Web-Based ********* Care App

Software ************ Specification

******* 1.0

Group ID: F24PROJECT1A165 (BC200405673)

********** Name: ****** Akmal
(haseedakmal@vu.edu.pk)

Revision History

Date (dd/mm/yyyy)	Version	Description	Author
29/11/2024	1.0	In this document, we ******* different document-oriented ****** to understand how website developed, functional and non-functional requirements, usage scenarios, adopted methodology, **** plan.	**BC200405673**

Table of Contents

SRS Document

Scope of Project:

The *********** Care Application** is a user-friendly, web-based platform tailored to assist domestic (home-owner) gardeners in managing and nurturing ***** **** gardens. Designed ** cater ** gardeners of all experience levels, *** application ******** a wealth of features, including personalized plant care recommendations based on specific needs, plant types, and environmental factors. Users *** ****** detailed plant **** schedules, ******** reminders, and pest control advice to ensure healthy growth and thriving plants.

** addition to its ********* gardening tools, the application ******** *********** features such as plant identification, gardening tips, and a virtual ****** planner. It also offers a robust community platform that allows ***** to connect, share ********* experiences, exchange advice, and participate in gardening challenges.

The application ********** advanced technology, **** ** AI-powered ***** health *********** *** weather forecasting tools, to enhance user experience *** ******** the gardening process. ******* its intuitive design and ******** content, the **Gardening **** Application** aims ** inspire individuals to embrace gardening as a ********* hobby, promote sustainability, and foster a ****** connection with nature.

******* & Technologies:**
- ********** Development:**
  - HTML, CSS, JavaScript
  - React ** Angular (*** a modern, component-based framework)
  - Material UI or Bootstrap (for responsive design and UI components)
- **Backend Development:**
  - Node.** or ****** (for server-side logic and API development)
  - Express.** ** Django (for web frameworks)
  - MongoDB ** ********** (for ******** management)
- **Cloud Platform:**
  - AWS, GCP, or Azure (for hosting the *********** and ******* resources)
- **Additional Tools:**
  - Git (for ******* control)
  - NPM or Yarn (for package management)
  - Webpack or ****** (for bundling *** optimization)
  - Testing ********** (e.g., Jest, Mocha)

Functional and Non-Functional Requirements:

Functional Requirements:

- **User Registration *** ******* Management:**
  - ***** users to create accounts, update ******** information, and manage their gardening preferences. *** users should include Gardener, Supervisor, Home ***** and System Admin.
- ******* Database:**
  - Maintain a ************ database of plants, including their types, characteristics, care requirements, age, ****** stages etc. *** plants are mainly of three types; ****** plants, vegetable ****** and fruit plants.
  - Enable users ** search for ****** by name, category, or specific attributes.
- **Personalized Plant **** Recommendations:**
  - Utilize user-provided **** (location, climate, soil type) to offer tailored ***** care advice.
  - Suggest appropriate ******** schedules, fertilization plans, and pest control measures.
- ******* Tracking and Monitoring:**
  - Provide features to track plant growth, record observations, and monitor health indicators.
  - Allow users to set ********* *** ***** **** watering, repotting, or pruning.
- **Interactive Tools and Resources:**
  - Offer interactive tools such as plant identification guides, garden planners, and ******* diagnosis assistance.
  - Provide access ** ********** resources, articles, and tutorials ** various gardening topics.
- *********** Features:**
  - Facilitate a community forum / ***** at Facebook and/or WhatsApp *** users to ***** experiences, *** questions, and connect with other gardeners, home ****** etc.
  - Enable users to create and **** gardening groups based on ********* ** locations. There should be location sharing service for ****** ***** platforms.
- **Alerts:**
  - Send ****** notifications ***** all users for ********* tasks, ******* updates, or plant-related alerts.
- ************* with External Services:**
  - Consider *********** **** weather APIs to provide localized ******* forecasts and gardening tips.
  - ******* partnerships **** gardening supply stores or ********* for product recommendations and discounts.

Non-Functional Requirements:

**** ******* defines requirements that *** not easily defined in *** *** case model. Requirements such as ***** standards, quality aspects, supportability and ********* ******** of the system. The scope of the supplementary requirements ** limited to all the non-functional requirements.

- **Usability:**
  Usability is the degree of **** with which the user will interact with **** products to achieve ******** goals *********** and efficiently.
- **Reliability:**
  Such a metric shows the possibility of your solution to fail. To achieve **** reliability, your **** ****** eliminate all **** **** may influence *** code ****** *** ****** with system components.
- **Performance:**
  *********** describes how your solution behaves when ***** ******** with it in various scenarios. Poor performance *** lead to negative user experience *** jeopardize system safety.
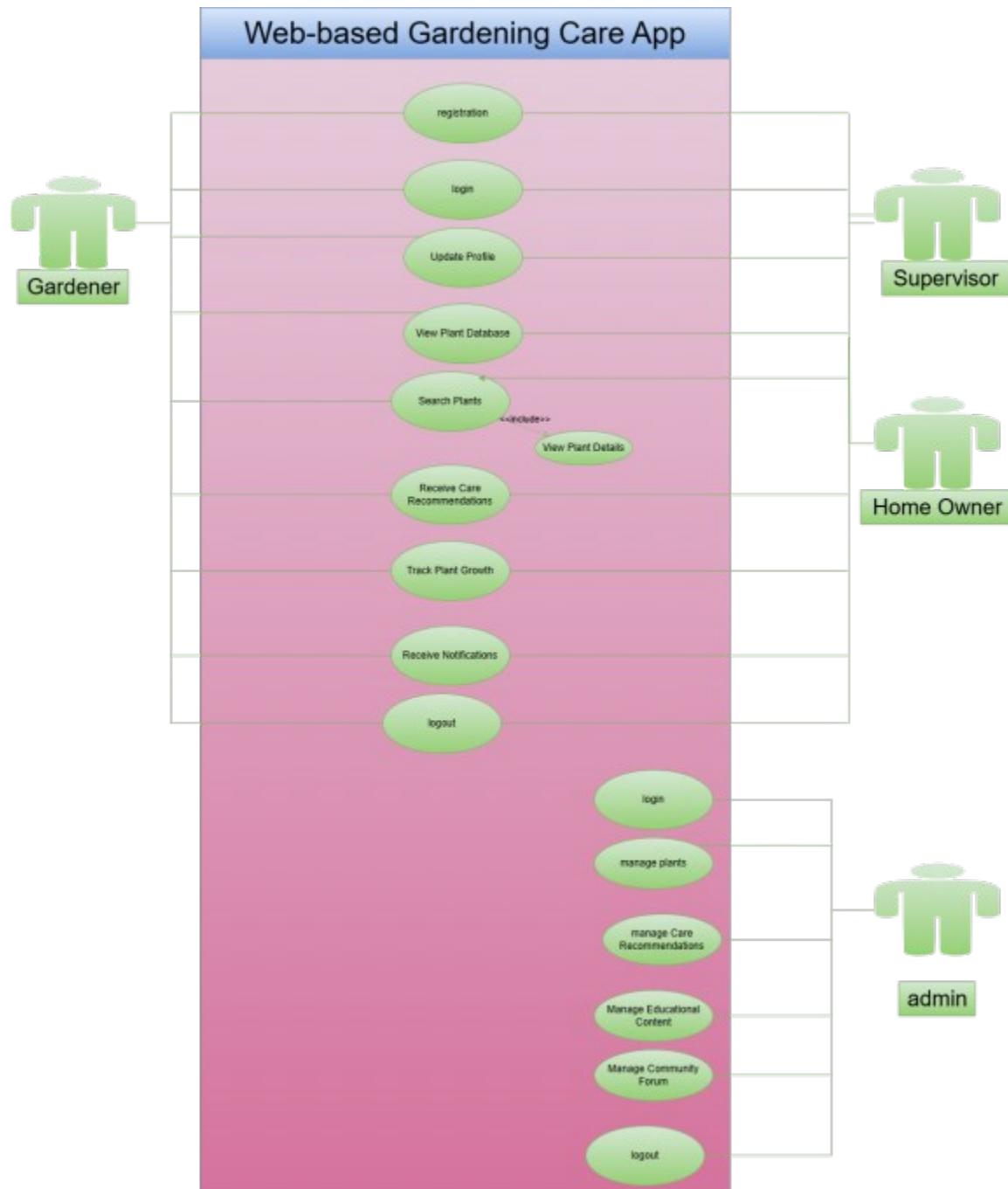- **Supportability:**
  System *********** ****** ** *********** ** current equipment **** as ********* monitors, printers, smart-****** etc.
- **Implementation:**
  The system implementation will be performed all day rather than in phases.

Use **** Diagram(s):

Web-based Gardening Care App

- registration
- login
- Update Profile
- View Plant Database
- Search Plants
  - <<include>>
  - View Plant Details
- Receive Care Recommendations
- Track Plant Growth
- Receive Notifications
- logout

- login
- manage plants
- manage Care Recommendations
- Manage Educational Content
- Manage Community Forum
- logout

Gardener

Supervisor

Home Owner

admin

Figure 1.1: * Case Diagram**

***** *Scenarios:*

*** Case 1: Registration

• **Use Case Title:** Registration
• **Use Case ID:** UC-01
• **Actors:** Gardener, Supervisor, **** Owner
• **Description:** Users can register themselves in *** system ** providing their details, such ** name, role, and email.
• **Alternative Path:** If the data ** ********** or invalid, *** system prompts the **** to re-enter the information.
• **Pre-Condition:** User must not have an existing account.
• **Action:**
1. User navigates to the registration page.
2. Fills ** ******** information (name, email, role, password).
3. Submits *** form.
4. ****** validates the **** and saves *** user profile.
5. Confirmation message is shown to the user.
• **Post-Condition:** User account is created successfully.
• **Exception:** ************ ***** if the email is ******* **** or data is invalid.
• **Author:** BC200405673

Use Case 2: Login

• **Use Case Title:** Login
• **Use Case ID:** UC-02
• **Actors:** Gardener, Supervisor, **** Owner, Admin
• **Description:** Users *** log in to the system using their registered email and password.
• **Alternative Path:** ** incorrect credentials are entered, the system ******** an ***** message.
• **Pre-Condition:** The **** must already ** registered.
• **Action:**
1. **** navigates to the login page.
2. Enters email and password.
3. ****** verifies credentials.
4. ** correct, the user is logged in and redirected to their dashboard.
• **Post-Condition:** User ** successfully logged **** *** system.
• **Exception:** Login fails if the *********** *** invalid ** *** ******* is locked.

• **Author:** BC200405673

Use Case 3: Update Profile

• ***** **Case Title:** Update Profile
• ***** ****** **ID:** UC-03
• **Actors:** Gardener, Supervisor, Home Owner
• **Description:** Users can ****** their personal details and gardening preferences in their profile.
• **Alternative Path:** If *** update fails, the system ******** *** user.
• **Pre-Condition:** **** **** be logged in.
• **Action:**
1. **** navigates to the profile section.
2. Updates ******** information or preferences.
3. Submits *** changes.
4. System validates and saves *** updates.
• **Post-Condition:** ******* is updated successfully.
• **Exception:** ****** ***** ** required fields *** ******* or invalid.
• **Author:** BC200405673

Use **** 4: View Plant Database

• **Use Case Title:** View ***** Database
• **Use Case ID:** UC-04
• **Actors:** Gardener, Supervisor, Home Owner
• **Description:** Users can browse the ************* plant database, including details **** types, care requirements, *** ****** stages.
• **Alternative Path:** N/A
• **Pre-Condition:** **** must ** logged in.
• **Action:**
1. User navigates to the ***** database.
2. Browses *** available plants.
3. Selects a ***** to **** detailed information.
• **Post-Condition:** User can view plant details.
• **Exception:** Database access fails due ** ****** issues.
• **Author:** BC200405673

Use **** 5: Search Plants

• **Use **** Title:** ****** Plants
• **Use **** ID:** UC-05

• **Actors:** Gardener, Supervisor, Home Owner
• **Description:** Users can search for ****** by name, category, ** attributes.
• *********** **Path:** If no results are found, a message ** ********* ** the user.
• **Pre-Condition:** User must be ****** in.
• **Action:**
1. **** ****** a ****** query.
2. System processes *** query and fetches matching results.
3. Results are displayed ** the user.
• **Post-Condition:** Search results *** displayed.
• **Exception:** Search fails if *** query is invalid or the database ** inaccessible.
• **Author:** BC200405673

*** Case 6: Receive **** Recommendations

• **Use Case Title:** ******* Care Recommendations
• **Use Case ID:** UC-06
• **Actors:** Gardener, Supervisor, Home Owner
• **Description:** Users receive personalized plant **** advice ***** on their preferences *** plant details.
• **Alternative Path:** N/A
• **Pre-Condition:** User must have a plant ********** in the system.
• **Action:**
1. **** views care recommendations.
2. System ********* advice ***** on ***** type, location, and climate.
3. Displays *************** such as watering ********* and pest ******* tips.
• **Post-Condition:** User receives care recommendations.
• **Exception:** System fails to generate recommendations due to missing data.
• **Author:** BC200405673

Use Case 7: ***** ***** Growth

• **Use Case Title:** Track Plant Growth
• **Use Case ID:** UC-07
• **Actors:** Gardener, Supervisor
• **Description:** Users *** track and monitor ***** growth, ********* health indicators and stages.
• **Alternative Path:** N/A
• **Pre-Condition:** **** must **** a plant registered in *** system.
• **Action:**
1. User records plant observations.
2. System logs and updates the growth data.

3. User *** review historical ****** data.

• **Post-Condition:** Plant growth **** is recorded and accessible.
• **Exception:** Tracking fails due to invalid data or system issues.
• **Author:** BC200405673

*** **** 8: ******* Notifications

• **Use Case Title:** Receive Notifications
• **Use Case ID:** UC-08
• **Actors:** Gardener, Supervisor, Home Owner
• **Description:** ***** ******* timely notifications *** tasks, ******* updates, or ***** alerts.
• **Alternative Path:** N/A
• **Pre-Condition:** User **** have a registered account.
• **Action:**
1. System sends relevant notifications.
2. User views and acts upon notifications.
• **Post-Condition:** Notifications *** ********* successfully.
• **Exception:** Notification ***** due to server issues.
• **Author:** BC200405673

Use Case 9: Manage Plants (Admin)

• **Use Case Title:** ****** Plants
• **Use Case ID:** UC-09
• **Actors:** Admin
• **Description:** ***** can add, update, or ****** plants in *** database.
• **Alternative Path:** N/A
• **Pre-Condition:** Admin must be logged in.
• **Action:**
1. Admin ********* to the plant management section.
2. Adds, updates, ** deletes plants.
3. ******* are saved ** the database.
• **Post-Condition:** Database is updated.
• **Exception:** Database update fails due to invalid data or ****** issues.
• **Author:** BC200405673

Use Case 10: Manage Educational Content (Admin)

• **Use Case Title:** ****** Educational Content
• **Use Case ID:** UC-10

• **Actors:** Admin	
• **Description:** Admin can add, update, or remove gardening ********* *** tutorials.	
• *********** **Path:** N/A	
• **Pre-Condition:** Admin must be ****** in.	
• **Action:**	
1. Admin navigates to the educational content section.	
2. Adds, updates, ** deletes resources.	
3. Changes are saved.	
• **Post-Condition:** *********** content is updated.	
• **Exception:** Content ****** ***** due to server issues.	
• **Author:** BC200405673	

Use Case 11: Manage Community Forum (Admin)

• **Use Case Title:** Manage Community Forum	
• **Use Case ID:** UC-11	
• **Actors:** Admin	
• **Description:** Admin moderates the community forum, including ***** and discussions.	
• *********** **Path:** N/A	
• **Pre-Condition:** ***** must ** logged in.	
• **Action:**	
1. Admin reviews posts and discussions.	
2. Approves, edits, or deletes content as necessary.	
• **Post-Condition:** ********* ***** ** moderated.	
• **Exception:** ***** moderation fails due to connectivity issues.	
• **Author:** BC200405673	

Adopted Methodology:

The approach ******** *** **** project is the V-Model, a ****** methodology that combines elements from **** the Waterfall model and the ****** model. This ******* approach ****** several key advantages, particularly in ***** of risk management. *** V-Model emphasizes thorough risk analysis, ***** allows for *** effective identification *** mitigation of ********* risks ** each phase of the project. Additionally, *** model ** known for its clarity and simplicity, making it **** easy ** understand and ********* for project teams.

To begin, ** will ***** explore *** fundamentals of the ********* model, ***** forms the backbone ** the V-Model's structure. This model ** sequential, **** **** phase ** *********** serving ** a foundation for the next, offering clear stages *** well-defined deliverables.

WATER FALL MODEL:
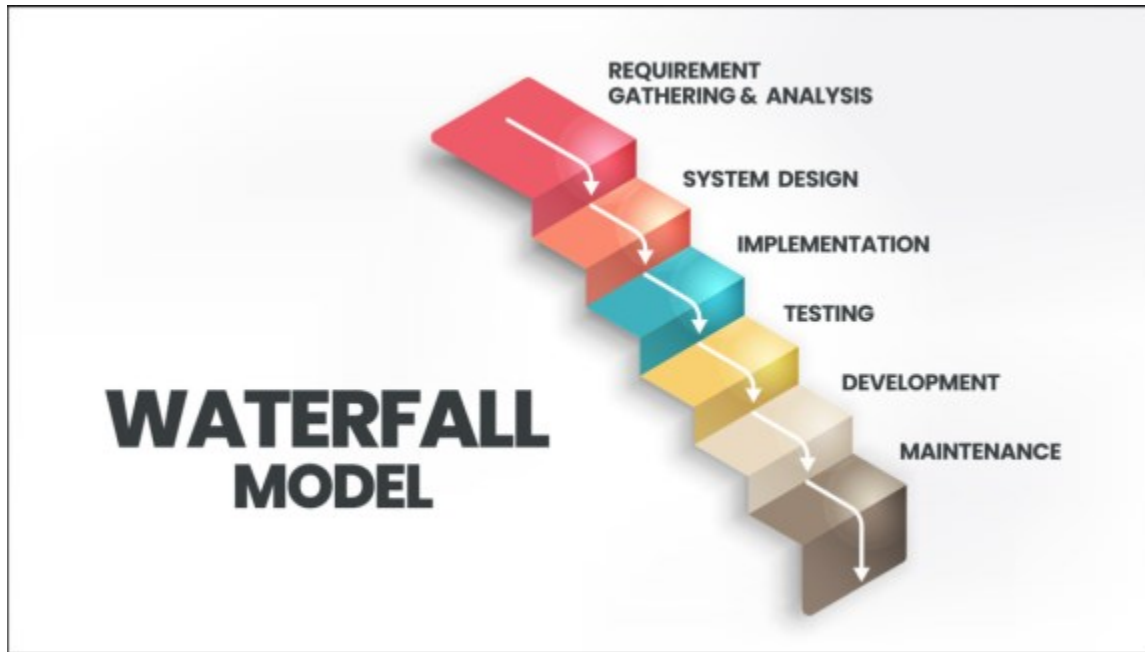
Figure 1.2: Waterfall Model

*** Waterfall ***** was *** first ******* model introduced in software development, often referred to as the linear-********** **** cycle model. ** ** a straightforward and easy-to-understand methodology. In the ********* Model, each phase **** be completed before the next one can begin, with ** overlap between phases. This sequential flow of stages offers a structured approach, making it one of the earliest and most ****** **** SDLC models. In *** Waterfall Model, development progresses in a linear sequence, meaning that one phase starts only after *** previous phase is fully completed. The ***** encompasses ******* ******** stages, beginning with the requirements phase *** concluding with the acceptance phase.

1. Requirement Gathering *** Analysis: This initial phase focuses on ********* and documenting all the ************ *** the system to be developed. It ********** ** the creation of a ******** *********** specification document, ***** ****** ** *** foundation *** *** next stages.

2. System Design: In this phase, *** requirement specifications **** the previous stage are ********** analyzed to prepare the system design. The system ****** ******* the hardware and software requirements, ** well as the overall architecture of the system, ******* *** groundwork for development.

3. Implementation: With *** system design in place, *********** begins. The system ** ***** in smaller units, each of which is ************ tested for functionality ** a process known as **** testing. These units *** later ********** into the full system ** *** next phase.

4. Integration * Testing:** **** all ********** units are developed and tested, they *** ********** into a complete system. The ****** is then subjected to ******** ******* to identify *** fix any ****** or failures.

5. Deployment: After the system has passed all functional and non-functional tests, it is deployed into the user environment ***** ** becomes operational.

6. Maintenance: After deployment, ****** may arise ** *** user environment. ***** are addressed ******* patches *** updates. Ongoing maintenance ensures **** the system continues to **** user needs *** remains functional.

Next, ** will explore *** Spiral Model and the V-Model.

Spiral Model:

The Spiral Model is employed to minimize the ***** associated with software development. This methodology is particularly useful in situations where the ********* risks could ************* affect the project, such as *** **** ** key personnel.

1. **Risk ********** Focus:** The primary strength of *** Spiral Model is *** ***** on risk assessment and mitigation. It allows for ********* ********** *** ********** at each phase, ******** that risks are identified and addressed early in *** process.
2. **Two *** Dimensions:** The Spiral Model operates on *** dimensions: the **###### dimension**, which represents *** ********** cost of the project up ** that point, and *** **angular dimension**, ***** tracks the progress **** through the spiral.
3. ********** *********** and Maintenance:** Unlike the Waterfall Model, the Spiral ***** allows for the development and *********** phases ** *** in parallel, making it suitable *** large-scale ** in-house software projects **** require continuous iteration *** adjustments.

** ******* MODEL:

Vu Process Model

<div align="center">
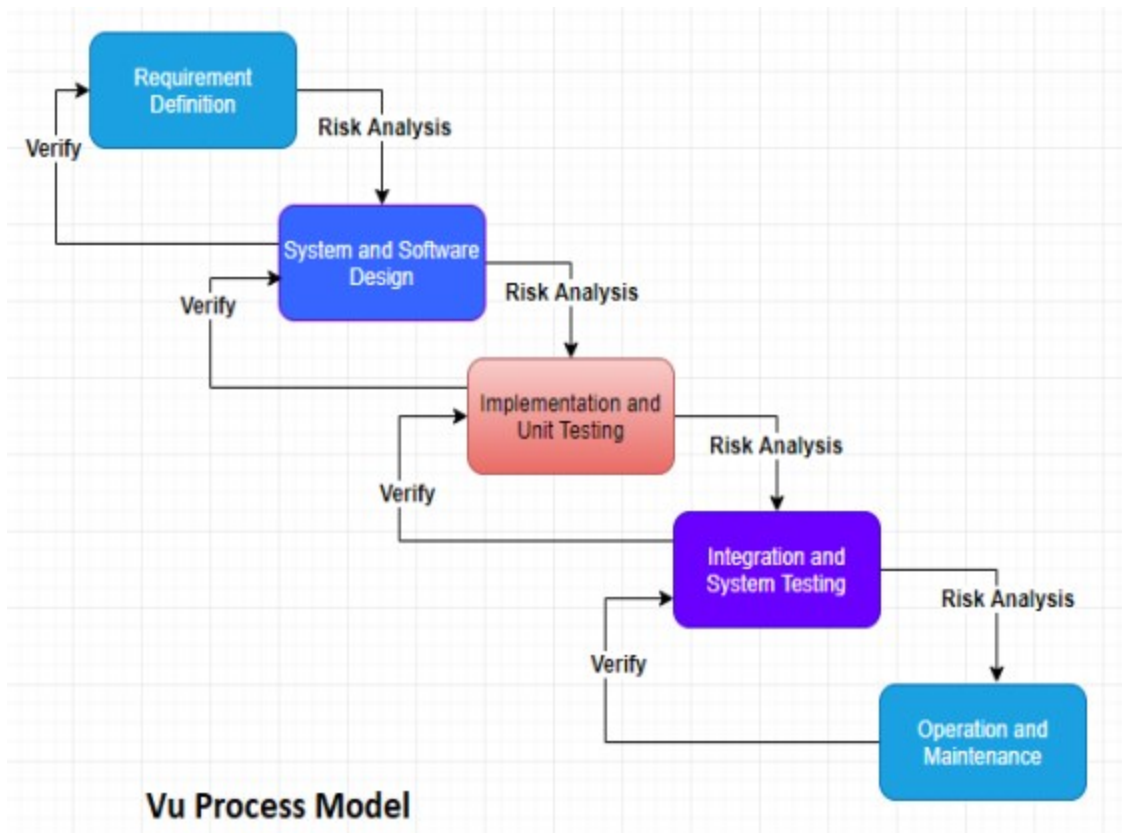
****** 1.3: VU Process Model

</div>

*** VU Process Model is a hybrid methodology that combines elements of **** the Waterfall *** ****** models. This integrated approach aims to ******** the quality of the system ***** ********** risks and ********** the disadvantages of each individual model. By merging these two methodologies, the VU Process Model ****** a structured yet flexible framework for development, ******** better control **** ******* ********* and more reliable outcomes.

******* for Choosing the VU Process Model:

The project has been ****** down into distinct stages, such ** requirement ********* and analysis, planning, design *** analysis, development, and the final report/viva stage. Each stage will be completed in sequence, *** at the end of each phase, it will be ********* ** our supervisor *** review. The ********** will provide ******** and suggest improvements *** the current ***** before ** proceed to the next one. Any necessary modifications **** be **** based on this feedback.
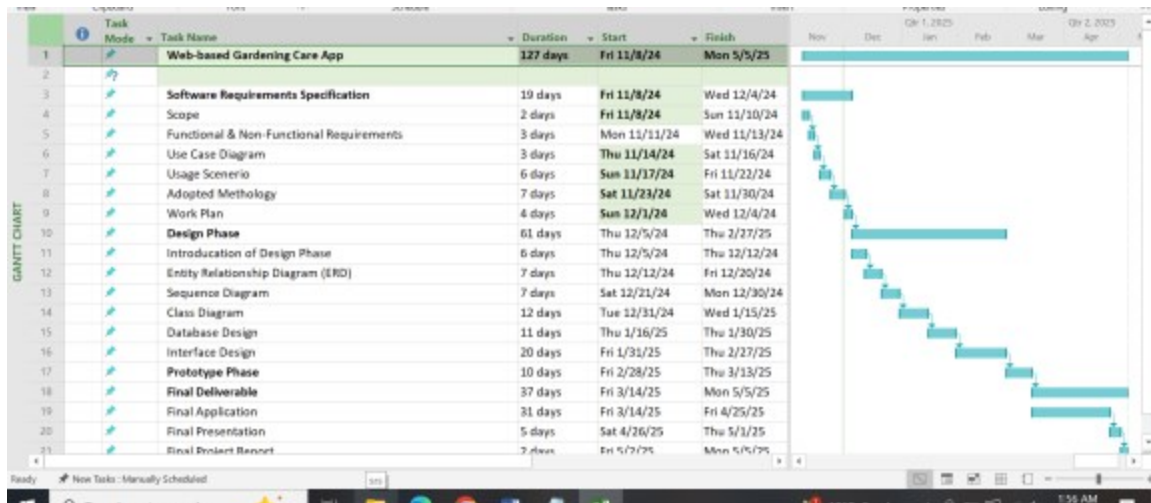
The ****** we are adopting this approach is due to the ********* nature of *** Spiral Model, which allows for continuous improvement at each phase. When a ***** is fully refined, well-executed, and approved ** *** supervisor, we will move on to *** next phase. This ensures that

each ***** is meticulously completed, aligning **** the ********* Model's **********
approach. ** combining ***** two models, the VU Process Model guarantees a systematic,
error-**** outcome, as each step is thoroughly reviewed and optimized before moving forward.

The VU Process Model ************ ******** of both the Waterfall and Spiral models,
offering a ******** approach to software development. This ****** methodology enhances
****** ******* while ********** risks and limitations. The VU ******* ******** of four
main stages, which ******* ****** the phases of the Waterfall model. ***** stages are
repeated until *** ****** ***** the client's requirements, ensuring a comprehensive and
iterative approach to development.

*Work Plan (Use ** Project to ****** Schedule/Work Plan):*

Figure 1.4: Work Plan Diagram