# Planit

The Plant-Based App

Ryan Tan Jinn En (U1920245A)

Neo Yong Tai (U1922050A)

Lee Yu Sheng Daniel (U1722891A)

Mamuduri Paulani (U1920947F)

Frankie Ye Htet Myat (U162116B)

Chen Xueyao (U1922640G)

Kundu Koushani (U1922997B)

Project Manager

Development Lead

Back-End Developer

Front-End Developer

Release Engineer / Manager

QA Manager

QA Engineer



#### **Contents**

- 1. Product Introduction
- 2. Design for Maintainability
- 3. Software Quality Assurance
- 4. Project Management
- 5. Risk Management







#### What on Earth is Planit?



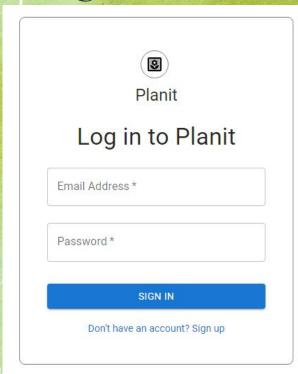
Plant Management App for all skill range of gardeners

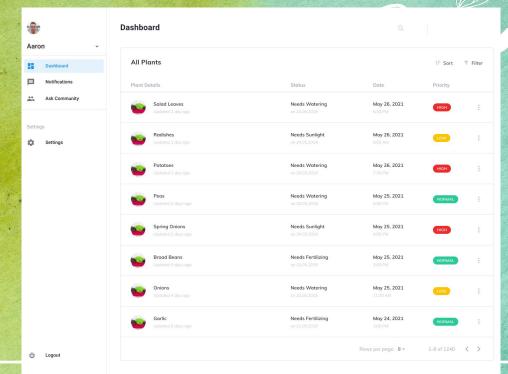


Organised gardening

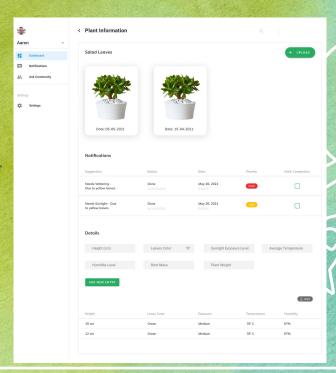


### **Login and Dashboard**





### Notifications & Plant details







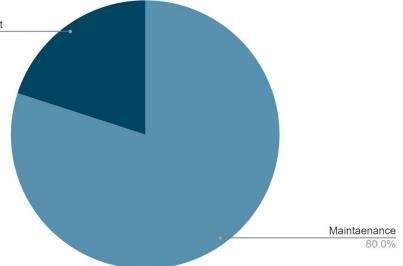
# Cost

Cost

- Maintenance - 80%

Software Development 20.0%

- Software Development - 20%



#### **Technical Consideration**

- Multi-platform Compatibility
- Faster time for development (Ease of integration from frontend to backend)
- Cost-saving (Pay per use)

- Frontend: React
- Backend : Node (Express JS)
- Database: MongoDB
- Server: AWS
- Frontend Testing: Jest/Enzyme
- Backend Testing: Mocha/Chai

# Design Pattern

State change

Dispatch call

React-Redux Hooks API

React components

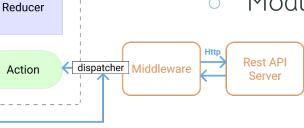
Store

State

#### Observer Pattern



- Separation of Concerns
- Modularity



# Software Management













## Software Maintenance

- Corrective Maintenance Use NPM to update packages when library change
- Preventive Maintenance React allows easy change of components
- Adaptive Maintenance AWS auto scaling for easy scaling of application
- Perfective Maintenance Modify notifications to be customised to suit users in the future



# **Software Standards**

- Functionality
- Usability
- Reliability

## **Product Assessments**

- CRUD operations on plant info
- User Interface Pages
- User Registration/Login
- Tasks
- Database system

# **Software Reviews**

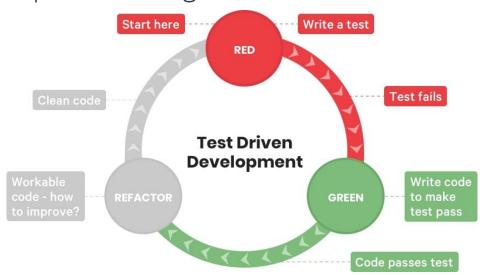
- Walkthrough
  - Q&A regarding software defects
- Technical Review
  - Peers examine the software product for client use and identify technical defects from specifications and standards
- Defects/poor quality in the software are identified early in the SDLC

# Software Quality Planning

- Requirement Management Process
- Change Management Process
- Maintainability Management Process
- Risk Management Process

# **Testing**

User Acceptance testing (UAT)



## **Standard Metrics**

- Mean Time to Failure
- Unit Test Coverage
- Number of defects in production
- Problems per User-Month during User Acceptance Testing
- Length of Code / Program Size in Number of Lines





# **Project Estimation**



### **Function Points**

| Element       | Complexity Detail |                                    |     |
|---------------|-------------------|------------------------------------|-----|
| Inputs        | Low               | Gathering Plant Information        |     |
|               | Low               | Gathering User Information         |     |
| Logical Files | High              | Plant Information                  | 100 |
|               | Medium            | User Information                   | 33  |
| Outputs       | High              | Display Search Results             |     |
|               | Low               | Display Plants                     |     |
|               | Low               | Display User Account Information   |     |
| Inquiries     | High              | Selecting Plant                    |     |
|               | Low               | Selecting Plant Notification       | 100 |
|               | Low               | Selecting User Account Information |     |
| Interfaces    | Medium            | PlantTypes, GrowthEnvironment      |     |
|               | Low               | Priority                           |     |



### **Unadjusted Function Points**

| Characteristic | Low | 141 | Mediu | ım   | High |      |
|----------------|-----|-----|-------|------|------|------|
| Inputs         | 2   | × 3 | 0     | × 4  | 0    | × 6  |
| Outputs        | 2   | × 4 | 0     | × 5  | 1    | × 7  |
| Inquiries      | 2   | × 3 | 0     | × 4  | 1    | × 6  |
| Logical Files  | 1   | × 7 | 0     | × 10 | 1    | × 15 |
| Interfaces     | 1   | × 3 | 4     | × 7  | 0    | × 10 |
| Unadjusted FP  | 30  |     | 28    |      | 28   |      |
| Total=L+M+H    | 86  |     |       |      | 100  | 20   |



- ▶ Influence Factors 14
- ▶ Total Influence Score 37
- ▶ Influence Multiplier 1.02



Total Lines of Code (29 LOC/FP) -  $2603\ LOC$ 





### **Distribution of Effort**

| 1990's Industry Data    | Work Package                    | Distribution | Estimates |  |
|-------------------------|---------------------------------|--------------|-----------|--|
| Preliminary Design      | Project Plan                    | 9%           | 48.24     |  |
| 18 %                    | Requirement Specification       | 9%           | 48.24     |  |
| Detailed Design         | User Interface                  | 7%           | 37.52     |  |
| 25 %                    | Technical Architecture          | 11%          | 58.96     |  |
|                         | Data Modeling                   | 7%           | 37.52     |  |
| Code & Unit             | Code & Unit testing             | 21%          | 112.56    |  |
| Testing 26 %            | Online Documentation            | 5%           | 26.8      |  |
| Integration & Test 31 % | Integration & Quality Assurance | 31%          | 166.16    |  |
|                         | Extrapolated total effort       |              | 536       |  |
| 9                       | 2% for project management       |              | 10.72     |  |
|                         | 3% for contingency              |              | 16.08     |  |
|                         | Total effort                    |              | 562.8     |  |





#### Hardware:

| Computers | DELL | 7 | \$1,000 | \$7,000 |
|-----------|------|---|---------|---------|

#### **Licensing:**



Total - \$83,000

#### **Software:**

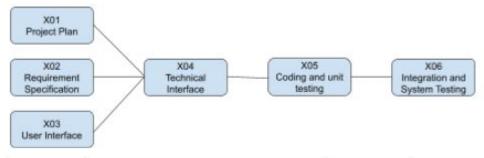
| Server   | AWS     | 1 | \$2,800  | \$2,800  |  |
|----------|---------|---|----------|----------|--|
| Database | MongoDB | 1 | \$15,000 | \$15,000 |  |

#### **Staff:**

| Project Manager         | 1 | \$30,000 | \$30,000 |
|-------------------------|---|----------|----------|
| Project Team<br>Members | 6 | \$10,000 | \$60,000 |



## **Activity Dependencies**



| Work                      |                               |         | Dependencies |  |
|---------------------------|-------------------------------|---------|--------------|--|
| Package #                 |                               |         |              |  |
| X01                       | Project Plan                  | 14 days |              |  |
| X02                       | XO2 Requirement Specification |         | 544          |  |
| X03                       | User Interface                | 24 days |              |  |
| X04                       | X04 Technical Architecture    |         | X01,X02,X03  |  |
| X05 Coding & Unit Testing |                               | 10 days | X04          |  |
| X06                       | Integration & System Testing  | 30 days | X05          |  |





#### Work Breakdown **Plantit Software System Project Plan** Requirement System System Implementation Specification Design High Level Architecture **UI** Design Platform Set-up Implementation System Data Modeling System Integration Generic Server Configuration Low-level Design System Testing Analysis & Requirement Requirement Specification Analysis Document System Requirement Specification Definition Review Software Review Project Schedule Review Plan Approval Process Resources Constraints

## **Project Timeline**

| Tasks                                 | Project Timeline |       |       |       |       |       |       |       |       |
|---------------------------------------|------------------|-------|-------|-------|-------|-------|-------|-------|-------|
|                                       | 23/08            | 30/08 | 06/09 | 13/09 | 20/09 | 27/09 | 04/10 | 11/10 | 18/10 |
| Project planning and setup            |                  |       |       |       |       |       |       |       |       |
| Design Sprint                         |                  |       |       |       |       |       |       |       |       |
| Establish Use Cases                   |                  |       |       |       |       |       |       |       |       |
| Plan epics, sprints and backlog       |                  |       |       |       |       |       |       |       |       |
| Craft Project Proposal                |                  |       |       |       |       |       |       |       |       |
| Product Design                        |                  |       |       |       |       |       |       |       |       |
| Low-Fi wireframe                      |                  |       |       |       |       |       |       |       |       |
| High-Fi wireframe                     |                  |       |       |       |       |       |       |       |       |
| Map User Journey                      |                  |       |       |       |       |       |       |       |       |
| Interactive Prototype                 |                  |       |       |       |       |       |       |       |       |
| Development                           |                  |       |       |       |       |       |       |       |       |
| Establish server and databases        |                  |       |       |       |       |       |       |       |       |
| Backend implementation                |                  |       |       |       |       |       |       |       |       |
| Login/Registration View               |                  |       |       |       |       |       |       |       |       |
| Main Panel View                       |                  |       |       |       |       |       |       |       |       |
| Add Plant Info View                   |                  |       |       |       |       |       |       |       |       |
| Notifications and Alerts              |                  |       |       |       |       |       |       |       |       |
| Edit/Delete Plant View                |                  |       |       |       |       |       |       |       |       |
| QA & Testing                          |                  |       |       |       |       |       |       |       |       |
| Write Unit Tests                      |                  |       |       |       |       |       |       |       |       |
| Set criteria for User Acceptance Test |                  |       |       |       |       |       |       |       |       |
| Finalize UAT                          |                  |       |       |       |       |       |       |       |       |
| User checks UAT of initial prototype  |                  |       |       |       |       |       |       |       |       |
| User checks UAT of final prototype    |                  |       |       |       |       |       |       |       |       |

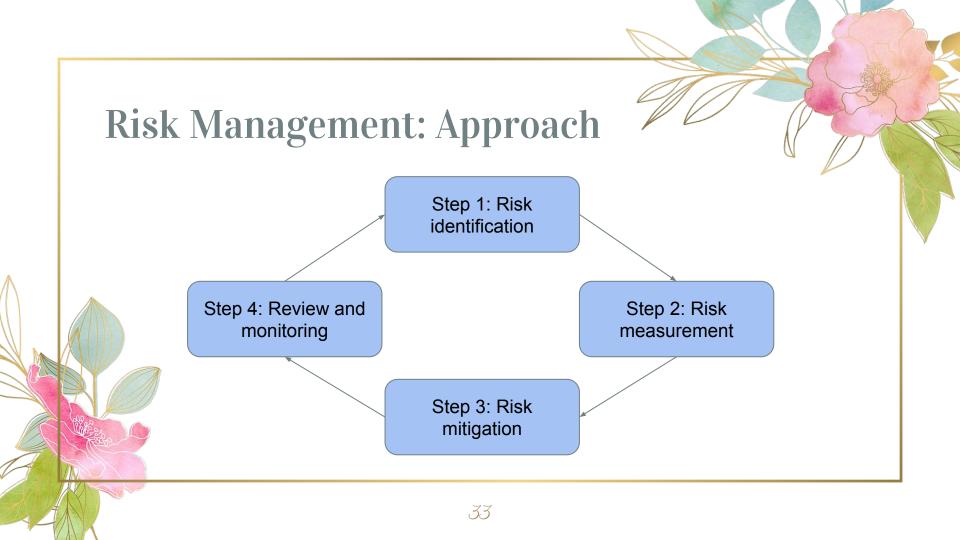




### Risk Management: Objectives

- Plan ahead so that the team can adapt to unplanned incidents, minimizing negative impacts on the project.
- Ensure entire team is in agreement with how to deal with risks.

Ensure document exists to keep track and log the team's discussions with respect to project risk.



## Risk Management: Risk identification

- Comprises of : Project Team + Stakeholders
- Evaluate:
  - Environment Factors
  - Organizational Culture
  - Project Management Plan



Identified risks need to be accessed and processed:

Qualitative

Quantitative

Risk Management: Qualitative Risk

Analysis

Defined using

Probability of occurrence of risk

High: P > 70%

Medium: P = 30 - 70%

Low: P < 30%

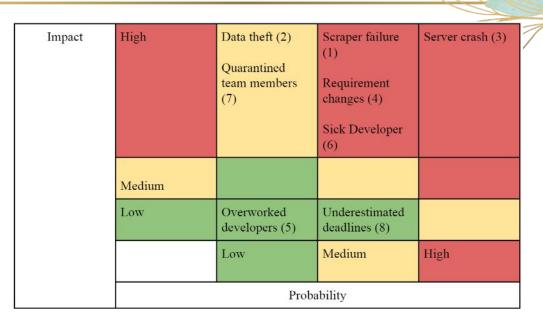
| Medium | High   | High   |
|--------|--------|--------|
| Low    | Medium | High   |
| Low    | Low    | Medium |

Probability

High: greatly impacts project

Medium: slightly impacts project

Low: relatively little impact on project



Risks that fall within the RED and YELLOW zones will have risk response planning, which may include both risk mitigation and risk contingency plan.

# Risk Management: Quantitative Risk Analysis

| Risk                                  | Severity | Likelihood | Level of Control | Significance |
|---------------------------------------|----------|------------|------------------|--------------|
| People                                | 2        | 2          | 3                | 7            |
| Technology                            | 3        | 1          | 1                | 5            |
| Requirement<br>Changes                | 3        | 3          | 1                | 7            |
| Underestimated<br>Development<br>Time | 3        | 2          | 1                | 6            |



# Risk Management: Quantitative Risk Analysis

Risks found are prioritized by estimating their effect on project activities, and numerical rating of 1-3 is given

- Severity: how much the project is affected
- Likelihood: probability of it occurring
- Level of control: control we have over a particular risk type
- Significance: sum of above 3 attributes

# Risk Planning

- Avoid
- Mitigate
- Accept
- Transfer



# **Risk Monitoring**

Constantly identify, analyze, and plan for new risks:

- Weekly scrum meetings
- Regular test runs

Monitoring previously identified risks:

Re-prioritize risks depending on probability and complexity



Risk type: People

Possible Risks:

Overworked Developers

Sick Developers

Probability (1-3): 2



Effects: Additional backlog due to lack of members / fatigue

Approach: Mitigate

#### Plan:

- Find temporary replacements
- Send regular updates to developers to keep them updated

Risk type: Technology

#### Possible Risks:

- Server Crashes and Loss of Data
- Data Theft

Probability (1-3): 1



#### Effects:

- An obsolete backend
- Loss of client trust
- Lost of client data

Approach: Mitigate

#### Plan:

Spend additional resources to update backend

Regain client trust with the help of professionals

Invest in a higher security database to prevent further data breaches



Risk type: Requirement changes

#### Possible Risks:

Complaints from users

New competitors with better UI/functionalities

Probability (1-3): 3

#### Effects:

Having to spend extra resources for perfective maintenance

Approach: Accept

#### Plan:

Listening to user complaints, and adding new features to keep up with the competitors

Risk type: Underestimated development time

#### Possible Risks:

Project deadlines not met

Complains from stakeholders

Probability (1-3): 2

#### Effects:

Delayed release of specific components or product

Over-dedication of resources to meet deadline

Approach: Avoid

#### Plan:

Devise a more concise and detailed plan, discussed and agreed by all team members in the project.



