urban gardeners, community organizations, government agencies, policymakers, and urban planners. They will benefit from access to smart irrigation systems, ecosystem monitoring tools, government support services, and policy considerations to promote sustainable gardening practices and enhance urban ecosystems.

Contribution to Scientific Development: Our project contributes to the understanding of how IoT sensors, AI algorithms, and data analytics can improve plant care, resource management, and environmental sustainability through their integration into gardening methods. We produce scientific results that illustrate the efficacy and effects of smart gardening solutions on plant health, water conservation, and ecosystem resilience through empirical research and data analysis. In order to further the scientific understanding of sustainable gardening methods, our study assesses the environmental effects of smart gardening techniques like carbon sequestration, biodiversity promotion, and urban greening. combining several technologies to create a comprehensive platform for smart gardening assistants. Addressing particular issues including interoperability, scalability, and usability that have been noted in earlier research.
Carrying out empirical studies to assess the impact and efficacy of smart gardening solutions in practical environments. Working together with industry partners to improve scientific understanding in the subject of smart gardening, exchange best practices, and validate methodology. improving accessibility and usability with voice commands, smartphone integration, and user-friendly interfaces.

Utilizing AI-powered analytics to deliver preemptive alerts, real-time insights, and predictive recommendations. Including social components to promote cooperation and knowledge exchange, such as online workshops, community forums, and cooperative gardening projects. Facilitating communication between current IoT gadgets, sensors, and gardening equipment to reduce duplication and increase usefulness. In general, the goal of our suggested solution is to provide gardeners with cutting-edge tools and individualized assistance so they may create healthy gardens, encourage environmental stewardship, and create thriving neighborhoods centered around sustainable living.

Process Model

The software described is a comprehensive solution aimed at enhancing the gardening experience through technological innovation. Here's an analysis of its nature and environment:

Nature of Software:

- **Integrated Solution:** The software serves as an integrated platform that combines features for government oversight, gardener management, smart device integration, and community interaction. It offers a holistic approach to gardening management and support.
- **Smart Technology:** Leveraging IoT devices, sensors, and automation, the software introduces smart capabilities to traditional gardening practices. This includes real-time monitoring of environmental conditions, automated watering systems, and plant health analysis.
- **Education and Awareness:** Access to gardening guides, tutorials, and resources can empower individuals to reconnect with nature, learn about local flora and fauna, and appreciate the importance of sustainable living.

- **Local Economy Support:** By promoting local gardening initiatives and providing funding opportunities, the project can contribute to the development of local economies, supporting small-scale agriculture, and gardening businesses.

Environment of the Software:

- **Digital Environment:** The software operates within a digital ecosystem, accessible via the web and mobile platforms. It facilitates seamless interaction between users, devices, and data through interconnected interfaces and APIs.
- **Collaborative Ecosystem:** The software fosters a collaborative environment where government agencies, gardeners, technology providers, and gardening enthusiasts can interact, share knowledge, and contribute to collective learning and improvement.
- **Real-World Integration:** Despite its digital nature, the software interfaces with the physical environment through IoT devices and sensors deployed in gardens. It bridges the gap between virtual and real-world gardening experiences, enhancing the user's connection to nature.
- **Dynamic Landscape:** The software operates within a dynamic landscape characterized by evolving technological trends, environmental factors, and user needs. It must adapt and evolve over time to remain relevant and effective in supporting sustainable gardening practices.

Environmental Impact:

- **Resource Conservation:** By providing real-time monitoring of environmental conditions and automated watering systems, the project can help conserve water by ensuring efficient use only when necessary.
- **Reduced Chemical Usage:** With features for monitoring plant health and providing recommendations, gardeners may use fewer chemical pesticides and fertilizers, promoting organic and sustainable gardening practices.
- **Biodiversity Promotion:** Encouraging community interaction and sharing gardening experiences can lead to the exchange of ideas about biodiversity conservation, such as planting native species or creating habitats for beneficial insects and wildlife.

In summary, the software represents a sophisticated yet adaptable solution that integrates smart technology with traditional gardening practices. It operates within a collaborative digital ecosystem while interfacing with the physical environment, aiming to enhance the gardening experience and promote sustainability.

Selection Process Model:

After studying several Software Engineering process models, we have decided to use Agile (SCRUM). We believe that this model is appropriate for our project **Smart green gardening**. Among all other process model to develop our proposed software we chose SCRUM process model because:
SCRUM has 3 phases-

1. **Pre-game**

Pre-game phase includes 2 sub-phases-

a. Planning
b. Architecture

2. **Development/ Game phase**

3. **Post-game**

As our proposed system is Smart Green gardening and it goes to category B, we preferred to have a product backlog list where all the currently known requirements are listed as well as the requirements are prioritized, and the effort needed for the implementation is estimated. Our product backlog list will be constantly updated with new and more detailed items, as well as with more accurate estimations and new priority orders.

Last update **in January 2022,** there isn't a specific system with all the functionalities that exist as a single, integrated solution worldwide. However, various components and features, such as smart gardening apps, IoT devices for monitoring environmental conditions, and community

platforms for gardeners, do exist individually or in partial combinations in different contexts and regions.

The development and implementation of such a comprehensive system would likely require collaboration between government bodies, technology companies, and gardening enthusiasts. While there are similar initiatives and technologies in existence, a fully integrated system combining all the functionalities may not be widespread or standardized globally.

However, advancements in technology and increasing interest in sustainable practices may lead to the emergence of more comprehensive gardening solutions in the future. If you're interested in creating such a system, you may need to research existing technologies and platforms, identify potential partners or collaborators, and develop a roadmap for implementation.

Next, the development/game phase is divided into parts. Each division is called sprint. In each sprint traditional phases of software development like requirements, analysis, design, evolution, and delivery are used.
Lastly, the post-game phase entered when an agreement has been made such as the requirements are completed. This phase does the work of integration, system testing and documentation. After this our project is going to be ready for release.

During the processing SCRUM has several meetings which will help our team to complete project deliverables quickly and efficiently. As our **Smart Green Gardening** is quite a long project and we need to change the requirements at any time, SRUM is best. SCRUM divided it into easily manageable sprints. Here developments are coded and tasted during sprint review. For, fast moving development project SCRUM works well. Moreover, SCRUM ensures effective use of time and money.

Project Role Identification and Responsibilities:

- **SCRUM Master:** SCRUM Master interacts with the project team as well as with the customer and the management during the project. He is responsible for ensuring that the project is carried out according to the practices, values, and rules of SCRUM and that it progresses as planned. Breaks down projects into manageable sprints.

- **Product Owner:** The Product Owner is officially responsible for the project, managing, controlling, and making visible the Product Backlog list.

- **SCRUM Team:** SCRUM Team is the project team that has the authority to decide on the necessary actions and to organize itself to achieve the goals of each Sprint. The SCRUM team is involved in effort estimation, creating the Sprint Backlog, reviewing the product Backlog list, and suggesting impediments that need to be removed from the project.
- **Customer:** Customer participates in the tasks related to product Backlog items for the system being developed or enhanced.
- **Management:** Management oversees final decision making, along with the agreements, standards, and conventions to be followed in the project.

SCRUM, Waterfall, and DSDM (Dynamic Systems Development Method) are project management methodologies used in software development. Here are the main differences between these methodologies:

- **SCRUM:** It is a flexible and iterative approach to project management. In this methodology, the development team works in short sprints, typically one to four weeks long, to deliver a working product iteratively. The focus is on delivering value to the customer and adapting to changing requirements as the project progresses. The team is self-organizing, and there is a high degree of collaboration between team members, stakeholders, and the customer.

    - ✓ Adapts quickly to changing requirements.
    - ✓ Encourages collaboration and communication.
    - ✓ Empowers teams to take ownership.
    - ✓ Provides frequent feedback loops.
    - ✓ Reduces time to market.
    - ✓ Mitigates project risks.
    - ✓ Focuses on delivering high-quality products iteratively.

- **V-Model:** The V-Model, also known as the Verification and Validation Model, is a software development methodology that emphasizes the relationship between each stage of the development process and its corresponding testing phase. This model is represented as a V-shaped diagram, where the left side represents the development phases such as requirements analysis, system design, and implementation, while the right side represents the testing phases such as unit testing, integration testing, and acceptance testing. Each stage on the left side is directly associated with a corresponding testing phase on the right side, ensuring that the software is thoroughly verified and validated at every step of the development lifecycle. The V-Model promotes early testing and emphasizes the importance of clearly defined requirements and specifications to minimize errors and defects. However, it can be perceived as a rigid and sequential approach, as each stage must be completed before proceeding to the next, which may not be suitable for projects with evolving requirements or where flexibility is needed.

- **Waterfall (Plan-Driven):** Waterfall is a linear, sequential approach to project management. In this model, the project is divided into distinct phases, such as planning, design, development, testing, and deployment. Each phase must be completed before moving on to the next one. The focus is on completing each phase before moving on to the next, and the project's requirements and objectives are defined at the beginning of the project. We cannot go back to the previous phase to change the requirements. This methodology is less flexible than SCRUM and can result in longer development times and less customer involvement. So, we cannot use this model for our project as in our project requirements are continuously changing and customer interaction is needed.

- **DSDM (Dynamic Systems Development Method):** DSDM is an Agile process that focuses on the development of systems that are timeboxed and have a fixed cost. The model is based on iterative development, and the focus is on delivering a working product while ensuring that the system's quality is maintained. The team is self-

organizing and there is a high degree of collaboration between team members, stakeholders, and the customer.

- **XP** (Extreme Programming):
  Extreme Programming (XP) is an agile software development methodology renowned for its emphasis on flexibility, collaboration, and delivering high-quality software. In XP, teams adhere to a set of principles and practices, including pair programming, test-driven development, continuous integration, and on-site customer involvement. These practices facilitate iterative development cycles where software is incrementally built, tested, and refined based on feedback.

Why we don't select XP, DSDM, WATERFALL, V-Model

XP practices, especially for teams unfamiliar with agile methodologies. This cultural shift requires time and effort to instill new habits and ways of working within the team. Additionally, XP can be resource-intensive, particularly due to practices like pair programming and continuous integration, which may demand additional time and personnel. Moreover, XP relies heavily on consistent customer availability and involvement throughout the development process, which can be challenging to maintain in practice, especially for large or geographically dispersed teams.
Plan-Driven and Waterfall models, with their sequential phases and fixed requirements, lack the flexibility needed for a project characterized by evolving needs and continuous customer interaction. These models may lead to lengthy development cycles, making it challenging to accommodate changes and deliver value incrementally, as required in the project. Moreover, the inability to revisit previous phases to modify requirements in Plan- Driven and Waterfall models contradicts the dynamic nature of the project.

Plan-Driven and Waterfall models, with their sequential and inflexible nature, are fundamentally at odds with the project's dynamic requirements and continuous adaptation. Therefore, considering the project's need for flexibility, customer interaction, and iterative development, an agile methodology like SCRUM emerges as the most appropriate choice.

One limitation of the V-Model is its inherent inflexibility, as it follows a sequential and linear approach to development and testing. This rigidity can make it challenging to accommodate changes or feedback during the development process, leading to potential delays and increased costs if revisions are required. Additionally, the model's emphasis on completing each phase before moving to the next can result in a lengthy development cycle, making it less suitable for projects with tight deadlines or evolving requirements. Furthermore, the V-Model may not adequately address the complexity of modern software projects, particularly those involving iterative or agile development methodologies, where flexibility and adaptability are paramount.

In summary, SCRUM is a flexible, iterative approach to project management, Waterfall is a linear, sequential approach, and DSDM focuses on timeboxing and fixed costs. Comparing with all these process models, we have selected SCRUM model for our project.

Functionalities:

1) **Government Features:**

- **Login:** Government can log in to the system securely to access administrative functions.
- **Logout:** Government can log out of the system to secure access.
- **Confirm Registration**: Government can verify and confirm registrations of new gardeners.
- **Monitor Gardening Activities**: Government can monitor gardeners' activities through the app or web dashboard, including tasks completed, resource usage, and garden maintenance.
- **Track Garden Locations:** Government can track the locations of registered gardens using GPS to ensure proper management and support.
- **Funding Opportunities:**
- **Policy Updates:**

2) **Gardener Features:**

- **Register:**
- **Login:**
- **Logout:**
- **Scan Smart Devices:** can use smart devices to scan plant information, monitor soil conditions, receive gardening tips and recommendations.
- **Emergency Support:** Gardeners can request emergency support or assistance for plant diseases, pest infestations, or extreme weather events.
- **Access Gardening Guides:** Gardeners can access gardening guides, tutorials, and resources for plant care, garden maintenance, and sustainable practices.
- Community Interaction: Gardeners can interact with other users, share gardening experiences, and seek advice or recommendations from the gardening community.

Smart robotic water fountain

- **Scan Plant Information:** Devices can scan plant tags or labels to provide details on plant species, care instructions, and growth requirements.
- **Monitor Environmental Conditions:** Devices can monitor soil moisture, temperature, humidity, and light levels to provide real-time feedback to gardeners.

- **Automated Water Dispensing:** The fountain can automatically dispense water based on preset schedules or triggered by motion sensors, ensuring a continuous supply of water for plants.
- **Automated Tasks:** can automate tasks such as watering, fertilizing, or adjusting lighting based on preset schedules or environmental data.
- **Alerts and Notifications:** Devices can send alerts or notifications to gardeners regarding plant health issues, maintenance tasks, or system updates.

3) Smart Gardening App Features:

- **Save User Information:** The app can save user profiles, preferences, and gardening history for personalized experiences.
- **Provide Gardening Guides:** The app can provide gardening guides, tutorials, and recommendations for beginners or experienced gardeners.
- **Support and Help Center:** The app can include a support center with FAQs, troubleshooting guides, and contact information for customer support.
- **Hire labor**: Gardeners can request emergency support or assistance through the app and receive prompt assistance from gardening experts or support staff.

Functional requirements

Login:

- Users can login to the system with their given username and password.
- For login to the system database records will be compared with the username and password.
- If the login is successful, the homepage will be shown.
- The system will randomly generate a verification code and send it to the user's email address to try again if the entered username and password are incorrect.
- If a user attempts to login more than three times, the system will display "Forgot Password?"
- Anyone who selects the "Forgot Password" option will see a page where they must enter their mailing address. The user's mailbox will receive a verification code.
- The user will be able to change the password once they have entered the verification code. The user will then be automatically logged in and the home page will appear.

Priority Level: High

Pre-condition: Registration

Confirm Registration:

- Users must register to log in to the system.
- In this registration process the user must provide country code, mobile number & email address.
- In this form the user needs to provide his/her name.
- Users should set a password for further login.
- Also, they have to provide date of birth, gender, occupation description, NID number as well as picture of NID, picture of birth certificate (for under 18 years old).

- After giving all the information, the user needs to click the submit button.
    - User will submit the code which has been sent via email or phone number.
    - The government will verify all the information.
- User must wait for the confirmation of the Government.
- After getting confirmation from the Government, the user will be successfully registered to the system.

Priority Level: High

Pre-condition: None

Monitor Gardening Activity

- The system shall provide the government with tools to monitor gardeners' activities.
- Have a government Dashboard included with search option.
- After putting on the gardener's Unique id in search box and click on search button government can see all information in a grid box which stored in database.

- Government shall be able to view data on complete tasks, resource usage, and garden maintenance through an app or web dashboard.

- In task menu, there a percentage value will be shown according to gardeners completed task.
- What kinds of resources are used in gardeners gardening activities, these will be shown by a list (bullet point) after clicking on completed task button.
- There will be a pie-chart appearing (what type of crops production is high in all gardeners). pie-chart will be show by clicking the analysis production button on the government dashboard.

Priority Level: High

Pre-condition: None

Track Garden Locations:

- Government users shall be able to track the locations of registered gardens using GPS technology.
  - Tracking buttons also will be included in the government dashboard.
  - After clicking on track button, a map will appear. The map will relate to GPS.
  - There will be a search box. After putting on the gardener's authentication id and clicking on the search menu a directional path will display in the map and show all information about gardeners' address in a grid box.

Priority Level:

High Pre-condition: None

Funding Opportunities:

The system shall provide information on funding opportunities available for gardening projects.

- Gardener's funding opportunities connected with gardeners completed task percentage in database by foreign key.
- After click fund Button two sub button are displayed (utility bill, Water bill)
- By clicking both bullet points a funding percentage box will appear.
- After giving the percentage of funds and clicking confirm button it will be added by gardener's profile. This fund was added by gardeners' utility bill number and water bill number.

Application procedures and deadlines for funding opportunities shall be clearly outlined.

- Application process related to all information and deadlines for funding opportunities edited or generated by government admin only. This information will be private by government admin. This information connected with gardener's profile.

Priority Level: High

Pre-condition: Log in the Government Dashboard.

Policy generate and Updates:

- The system should deliver timely updates on policies relevant to gardening and related activities.
  - Policy related all information edited or generated by government admin only. This information will be private by government admin.
  - This information connected with gardener's profile.
  - The government click on Policy update button then an information storage box display. After writing policy click on confirm button policy will be stored in database and connected with gardeners' profile.
  - There can be updated information based on gardeners' production reports.so there will also be an update button.

Priority Level: High

Pre-condition: Log in Government profile

Scan Plant Information:

- The device should be equipped with a scanning mechanism capable of reading plant tags or labels.
  - The software must enable the device by pressing on button to capture images of plant tags or labels using the built-in camera or scanning hardware.
  - Provide options for automatic or manual focus to ensure clear image capture.
  - Include settings for adjusting exposure, white balance, and other camera parameters to optimize image quality.
  - Develop algorithms to process captured images and extract relevant information from plant tags or labels.
  - Implement algorithms to detect and isolate tags or labels based on predefined characteristics such as shape, color, or text patterns.
  - Include options for reviewing captured images, zooming in/out, and deleting unwanted images.
  - After scanning it will be searched in database plant information according to

Plant labels or shape and pictures.
Matching information from databases (species name, care growth requirements) will be displayed by display of scanner.

Priority Level: Medium
Pre-condition: users need Login and need Database storage about plants information

Monitor Environmental Conditions:

- The device should have sensors to monitor soil moisture, temperature, humidity, and light levels.
  - The device should be equipped with an IOT sensor capable of accurately measuring the moisture content of the soil. The device transmits laser rays to detect.
  - The IOT sensor emits short pulses of laser light and measures the time it takes for light to reflect off objects in the environment.
  - It should provide options for calibrating the laser alignment and focus on ensuring accurate detection.
  - Implement data fusion algorithms to integrate laser scanning data with environmental sensor data effectively.
  - It should continuously collect data on environmental conditions and provide real-time feedback to gardeners in the display of the device.
- Gardeners should be able to view environmental data (like temperature in Celsius , environmental status like cloudy, shiny, rainy) through a user-friendly interface, such as a mobile app or web dashboard.
- Will show weather forecast precipitation in the display when clicking on environmental details.

Priority Level: **High**

Pre-condition: Users must have an authenticated account and need to log in and device capable for sensor capabilities.

Automated Water Dispensing:

The fountain should have the capability to automatically dispense water to plants.

- Install a drip irrigation system with tubing running from the water source to each plant location.
- Drip emitters or drippers are placed near the base of each plant, delivering water directly to the root zone at a slow and controlled rate. It can be controlled by mobile apps. Have an option for a control rate in percentage.

- Provides precise watering to individual plants, conserves water by minimizing evaporation and runoff and can be automated with a timer or controller.
- Attach a nozzle with adjustable flow rates to the water source, such as a faucet or hose outlet.
- Use a pump system connected to a watering hose to deliver water from a reservoir or water source to the plants.
- It should be programmable to follow preset watering schedules tailored to the specific needs of different plant species.
- Integrate multiple dispensing mechanisms for different areas or types of plants within the water fountain system.

Priority Level: **High**

Pre-condition: Users must have an authenticated account and need actuators movement ability.

Automated Tasks:

- The device should support automation of various gardening tasks, including watering, fertilizing, and adjusting lighting.
  - Integrate a fertilizer dispenser or dosing system into the gardening setup. This could include pumps, valves, or mechanisms to release fertilizer.
  - Include containers or reservoirs for liquid or granular fertilizers, which can be connected to the dispensing system.
  - Develop algorithms to calculate the optimal dosage of fertilizer based on factors such as plant type, growth stage, and nutrient requirements.
  - Install LED grow lights or lighting systems capable of adjusting intensity and spectrum. These lights may include timers, dimmers, or color controls.
  - Implement motorized mechanisms or systems to adjust the positioning of grow lights for optimal coverage.
  - Implement protocols or APIs for data exchange and command execution between different devices and modules.
  - Develop a user-friendly interface, such as a mobile app or web dashboard, for centralized control and monitoring of gardening tasks.

Priority Level: **High**

Pre-condition: Users must have an authenticated account and need knowledge ability like rationally thinking.

Alerts and Notifications:

- The device should be capable of sending alerts or notifications to gardeners regarding plant health issues, maintenance tasks, or system updates.
- Connect the sensors to a microcontroller or IoT device capable of processing sensor data and triggering alerts based on predefined thresholds.
- Develop firmware or software for the microcontroller or IoT device to handle sensor data processing and alert generation.
- Implement algorithms to continuously monitor sensor readings and compare them against the predefined thresholds.

- Alerts can be triggered based on predefined thresholds or anomalies detected in environmental data.
- Notifications should be delivered in real time through multiple channels, such as mobile push notifications or email.

priority Level: High

Pre-condition: Users must have an authenticated account and time identification ability.

Emergency Support:

- Gardeners shall have the ability to request emergency support or assistance for issues such as plant diseases, pest infestations, or extreme weather events.
- Provide a form or interface where users can report emergencies. Include fields for:
  - ✓ Uploading images of the issue
  - ✓ Providing a description of the problem
  - ✓ Specifying the urgency level (e.g., low, medium, high)
  - ✓ Selecting the type of emergency (e.g., plant disease, pest infestation, extreme weather)
  - ✓ Location (either automatically detected through GPS or manually entered)
  - ✓ Contact information (name, email, phone number)
- When users submit an emergency support request, the app can automatically detect their location using GPS functionality built into their device.

- This manual input can be done through a text field where users can enter their address, city, or postal code and can send text according to support.
- incorporates a feedback form or interface within the gardening emergency support app or platform, ensuring it's easily accessible and intuitive for users, consider the following steps:
- Feedback Form Design: Design a user-friendly feedback form with fields for relevant information such as:
    - User's name (optional)
    - Date and time of assistance received
    - Type of emergency or issue addressed
    - Rating scale or selection options for evaluating the effectiveness of the assistance (e.g., star ratings, Likert scale)
    - Text box or comment field for users to provide detailed feedback, share their experiences, and offer suggestions for improvement.

Priority Level: High

Pre-condition: Must be

logged in <u>Community</u>

<u>Interaction:</u>

- **All gardeners of the system can form a Gardening group where they can share their all-garden-related information.**

Provide a user-friendly interface for creating Gardening groups. This interface should include fields for the group name, description, and privacy settings. Allow users to upload a group cover photo to personalize their group page.

Offer multiple privacy settings for Gardening groups:

- ✓ Public: Anyone can join and view the group's content
- ✓ Private: Users must request to join, and group content is visible only to members.
- ✓
- Create chat rooms within each Gardening group where members can engage in real-time conversations.
- Allow users to join different chat rooms based on their interests or specific gardening topics.
- Enable members to send direct messages to each other within the Gardening group.
- Allow users to customize their preferences for receiving notifications about new messages, mentions, or replies.
- Incorporate emojis and reaction buttons into the chat interface to enable members to express their emotions and reactions to messages.
- There will be option for audio and video call.

Save User Information:

- Allow users to input their gardening history, including their experience level, previous gardening projects, and any relevant achievements or awards.
- Provide a dropdown menu or selection options for users to indicate their experience level in gardening. Options may include beginner, intermediate, advanced, or expert.
- Alternatively, allow users to input the number of years they've been gardening to gauge their experience level.
- Include a section where users can list and describe their previous gardening projects inby clicking save experience button.
- Allow users to list any relevant achievements or awards they have received for their gardening efforts. List will bullet form.
- Enable users to upload photos of their previous gardening projects to accompany their descriptions. There will be a file section where they can select from their phone photo or by click in live.
- Users shall have the ability to update and manage their profiles within the app for personalized experiences.

Priority Level: High

Pre-condition: Must have a registered account

Help Center:

- The app shall feature a support center containing frequently asked questions (FAQs).
  - ✓ Create a dedicated section within the app where users can access frequently asked questions related to gardening, app usage, and common issues.
- Users shall be able to access self-help resources to address common issues or inquiries related to the app's functionality.
- Add a button labeled "Tips " or "Helpful Tips" to the app's navigation menu or toolbar.
- This button should be easily accessible to users and prominently displayed on the screen.

Priority Level: medium

Pre-condition: Must be logged in

Hire Labor:

- The app should feature a prominent "Emergency labor" button on the main interface.
- Place the button prominently within the app's interface, ensuring it's easily accessible during urgent situations.
- The gardener should be able to specify the type of assistance needed, such as grass cutting or general garden care.
- Include predefined categories for common tasks like grass cutting, weeding, pruning, etc.
- The gardener should have the option to provide additional comments or details about the specific assistance required.
- Include a text field or comment box in the emergency support request form where the gardener can add any relevant information or instructions.
- The gardener should be able to cancel the emergency labor request if necessary.
- The app should provide a dedicated feature for users to submit requests for labor assistance, such as gardening experts or support staff.
- The app should enable the gardener to share their current location with the support team.
- Users shall receive prompt responses and assistance from qualified personnel to address their gardening needs effectively.

Priority Level: High

Pre-condition: Must be logged in account

Non-functional requirements

- **Security:** The system maintains the privacy of user information.
- **Ease of use:** The system is very user friendly.
- **Reliability:** Our system will run without a failure for a given period under predefined conditions.
- **Availability:** Our system is accessible for all types of user.
- **Performance:** The system returns the results quickly.