

## **Experiment No. 3**

**Aim: Develop SRS document in IEEE Format for the project.**

# **Project Title: UrbanGarden**

**Mohib Abbas Sayed - 2103158**

**Hamza Sayyed - 2103159**

**Om Shete - 2103163**

# **Software Requirements Specification**

**For**

**Urban Garden**

**Prepared by**

**Mohib Abbas Sayed**

**Hamza Sayyed**

**Om Shete**

**Thadomal Shahani Engineering College**

**31<sup>st</sup> July 2023**

# Table of Contents

<b>Table of contents.....</b>	<b>ii</b>
<b>Revision History.....</b>	<b>iii</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Product Scope.....	2
1.5 References.....	2
<b>2. Overall Description.....</b>	<b>2</b>
2.1 Product Perspective.....	2
2.2 Product Functions.....	3
2.3 User Classes and Characteristics.....	3
2.4 Operating Environment.....	3
2.5 Design and Implementation Constraints.....	4
2.6 User Documentation.....	4
2.7 Assumptions and Dependencies.....	5
<b>3. External Interface Requirements.....</b>	<b>6</b>
3.1 User Interfaces.....	6
3.2 Hardware Interfaces.....	6
3.3 Software Interfaces.....	6
3.4 Communication Interfaces.....	6

<b>4. System Features.....</b>	<b>7</b>
4.1 Authentication and Authorization.....	7
4.2 Add Nursery Feature.....	8
4.3 Add Photographs Feature.....	8
4.4 Add location Feature.....	9
4.5 Search Nurseries Product Feature.....	9
4.6 Shortlisting Nurseries Product Feature.....	10
4.7 Delete Nurseries Feature.....	10
4.8 Deleting Shortlisted Nursery Product Feature.....	11
<b>5. Other Nonfunctional Requirements.....</b>	<b>12</b>
5.1 Performance Requirements.....	12
5.2 Safety Requirements.....	12
5.3 Security Requirements.....	12
5.4 Software Quality Attributes.....	13
<b>6. Other Requirements.....</b>	<b>14</b>
<b>Appendix A: Glossary.....</b>	<b>15</b>
<b>Appendix B: Analysis Models.....</b>	<b>16</b>
<b>Appendix C: To Be Determined List</b>	

**Revision History**

<b>Name</b>	<b>Date</b>	<b>Reason For Changes</b>	<b>Version</b>

# **1. Introduction**

## **1.1 Purpose**

The purpose of this report is to provide an in-depth analysis of the key elements needed to build a successful nursery plant e-commerce website. With the global online plant delivery market experiencing rapid growth, it is essential for nurseries to establish an online presence and provide customers with a convenient and efficient way to purchase plants.

## **1.2 Document Conventions**

Key points have been underscored to highlight significance. Main headings and subheadings are emphasised in bold for added distinction. All points within sections are structured by priority, ensuring essential elements are not overlooked. Abbreviations, familiar to application developers, are employed in specific instances to enhance communication efficiency.

## **1.3 Intended Audience and Reading Suggestions**

The target readership comprises the group of developers tasked with conceptualising and executing the Urban Garden project. Additionally, the content is intended for utilisation by the testing team responsible for scrutinising and assessing the application's performance and design. This document encompasses comprehensive essential details tailored for the software engineering team engaged in the project's execution.

## **1.4 Product Scope**

UrbanGarden encompasses an e-commerce platform enabling nursery vendors to seamlessly register, manage shops, and showcase plant varieties and gardening products. Customers benefit from an intuitive interface, shopping cart, secure checkout, and user profiles. Interactive forums foster community engagement. Admin tools ensure smooth platform management. UrbanGarden also recommends a variety of products based on the location of accessing the website. Prioritised for security, and performance. Thorough testing, documentation, and customer support ensure a seamless experience. UrbanGarden aims to create a vibrant online gardening hub, connecting vendors and customers in a user-friendly ecosystem.

## **1.5 References**

Websites:

- MaterialUI: <https://mui.com/>
- Redux: <https://redux.js.org/>
- w3schools: <https://www.w3schools.com/>



## **2. Overall Description**

### **2.1 Product Perspective**

The traditional method of buying and selling nursery plants involves physical visits to local stores, limited choices, and minimal information. In contrast, the Urban Garden platform revolutionises this process by providing vendors with an easy way to register and showcase their products online, offering customers a wide selection, rich product details, secure transactions, and a vibrant community for knowledge sharing. It transforms nursery shopping into a convenient, diverse, and interactive experience that transcends geographical limitations.

### **2.2 Product Functions**

- **Streamlined Transaction Process:** UrbanGarden eliminates the requirement for intermediaries in the buying and selling of nursery products, fostering a direct interaction between vendors and customers, resulting in efficient and transparent transactions.
- **Effortless Product Search:** Users can effortlessly explore a diverse range of nursery items from various locations, saving them the need to physically travel, and thereby enhancing convenience.
- **Enhanced Communication:** The platform facilitates direct communication between buyers and nursery vendors, enabling easy clarification of queries and concerns, thus promoting effective engagement.
- **Time Efficiency:** UrbanGarden optimises the buying and selling process, conserving time for both parties involved by offering a seamless online platform for transactions.

## **2.3 User Classes and Characteristics**

UrbanGarden accommodates two distinct user classes: Vendors and Customers. Vendors, comprising nursery owners, engage through the Vendor interface, enabling seamless product showcasing, inventory management, setting up their profiles and direct interaction with customers. Customers, utilising the Customer interface, enjoy convenient browsing, purchase options, and engagement with vendors, fostering a vibrant gardening community.

## **2.4 Operating Environment (OE)**

Since the application is a web application it can work in any browser.

- Device: Computers, Laptops, Tablets.
- Operating System: Windows, Linux distributions, Mac OS, Android
- RAM: 3GB or more
- Disk Space: 20 MB or more.
- Browsers: Mozilla Firefox 30+, Google Chrome 27.0+, Microsoft Edge.
- Internet connection: Strong internet connection with a speed of at least 1 Mbps for the best experience.

## **2.5 Design and Implementation Constraints**

CO-1:

The time allotted for this project is at most 3 months.

CO-2:

The front end of the application is made using ReactJs, and MaterialUI.

CO-3:

NodeJs and ExpressJs will be used as the language for the backend of the application and MongoDB will be used for the database of the application.

CO-4:

The website will be in the English language. Users who do not know English will face difficulties in using the website.

## **2.6 User Documentation**

Comprehensive guidance will be seamlessly integrated throughout the application's user journey to ensure a user-friendly experience. Future enhancements include the incorporation of a user-assisting chatbot. Clear instructions will be available during form completion, photo uploads, and location inputs. Inaccurate or erroneous entries will trigger informative error messages, preventing user frustration and enhancing application usability.

## **2.7 Assumptions and Dependencies**

AS-1:

The application supports only the English language. We assume the users of the application will be well-versed in English.

AS-2:

The users of the application should have basic knowledge of uploading images and location.

*Software Requirements Specification for Urban Garden*

*x*

DE-1:

The application will require the MERN framework as a dependency since we have used ExpressJs as the backend language.

DE-2:

ReactJs Framework will be used for the front end of the application.

DE-3:

\*For maps and geolocation, we will be using Mapbox APIs.

### **3. External Interface Requirements**

#### **3.1 User Interfaces**

UI-1:

The website will start with a landing page. The landing page will have all information about the web application. It will include instructions for maintaining the smooth functioning of the web application.

UI-2:

There will be a navigation bar at the top of the web page which will help users to navigate to different web pages. This will also include a search bar to provide a smooth search of desired/required products.

UI-3:

Instructions will be provided to the users on top of the forms to be filled.

UI-4:

There will be alerts and pop-ups which appear in case the user makes a mistake while using the application.

UI-5:

The interface will be responsive for all screen sizes as much as possible to provide the users with a seamless experience.

## **3.2 Hardware Interfaces**

N/A

## **3.3 Software Interfaces**

- Browsers: Mozilla Firefox 30+, and Google Chrome 27.0+ are the preferred browsers.
- Operating System: Android, Windows 7, 8, 10, Mac OS, Linux distributions.

## **3.4 Communication Interfaces**

The application will be using HTTPS protocol.

## **4. System Features**

### **4.1 Authentication and Authorization**

#### **4.1.1 Description and Priority:**

The application will have multiple users and so authentication becomes a high-priority system feature. The application will be using the JWT authentication module in order to implement this functionality. When the user creates a new account on the application, they will have to provide their email address and password. The password must be at least 8 characters long and must have at least one uppercase character, one digit and one special character. The passwords in the system will be hashed and stored so that no other person can get to know the password.

#### **4.1.2 Response Sequences:**

Once the user registers in the application, they will be guided to a Home page where they can start using the application. There will also be a logout button on the navigation bar. On clicking the logout button, the user will be logged out.

#### **4.1.3 Functional Requirements:**

REQ-1: We use the JWT authentication module for authentication and authorization functionality. The authentication will be session-based authentication.

### **4.2 Add Nursery Feature**

#### **4.2.1 Description and Priority:**

Features to add nurseries on the website will be provided to the user (owner in this case). The owner will have to provide all specifications about the nursery which he intends to put on the lease. The feature is of high priority as the application is based on owners adding nurseries on the website to get more potential buyers.

#### 4.2.2 Response Sequences:

The user will have to provide all the necessary information about the nursery. On filling out all the information the user will be redirected to a page where he will have the option to add photographs of the nursery. If there are any errors in the description of the nursery, they will be pointed out using alerts to notify the user.

#### 4.2.3 Functional Requirements:

REQ-1: The information provided will be validated using JavaScript on the client side of the application.

### **4.3 Add Photographs Feature:**

#### 4.3.1 Description and Priority:

The user gets to add photographs of the nursery products which he intends to put on the lease. This is again a high-priority feature. The user gets the option to choose images using the file system or simply drag and drop the pictures into the provided area.



#### 4.3.2 Response Sequences:

The user can add images using the drag-and-drop functionality or simply choose images from the files that he wants to upload. In case any image exceeds the maximum file size the user will be notified using alerts. On adding the images, the user is redirected to the add location page.

#### 4.3.3 Functional Requirements:

REQ-1: For the drag and drop functionality we use the Dropzone JavaScript library.

### **4.4 Add Product Feature**

#### 4.4.1 Description and Priority:

The user gets to add the product of the nursery. The user will simply have to add details of the nursery product in order to let the user know more about their product if they want to sell it faster. The priority of the feature is moderate.

#### 4.4.2 Response Sequences:

On adding the product, the nursery product is successfully added to the database and the user is redirected to the Add new product page of the website.

#### 4.4.3 Functional Requirements:

REQ-1: The information provided will be validated using JavaScript on the client side of the application.

## **4.5 Search Nurseries Feature:**

### **4.5.1 Description and Priority:**

The user can search different nurseries using the search functionality. This feature is of high priority and is primarily for experienced users. The user can search based on his requirements such as desired location, area, type etc.

### **4.5.2 Response Sequence:**

On filling in the required information based on desired features, the results will be displayed on the webpage.

### **4.5.3 Functional Requirements:**

REQ-1: The information provided will be validated using JavaScript on the client side of the application.

## **4.6 Shortlisting Nurseries Feature**

### **4.6.1 Description and Priority:**

This feature will allow users to shortlist or save nursery products that they are interested in, for future reference. This is a moderate priority feature.

### **4.6.2 Response Sequence:**

Once the user clicks on the shortlist button the nursery product will be added to the list of nursery products that have been shortlisted by that particular user.

#### 4.6.3 Functional Requirements:

REQ-1: For shortlisting nurseries, AJAX calls will be used. Therefore, JQuery will be used on the client side.

### **4.7 Delete Nurseries Feature**

#### 4.7.1 Description and Priority:

This feature will allow the owner of the nursery to remove the nursery from the system. This is a moderate priority feature.

#### 4.7.2 Response Sequence:

Once the user clicks on the delete nursery button, a popup will appear asking the user if he is sure if he wants to delete the nursery. If the user responds if Yes, then the nursery will be removed from the system.

#### 4.7.3 Functional Requirements:

REQ-1: For deleting nurseries, AJAX calls will be used. Therefore, JQuery will be used on the client side.

### **4.8 Deleting Shortlisted Nursery Product Feature**

#### 4.8.1 Description and Priority:

This feature will allow users to remove nursery products shortlisted by them. This feature is of moderate priority.

#### 4.8.2 Response Sequence:

Once the user clicks on the remove button the nursery product will be removed from the list that has been shortlisted by that particular user.

#### 4.8.3 Functional Requirements:

REQ-1: For removing shortlisting nurseries, AJAX calls will be used.

Therefore, JQuery will be used on the client side.

## **5. Other Nonfunctional Requirements**

### **5.1 Performance Requirements**

#### **5.1.1 Scalability:**

The application should be scalable and should perform without any interruption for all users.

### **5.2 Safety Requirements**

- A backup power supply should be present for the server so that it does not stop functioning in case of power failure.
- API keys of the APIs used should not be made open source.
- The credentials of the Database connection need to be hidden in the env file and made to be ignored.
- Code backup should be taken at regular time intervals.

### **5.3 Security Requirements**

- The passwords of the users are hashed and then stored in the database so that no person can access the passwords of the users.
- The passwords should be at least 8 characters long and must have at least one uppercase character, one digit and at least one special symbol.
- The website should use the HTTPS protocol for security.

- POST requests are used for transferring information regarding authentication, adding nurseries, adding advertisements etc. through forms.

## **5.4 Software Quality Attributes**

### **5.4.1 Usability:**

The user interface should be simple to use and not cluttered with a lot of information.

### **5.4.2 Availability:**

- The system should be available at all times.
- The system should be reliable and there should be no loss of data in case the server breaks down when operations are going on.

### **5.4.3 Maintainability:**

The code for the application should be written cleanly and should be well documented. The code should contain comments to help new programmers and developers make changes in the application.

### **5.4.4 Testability:**

The code should be written with proper test cases to be tested upon so that no errors during production take place.

## **5.5 Business Rules**

The administrator of the application has full permission to control the system.

## **6. Other Requirements**

### **Appendix A: Glossary**

- HTTPS: Hypertext Transfer Protocol Secure
- API: Application Programming Interface
- GUI: Graphical User Interface

### **Appendix B: Analysis Models**

#### **ER Diagram of the Application**

#### **Use Case Diagram for the application**