Software Requirements Specification

for

Lawn Depot

Version 1.0 approved

Prepared by Charles Ueltschey, Hunter Pope (Team Leader), Kanyan Gressett, Matthew Hoang, and Stephen Martin

Group 5

Jan 22, 2024

1. Introduction

1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

The purpose of this document is to detail the creation and structure of a lawn care ecommerce website. It will explain the interface, purpose, and structure of the website. It will delve into the details of the system, what requirements it needs to fulfill, and the constraints under which it will operate. This document is needed for both the client and the developers, to outline the project as a whole. It will be proposed to our lab TA for approval.

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

All fonts are Times New Roman 12pt. Each subsection is Times New Roman 13pt. Headers are bolded. Each section is divided by numbers <Section>.<subsection>.<description number>.

1.3 Intended Audience and Reading

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

The intended audience and readers for this document primarily consists of developers, testers, and instructors. The developers and testers consist of our team members, and the instructors consist of the Professor and the lab TA. The rest of this SRS contains information that helps further explain our website. Each section in this document delves into a different aspect of the project. This document begins with the introduction which gives a snippet of information about the website. The introduction is split into different sections highlighting various topics. Following the introduction is the Overall Description section. This section goes into more specific requirements than the introduction, but is still just an overview of the project. The System Features section is third. This part of the SRS describes the majority of the features that will allow our website to operate and thrive. It will detail the different classes that will make up our project. The last two sections in our SRS are the Other Nonfunctional Requirements and the Other Requirements. These two sections are

primarily centered around requirements that were not included earlier in the document. The nonfunctional requirements will contain less important information. Then the last section is only for the extra requirements that are not specified anywhere else.

1.4 Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

The website will be a simple ecommerce website which allows for browsing, purchasing, and publishing reviews on various yard decorations. It will be designed to facilitate ease of use for consumers who want to purchase them. The website will provide a smooth navigating experience for the customer, and a space for niche lawn decoration lovers to find their dream items. More specifically, the website will be open to view without subscription, but will require an account to purchase products and leave a review. It will allow for browsing, reviewing, and ordering any products on the system. The system will hold pre-made image urls, prices, vendors, and names in a database.

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

Lawn Depot is an e-commerce website specializing in lawn decorations and hanging ornaments. This website will allow the sale of any number of lawn items, separated into categories

and filtered by price, name and other characteristics. The website is based on a popular lawn decor ecommerce site “happy gardens”. The usage of this website will be limited to the computer science department at MSU and all team members involved. Lawn Depot is a project designed to fulfill the credits for Intro Software Engineering course at Mississippi State University. The website will imitate the purchasing and shipping of products, but will not actually charge users any money.

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

- A user will be able to login for purchasing
- A user can register a new account and link a credit card for purchases
- An admin can customize the website and database with new data
- Users can purchase products from the website, assuming they are logged in
- Users can view their order information and cancel if necessary
- Users can browse by category
- Users who have purchased the product can post a review
- Users can cancel previous orders
- Users will be able to view and edit their cart

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

1) Users - user must be logged in to purchase, but can browse without an account
    a) new - just registered (no purchases recorded)
    b) returning - has a record of previous orders, allows user to cancel and view orders
2) Admin - can add or remove products, change quantity etc.

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

The system will operate on a Ubuntu 23 operating system with minimum 25 GB hard disk space, 2 GHz quad-processor, and 2GB RAM.

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

1) One constraint that affects our product is that we are running the site on Ubuntu with limited storage space, so the number of products is constricted.
2) Another constraint is the use of SQLite databases, which can be difficult to scale and manage properly when not in the cloud. Additionally, querying SQL databases comes with vulnerabilities to SQL injection, therefore databases such as products, which are open to any queries, should not hold sensitive data.

3. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

3.1 Login

3.1.1 Description and Priority

Summary: The login feature allows the user of the software to access the website as a registered user.

Priority: High

Precondition: The user must register an account.

Triggers: The user selects the sign in screen or the user attempts to purchase a product while not logged in.

3.1.2 Stimulus/Response Sequences

- When a user is trying to check out, they will be prompted to login, if they are not already.

- The user will enter their username and password
- The user clicks the submit button
- Access to the website will be granted if correct, or the user will be prompted to re-enter information

3.1.3 Functional Requirements

- REQ 1: User Interface, working API in javascript
- REQ 2: Query Authentication Database
- REQ 3: Send Response with Cookie on success

Use Case

User	**System**
The user selects the sign in screen.	The website page shows username and password.
The user completes the fields.	The website validates the user as a registered user.

3.2 Register Account

3.2.1 Description and Priority

Summary: The register feature allows the user of the software to create an account.

Priority: High

Precondition: The user must be on the website.

Triggers: The user selects the sign in screen and clicks text under password.

3.2.2 Stimulus/Response Sequences

- The user will click the register account button
- The user will enter account information and payment
- The user will hit submit
- The user will be automatically signed in

3.2.3 Functional Requirements

- REQ 1: User Interface, working API in javascript
- REQ 2: Write data to authentication database
- REQ 3: Send response with cookie on success

Use-Case

User	**System**
The user selects the sign in screen.	The website page shows username and password.
The user clicks small text that asks user to register an account	The website loads a page for guests to register.
The user completes the registration page.	The website validates the user as a registered user.

3.3 Search Items

3.3.1 Description and Priority

Summary: The login feature allows the user of the software to access the website as a registered user. This feature's priority is medium.

Priority: Medium

Precondition: The user must be on the website.

Triggers: The user selects the search bar and types in a suggestion.

3.3.2 Stimulus/Response Sequences

- User will click on the search bar
- User will type the name of the item they need
- User will toggle any details they specifically want for their product
- User will be able to see price, details, reviews on product
- User will be able to add item to cart

3.3.3 Functional Requirements

- REQ 1: Search bar and results page (similar to products)
- REQ 2: Backend logic to efficiently search database for similar words

Use-Case

User	**System**
The user selects the search bar.	The website waits for user input.

The user types in the search bar.	The website shows suggestions based on the user's input.
The user hits enter.	The website shows all final results that match the user's input.

3.4 Create Review

3.4.1 Description and Priority

Summary: The login feature allows a logged in user to create a review for a product.

Priority: Low

Precondition: The user must be logged in and had previously bought the item.

Triggers: The user clicks on the item they bought and then clicks on the “review item” text.

3.4.2 Stimulus/Response Sequences

- The user purchases an item from the store
- The user clicks the review button and is verified if they have purchased the item
- The user enters rating and review (optional)
- The user clicks submit to post the review

3.4.3 Functional Requirements

- REQ 1: Functional subsection for reviews on the frontend
- REQ 2: API endpoint to check if user has ordered item
- REQ 3: Review database written to

Use-Case

User	**System**
The user selects the item they had purchased.	The website loads the item with additional text as “review item”.
The user selects the “review item” text.	The website loads a new page which allows the user to add pictures and text.
The user fills out the form and completes the review.	The website validates the review and publishes it alongside the item.

3.5 Add to Cart

3.5.1 Description and Priority

Summary: The login feature allows the user of the software to add a product to the user's cart.

Priority: Medium

Precondition: The user must be on the website.

Triggers: The user selects the "add to cart" option listed next to the item's price.

3.5.2 Stimulus/Responses Sequences

- User clicks add to cart on an item
- The item is then added to the users cart

3.5.3 Functional Requirements

- REQ 1: Cart page frontend to display cart items
- REQ 2: API endpoint for adding a new item to cart
- REQ 3: API endpoint to get items in cart
- REQ 4: Error checking if Quantity is 0

Use-Case

User	**System**
The user selects add to cart.	The system sends the product into the user's cart.

3.6 View Shopping Cart

3.6.1 Description and Priority

Summary: This feature allows a user or guest to view their shopping cart with the items they've added and the total price.

Priority: Medium

Precondition: The user must be on the website.

Triggers: The user selects the user icon in the top right of the webpage and clicks the view cart option.

3.6.2 Stimulus/Response Sequences

- User clicks cart section
- If not logged in they will be prompted to
- Cart can be viewed and changed
- User can edit item quantity

3.6.3 Functional Requirements

- REQ 1: Frontend to allow for cart viewing and manipulation
- REQ 2: API to edit cart quantity
- REQ 3: backend logic to ensure Quantity is sufficient

Use-Case

User	**System**
The user selects the user icon.	The website loads several options in a small box.
The user selects the view cart option.	The website loads a page containing the user's cart.

3.7 Remove from Shopping Cart

3.7.1 Description and Priority

Summary: This feature allows a user to remove items from their shopping cart.

Priority: Medium

Precondition: The user must have items in their shopping cart.

Triggers: The user selects the view shopping cart option and removes the item they want.

3.7.2 Stimulus/Response Sequences

- The user wishes to remove an item from their cart, so they click a “Remove From Cart button” while viewing their personal cart. This removes the selected item from the user's cart.
- User clicks cart section
- If not logged in they will be prompted to
- The user’s cart will then be displayed
- The user can then click on the remove from cart button that will be on every item in the cart

3.7.3 Functional Requirements

- REQ 1: Frontend to allow for cart viewing and manipulation
- REQ 2: API call to remove selected item from cart

Use-Case

User	**System**
The user selects the view shopping cart option.	The website loads a page of the users current shopping cart.
The user chooses the remove item option next to the item they want to delete.	The website gives the user a warning that the selected item is about to delete.
The user accepts their choice.	The item is deleted from their shopping cart. The page and the total price is updated.

3.8 Proceed to Checkout

3.8.1 Description and Priority

Summary: The login feature allows the user of the software to access the website as a registered user.

Priority: High

Precondition: The user must create an account

Triggers: The user selects the sign in screen or the user attempts to purchase a product while not logged in.

3.8.2 Stimulus/Response Sequences

- User will be told the amount they need to pay
- User will be asked their card information if not already linked to their account

3.8.3 Functional Requirements

- REQ 1: Frontend for processing payment and giving order details
- REQ 2: Backend to make order and decrement item quantity

Use-Case

User	**System**
The user selects the sign in screen.	The website page shows username and password.
The user completes the fields.	The website validates the user as a registered user.

3.9 Logout

3.9.1 Description and Priority

Summary: The logout feature allows the user of the software to logout of the current sessions account.

Priority: Medium

Precondition: The user must be logged into an account.

Triggers: The user selects the user icon in the top right of the webpage and clicks the logout option.

3.9.2 Stimulus/Response Sequences

- When a logged in user is ready to logout, they can select the top right icon and select logout.
- The user will be prompted an “Are you sure?” query.
- The user selects the Yes button.
- The website will unlink the user’s information for the current session.
- The system will invalidate the cookies attached to the account.

3.9.3 Functional Requirements

- REQ 1: User Interface, working API in javascript
- REQ 2: Backend logic to invalidate cookies.

Use-Case

User	**System**
The user selects the user icon.	The website loads several options in a small box.
The user selects the logout option.	The system unlinks the user from the previously signed in account.

3.9 Remove Account

3.9.1 Description and Priority

Summary: This feature removes the current user's account and deletes all data associated with their account.

Priority: Medium

Precondition: The user must be logged into an account.

Triggers: The user selects the user icon in the top right of the webpage and clicks the Delete My Account koption.

3.9.2 Stimulus/Response Sequences

- When a logged in user wants to delete their account, they click the icon on the top right of the screen which brings up many options, one being to delete their account.
- The user will select “Delete my Account”.
- The webpage will show a screen saying “Are you sure”.
- The user will confirm.
- The website will ask the user to log in again to confirm the deletion of the account.
- All data associated with the user will be deleted.

3.9.3 Functional Requirements

- REQ 1: Delete Account Button
- REQ 2: Are you sure page and re-enter credentials page
- REQ 3: API endpoint (secured) that deletes accounts

Use-Case

User	**System**
The user selects the user icon.	The website loads several options in a small box.
the user selects the delete account option.	The system asks the user to confirm their choice.
The user confirms.	The system asks the user to log in again for deletion.
The user types in their credentials.	The system logs the user out and deletes all data associated with the user.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

Our website has several performance requirements that we wish to implement. The first is we want there to be no lag when switching between pages or at least sub 10 millisecond load times. All of the navigation throughout our website should be smooth and consistent. Images should load seamlessly and the page should remain visually stable while loading. This is to ensure that the user experience is good, and that the use of our website is smooth, thereby promoting more purchases.

4.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

Some action should be made to prevent the posting of vulgar or harmful reviews. A system should be made to parse the reviews for bad language and censor them accordingly. Additionally only those who have ordered an item should be allowed to review, thereby preventing spam. There should be a terms page that will describe the proper use of the website, and what classifies as vulgar language.

4.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

Our website must be secure against a variety of attacks, including cross site scripting and SQL injection. We should implement a query parser and a WAF to ensure that the url and database queries cannot be used to obtain excess information. Most importantly all sensitive data will be held in a separate database, with a more secured and limited access to the website, only being accessed

when logging in and placing an order. All queries related to the sensitive database will be encrypted, parsed by a WAF and checked for SQL injection risks.

4.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

Our product will be designed in such a way that it will be easily adaptable. Each service will be compartmentalized to facilitate lower coupling. Login page, homepage, and cart will be separate web pages to allow for a less coupled system. Additionally each API endpoint will be responsible for one task only. All these characteristics will make testing and maintenance more manageable.

5. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

A time requirement is enforced by the constraint of this project being completed in a course. The course is only a semester long, so this product must be completed within that time frame. Also, the project is broken up into sprints that are to be completed every couple of weeks.

We are using SQLite, so it is required that we use python in order to get the most use out of the database system. Python has a larger library for SQLite than most other coding languages.

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

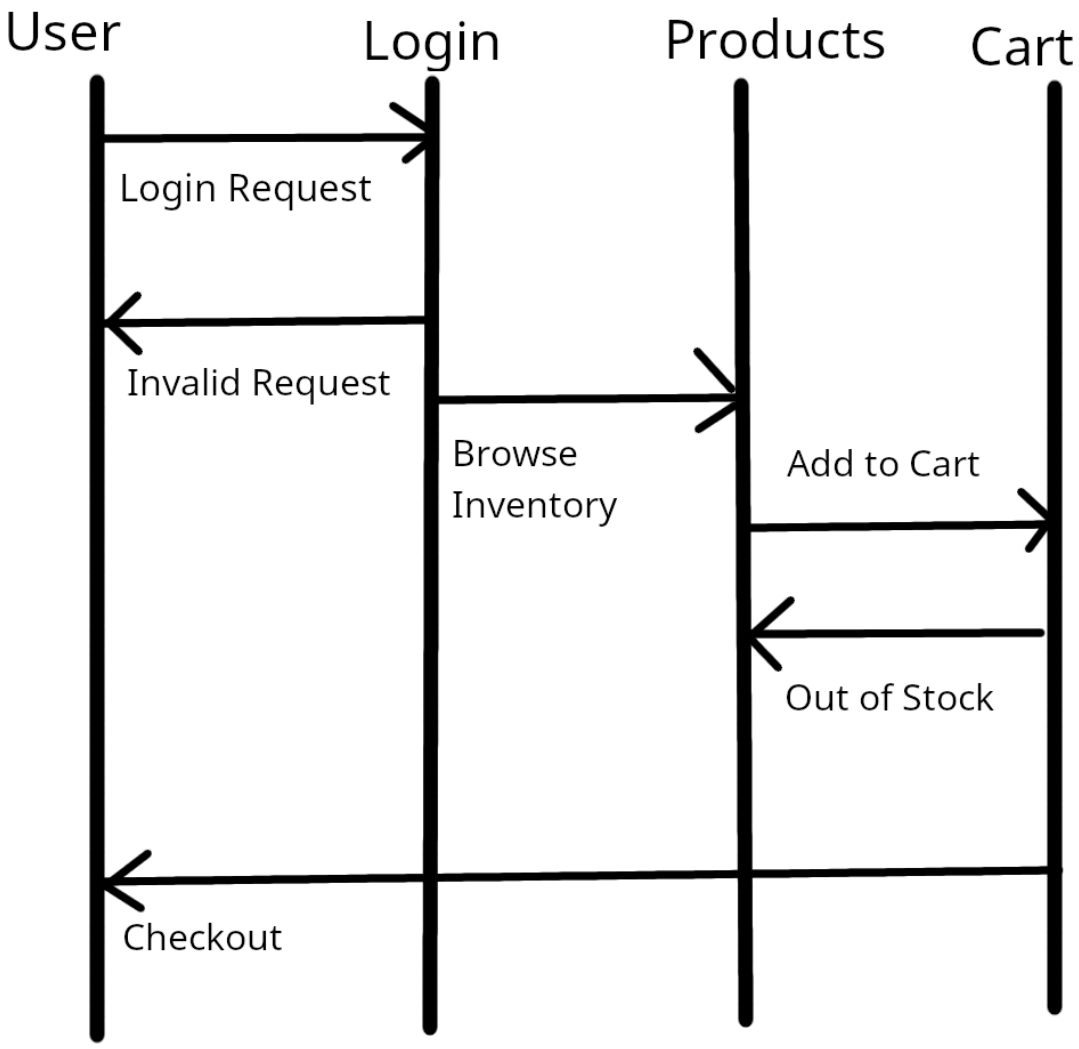

Figure 1. Behavior Diagram

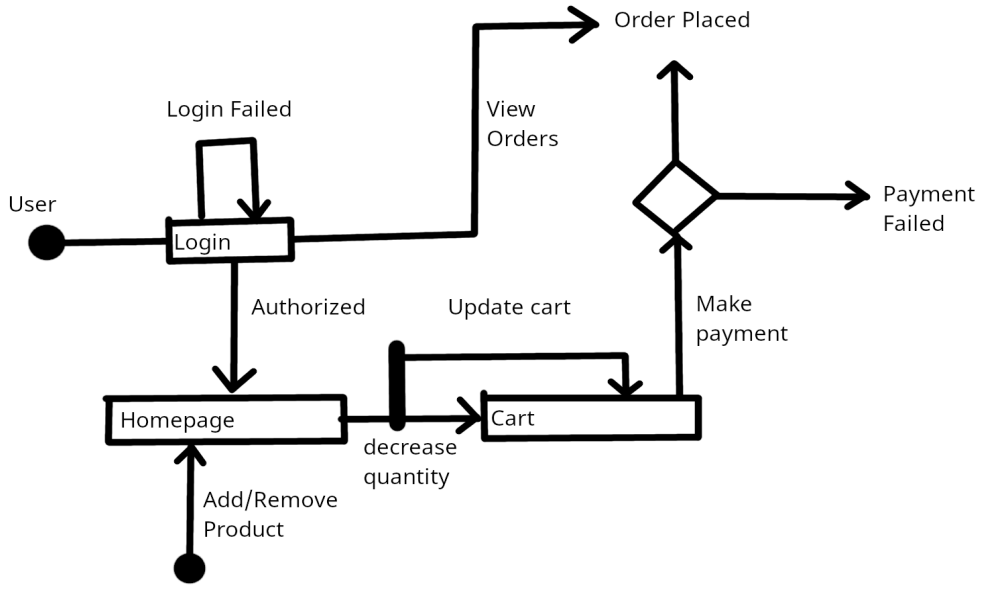

Figure 2. Dynamic Model

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

TBD: make a reset password feature using email
TBD: Allow for a seller type user that can post their own products (A lot to figure out here)