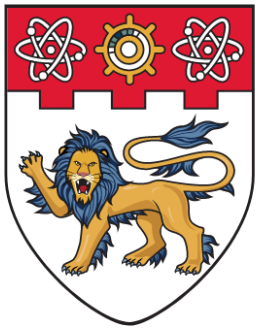


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

System Requirement Specifications (SRS) for Planit:

Team Members	Role
Ryan Tan Jinn En	Project Manager
Neo Yong Tai	Development Lead
Lee Yu Sheng Daniel	Back-End Developer
Mamuduri Paulani	Front-End Developer
Frankie Ye Htet Myat	Release Engineer / Manager
Chen Xueyao	QA Manager
Kundu Koushani	QA Engineer

Name of Design Team: **Team Syan**
School of Computer Science and Engineering, Nanyang Technological University

1 Table of Contents

2 Problem Statement

The recent Covid-19 Pandemic has forced the Singapore government into adopting the long term goal of having 30 percent of the total food consumption be sourced locally by 2030. However, the actual adoption rate by the local population has been slower than expected. Problems such as lack of time, insufficient resources and highly discouraging results have been cited as the main cause. Despite the National Environmental Agency and the National Parks Board providing a repository of information related to starting and sustaining a home grown garden, users still find trouble in dissecting relevant sectors specific to their problem.

3 Overview

3.1 Background

Following the recent interruption of trade routes due to the Covid-19 pandemic and political unrest, the need to find new avenues to counteract this problem has arisen. One of the recent approaches taken by the Singapore government was to encourage the growth of local gardening communities around the country. This was achieved through the handing out of complimentary seed packets, the construction of more community garden spaces, and increasing awareness through advertising campaigns.

Due to the cultural needs of most Singaporeans, the plan has not seen full nationwide adoption. Actions taken by the government, such as increasing information repositories and holding seminars are meant to act as a remedy to the

ongoing problem. There also exist several applications that aim to help gardeners with their plants. However, there are some limitations to these applications that ultimately hinder the user experience rather than aid them.

3.2 Overall Description

The application will provide visual references for a variety of gardens, monitoring tools for specific crop growth, and streamlined guidelines plus solutions for easy user reference. Its features will remove the guesswork involved in gardening, ensuring gardeners with peace of mind.

4 Investigation & Analysis Methodology

4.1 System Investigation

Planit is being developed as a web application for gardeners with the aim of streamlining the learning and growing process for a variety of home garden plants. The main users will be a mix of beginner and experienced gardeners. Existing software and resources will be researched thoroughly in order to gleam both their useful attributes as well as their shortcomings in the eyes of users. Existing gardening software is tailored more towards farmers who manage a large crop size, usually resulting in a complex spreadsheet interface that is not intuitive to less experienced gardeners as well as software users.

With Planit, information will mainly be presented to users graphically and information will be condensed into bite sized easy to understand segments, leaving gardeners with just the physical work of starting and maintaining a garden. The application will also allow them to keep track of their progress and harvests, providing the user with encouragement to sustain their efforts. A cloud-based database and server will store all crucial information, ensuring safe storage of user data.

4.2 Analysis Methodology

4.2.1 Feasibility study and requirements elicitation

Organize a development and implementation team composed of people knowledgeable about the process of growing plants with which regular meetings will be held. A series of interviews with the managers and the developers of the current plant management system will be arranged. Interview and feedback from the personnel and staff working directly with the system is needed to define the current environment and future system requirements. A Feasibility and Risk Assessment study will be conducted to determine which solution(s) are most appropriate based upon the results of the interviews.

4.2.2 System analysis and requirements specification

4.2.2.1 Perform an analysis of the problem using object-oriented techniques

An external view of the model of the Planit app including plant monitoring information, task notifications, and search queries will be developed using Unified Modeling Language (UML). This SRS document will form part of the documentation for the project. Some desired features of the proposed system include:

Some desired features of the new system include:

- The ability to view plants that you are maintaining
- Provide tasks to be completed for the week
- Search for communities and forum for answers to queries
- Login and Logout

4.2.2.2 Scope and Limitations

Analysis methodology will involve requirement analysis, data analysis, and application architecture:

Requirement analysis:

1. The application will be developed for the web browser.

2. The application will allow the user to login with their email address and password registered through our system, to view, perform CRUD operations on the plants that they are managing, see upcoming tasks, warnings and special instructions if supplied by the user. The admin will be able to delete users as well as modify the warning tasks if it is applicable.
3. For secure storage of passwords in MongoDB database, JSON Web Tokens will be used.

Data analysis:

1. All data will be extracted through the Express server and stored into the MongoDB database.
2. Retrieval of data will be done through fetch instructions from the front-end to API endpoints.
3. Data will be manipulated using Mongoose schemas with MongoDB.
4. Data validation will be done using Express.

Application architecture:

1. Information will be stored as Schemas in MongoDB. The schemas will be for the User and Admin.

4.2.3 Object-oriented design using UML

A detailed object-oriented design for the plant management system will be developed. UML will be used again for the graphical representation and documentation of the design. The system will primarily concern itself with the plant monitoring.

4.2.3.1 Use Case 1

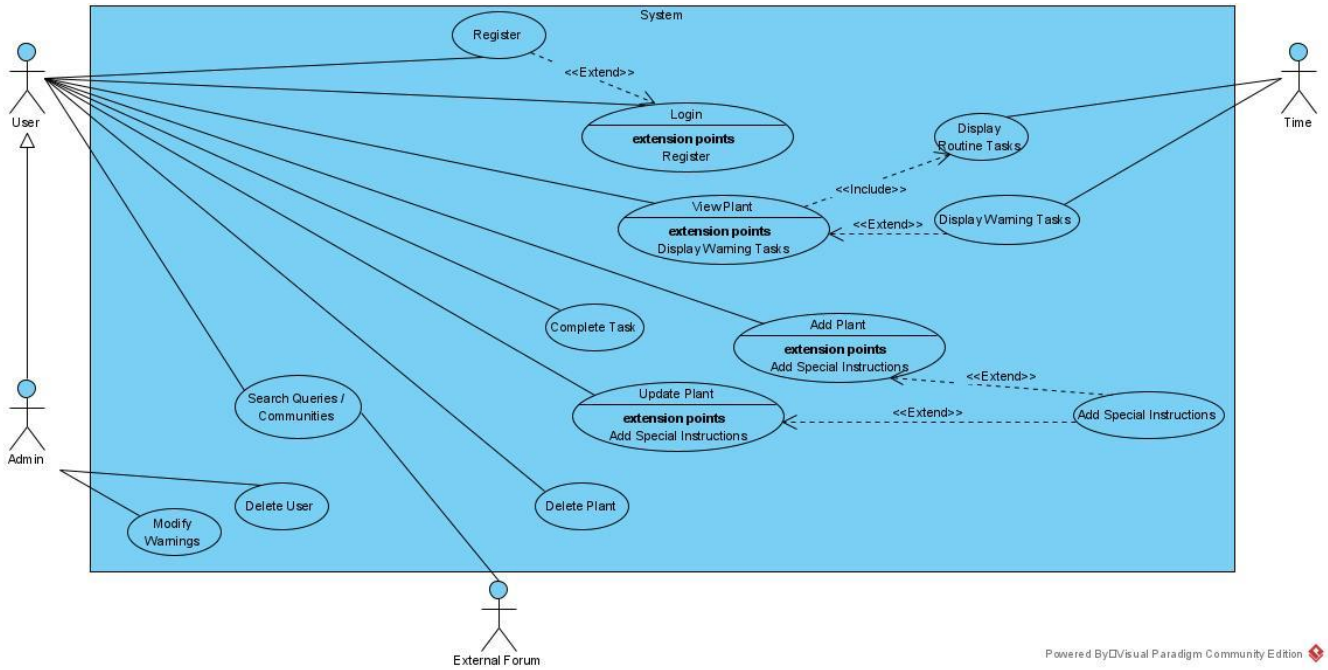


Figure 1: Use case Diagram for Planit

4.2.4 Prototyping

Figma, an online tool, will be used for implementing a limited and functional prototype for the plant management system. The prototype will include the end user interface, and at least have a working example for demonstration purposes. This prototype will be handed over to the implementation team.

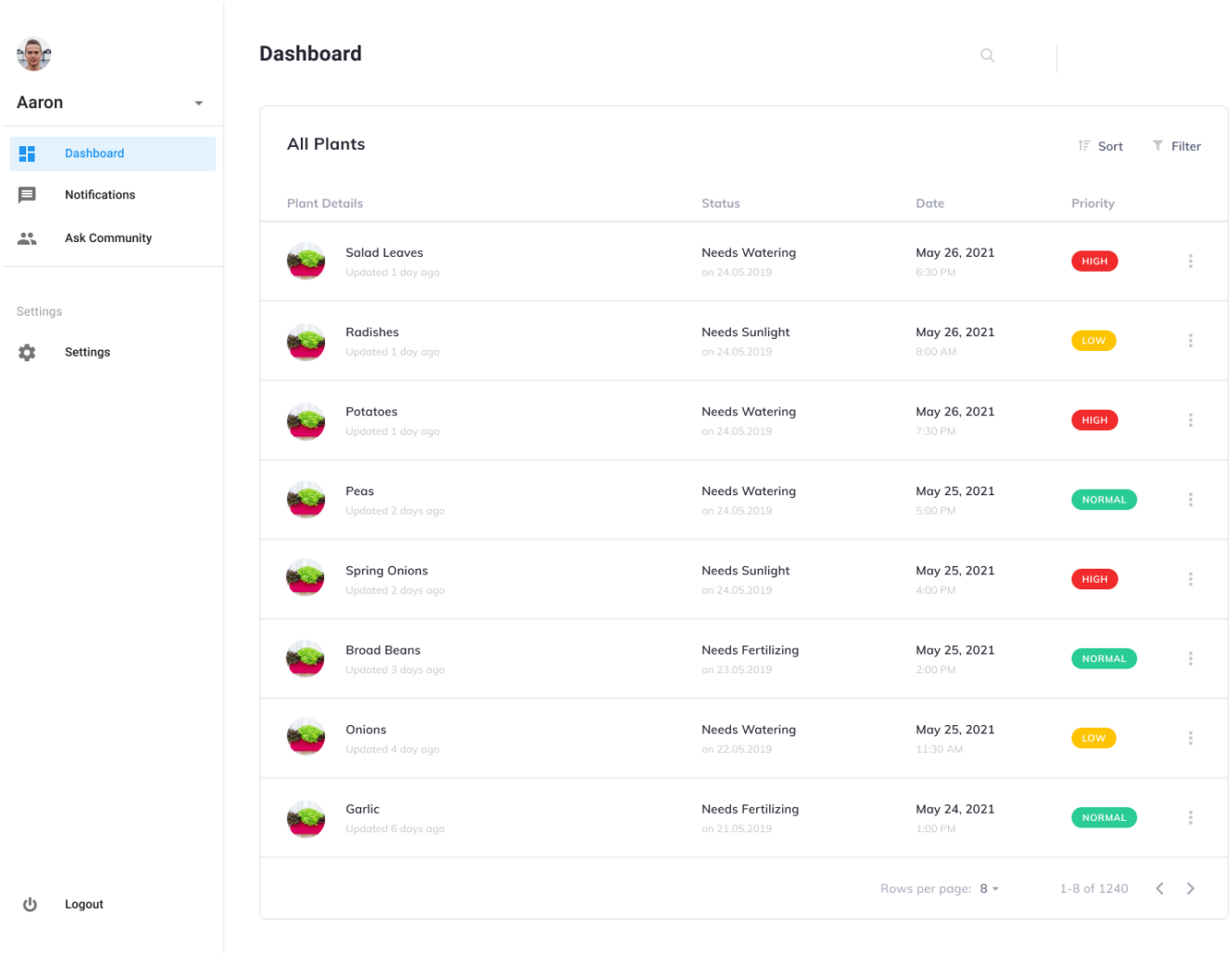


Figure 2.1: Samples of Figma Prototypes for Planit Dashboard

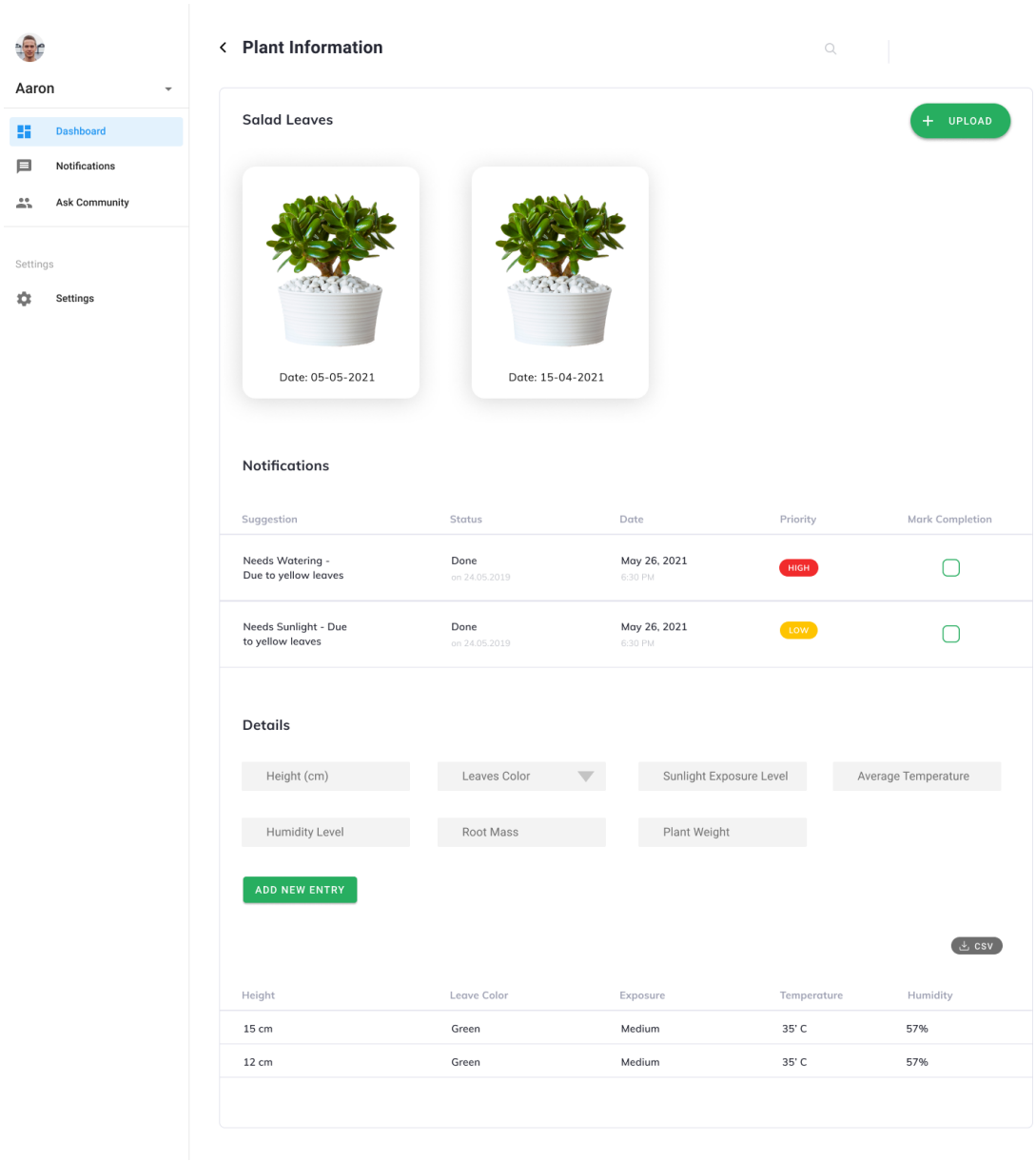


Figure 2.2: Samples of Figma Prototypes for Planit Plant Info

5 Constraints

5.1 Scalability

The Planit system scales well to increasing system demands. Error handling is limited to few anticipated or common errors.

5.2 Data and Function Mapping

Adding new plants or updating the monitoring system can be readily updated and used across devices using the same account. New functions of the system can be easily mapped to the existing Planit system.

5.3 Proprietary hardware and software

Planit system requires an updated chrome browser, version 93.0.4577.63 with active internet connection to run on any device that supports web browsing.

5.4 Batch updates vs. (close) Real-time updates

There is no real-time update of mainframe DB2 registration system data for transactions through the Planit system. Accumulated transaction records are applied overnight via a scheduled job.

5.5 Project Schedule

There is a three-month timeframe to implement a production system of the Planit system from project commencement.

6 Operational Requirements

6.1 Help Desk Support

System will provide a contact us page with an email to contact us. System users can email questions that are technical in nature, such as, slow or sluggish system response time, incompatible browser features, application errors, system downtime inquiries, account lock-out assistance, etc.

6.2 Application Services and Technical support

Programmers and application developers will have access to source code to address bugs or system enhancements as deemed necessary. Network Administrator and DBA support is also required to maintain a 24x7 system uptime.

6.3 Administration Features

System security and access levels are provided in the online system. There are varying levels of system access and functional authority. Each user's access is limited to his/her own registration records. Only authorized system administrator(s) has access to all user registration records and the ability to modify warnings.

6.4 System Interface

The system will remain operational and is independent of the users' use of the system. At any given time, multiple users can use the system.

6.5 System hardware fail over and routine back up

The back-end developers will handle system hardware tasks such as data tape back-up, hardware maintenance, fail over, scheduled system patches and maintenance.

6.6 Audit Trail

System audit trails will log all transaction records to capture what action was taken, when (time-stamp) the transaction occurred and who made the transaction. The records will be stored for one month before being discarded.

7 Functional Requirements

Planit is a "plant monitoring" system that assists users in the management of their plants and search for queries when they need help.

7.1 User

7.1.1 Login / Register

Login
Register
Grant admin privilege

7.1.2 View Plants

Display routine tasks
Display warning tasks
User updates status that task has been completed

7.1.3 Modify Plants

Add a plant
Update a plant information

Delete a plant
Add special instructions for specific plant

7.1.4 Search

User input queries to external search engine
Display search results

7.2 Admin

7.2.1 Delete User

Delete a user

7.2.2 Modify Warnings

Update warning task
Delete warning task

8 Input Requirements

8.1 User account authentication and user access

Each user logs into the system using their email address and password. The email address will serve as an identification purpose to their account and future access. This email and password maps to his/her registration record information in our system. When the user no longer wants to use the system, he/she will request for deletion for the admin to get the registration record disabled.

8.2 Search Function

Users are required to input text in the search bar. Once it has been entered, the query result will be returned by the external engine and displayed on the web browser.

8.3 Add Plant Function

Users can add plants of their choosing from a drop down list of types, variety and species. They are required to enter the estimated amount of seeds per plant sowed as well as the date and time they did so. The information gathered will be reflected in the virtual plot of land and notifications relevant to that specific plot will be set as a reminder for the gardener to perform the required tasks in the future.

9 Process Requirements

The following are among the inherent requirements that the online registration system must be able to handle.

9.1 MongoDB Transaction

The system must be able to send, trigger and receive updates from the database.

9.2 Data Integrity

1. All data sent to the database must be accurate and consistent.

2. The system must verify the integrity of the data sent to the database.
3. The system must verify the integrity of the data received from the database.
4. The system must commit transactions that are completed.
5. The system must rollback unfinished transactions or time-out transactions.

9.3 Data validation

1. The system must validate all user inputs before sending the data to the database.
2. The system must be able to perform data validation and error-handling routines in all related layers of the system architecture.
3. The system must be able to prevent data error from the frontend and backend.

9.4 Performance

1. The system must be able to handle concurrent use of the system on a 24x7 basis.
2. Live changes to the garden are reflected immediately on the visual element of the application.
3. Notifications for all daily tasks should be refreshed and updated at exactly 00:00 daily.

9.5 Data repository

The system must maintain the existing MongoDB database as the main repository of data.

10. Output Requirements

10.1 User Account

All user accounts should contain the user's personal information. The user account should allow users to add, delete or update their plants, view plant growth reports and perform community actions such as creating a post or commenting on a post in the community forum.

10.2 Plant Growth Record

The Plant Growth Record should be kept hidden from the user but accessible on demand. The record should have information on the plant's species, the age of the plant, the plant's height over time and number of fruits (if any) over time.

10.3 Plant Growth Chart

The system must provide each plant in an user account with a Plant Growth Chart that visualizes the plant's Plant Growth Record. The Plant Growth Chart should contain all information from the Plant Growth Record.

10.4 Search Web Results

Each user's search filter should return results containing specific gardening-related forum posts, replies, images or text posts found on social media platforms.

11. Hardware Requirements

11.1 Network

1. Mobile users must have a cellular or Wifi connection.
2. Desktop users must have an ethernet or Wifi connection.

11.2 Client Computers

1. Mac, Unix and Windows client computers
2. iOS or Android client mobile devices

11.3 Server

1. 24-hour uptime server with a backend database.

11.4 Production support systems

AWS support (back-up tapes, redundant drives, UPS, etc.)

12 Software Requirements

12.1 Client Operating System

1. Any operating system that supports web browsing: MacOS, Windows, iOS, Android, etc.

12.2 Client Application

1. Any Javascript compatible browser: IE, Chrome, Firefox, etc

12.3 Network system

Network software and protocols in order for systems to communicate:

- TCP/IP
- HTTP
- HTTPS
- FTP

12.4 Mainframe system

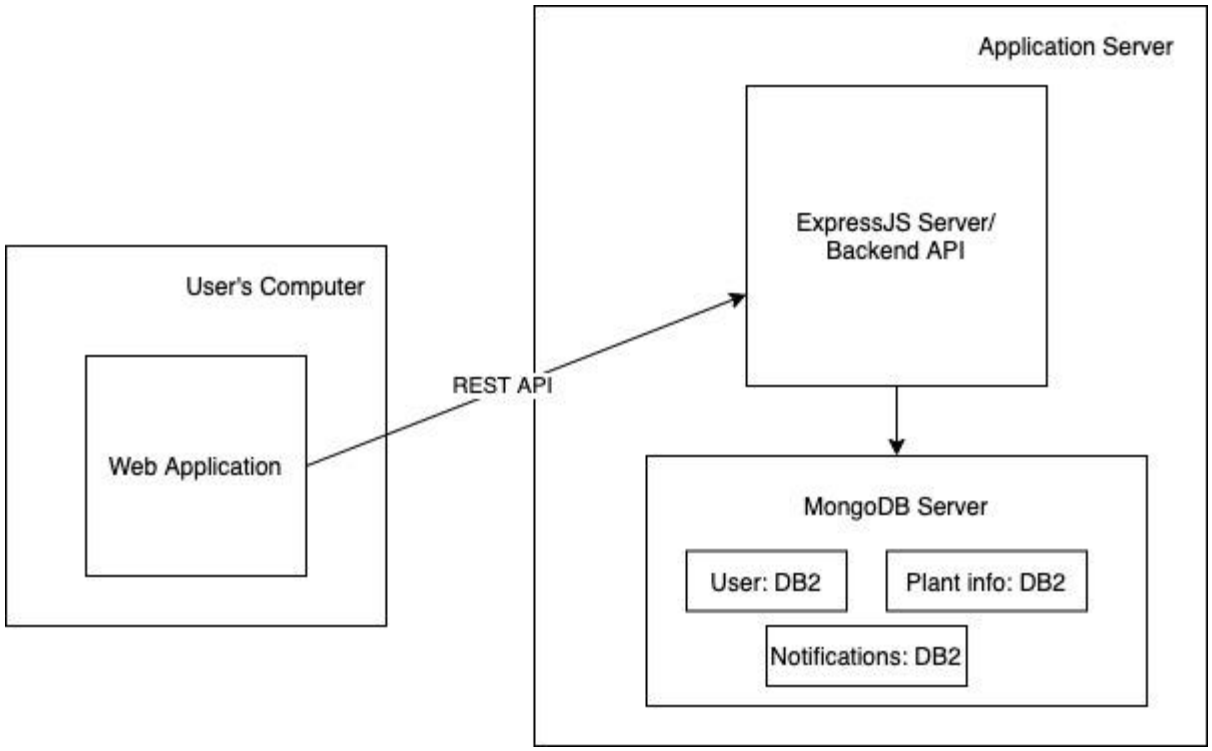
1. MongoDB Database

12.5 Licenses

Valid licenses are required to run software from third party vendors:

- To use application development tools
- To use web server, application server and database software in development, test and production mode

13 Deployment Requirements



The backend API developed on the ExpressJS server will be responsible for fetching and writing data. It will be deployed on an AWS S3 instance with Linux environment. The user and plant information data will be stored on MongoDB (database) deployed on Mlab. Finally, the web application will be rendered client side, developed using ReactJS framework.