

---

# SOFTWARE REQUIREMENTS SPECIFICATION

for

BurnLab

Version 1.0

Prepared by : Noah McCord, Alexander Jones,  
Maddox Hodges, Azizbek Negmatov

CSE-Intro to SE

September 11, 2025

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Purpose . . . . .	4
1.2	Document Conventions . . . . .	4
1.3	Intended Audience and Reading . . . . .	4
1.4	Product Scope . . . . .	5
1.5	References . . . . .	5
<b>2</b>	<b>Overall Description</b>	<b>6</b>
2.1	Product Perspective . . . . .	6
2.2	Product Functions . . . . .	6
2.3	User Classes and Characteristics . . . . .	6
2.4	Operating Environment . . . . .	7
2.5	Design and Implementation Constraints . . . . .	7
<b>3</b>	<b>System Features</b>	<b>8</b>
3.1	Profile Registry . . . . .	8
3.1.1	Description and priority . . . . .	8
3.1.2	Stimulus/Response Sequences . . . . .	8
3.1.3	Functional Requirements . . . . .	9
3.2	Login and Logout System . . . . .	9
3.2.1	Description and priority . . . . .	9
3.2.2	Stimulus/Response Sequences . . . . .	10
3.2.3	Functional Requirements . . . . .	11
3.3	Marketplace . . . . .	11
3.3.1	Description and priority . . . . .	11
3.3.2	Stimulus/Response Sequences . . . . .	11
3.3.3	Functional Requirements . . . . .	12
3.4	Order History . . . . .	12
3.4.1	Description and priority . . . . .	12
3.4.2	Stimulus/Response Sequences . . . . .	13
3.4.3	Functional Requirements . . . . .	13
3.5	Shopping Cart . . . . .	13
3.5.1	Description and priority . . . . .	13
3.5.2	Stimulus/Response Sequences . . . . .	14
3.5.3	User Makes Purchase . . . . .	14
3.5.4	Functional Requirements . . . . .	14

3.6	Homepage . . . . .	14
3.6.1	Description and priority . . . . .	14
3.6.2	Stimulus/Response Sequences . . . . .	15
3.6.3	Functional Requirements . . . . .	15
3.7	Search Toolbar . . . . .	15
3.7.1	Description and priority . . . . .	15
3.7.2	Stimulus/Response Sequences . . . . .	15
<b>4</b>	<b>Other Nonfunctional Requirements</b>	<b>17</b>
4.1	Performance Requirements . . . . .	17
4.2	Safety Requirements . . . . .	17
4.3	Security Requirements . . . . .	17
4.4	Software Quality Attributes . . . . .	18
4.4.1	Functional Requirements . . . . .	18
<b>5</b>	<b>Other Nonfunctional Requirements</b>	<b>19</b>
5.1	Performance Requirements . . . . .	19
5.2	Safety Requirements . . . . .	19
5.3	Security Requirements . . . . .	19
5.4	Software Quality Attributes . . . . .	20
<b>6</b>	<b>Other Requirements</b>	<b>21</b>

# 1 Introduction

## 1.1 Purpose

The software requirements specification is intended to provide a detailed explanation of the features and design specifications of this website. This document is intended for the stakeholders and developers of this website.

## 1.2 Document Conventions

This document follows a consistent structure and formatting style to make the requirements clear and easy to reference. Section headings are numbered according to IEEE SRS standards, with subsections used to break down specific requirements.

Plain text is used for general descriptions. **Bold text** is used for emphasis on key terms or priorities.

All functional requirements are labeled with the prefix **REQ-** followed by a number (e.g., REQ-1, REQ-2) to provide a unique identifier for traceability. High-level priorities defined in system features are inherited by their detailed requirements unless otherwise stated.

Fonts are standardized throughout the document for consistency, and requirements are written in complete sentences to avoid ambiguity.

## 1.3 Intended Audience and Reading

This SRS is designed to serve as an easy-to-understand framework for all stakeholders involved with the BurnLab platform. It provides developers with a clear set of functional and nonfunctional requirements to guide implementation. Project managers can use this document to track progress and verify that all planned features align with business goals. Marketing staff may refer to it to understand the scope of features and the value the platform provides to end users. Testers will use the requirements to create test cases and verify system functionality, while administrators and support staff can reference it for understanding expected system behaviors.

The document begins with an introduction and overall description of the product, followed by detailed system features and functional requirements. These sections are most useful to developers, testers, and managers who need a precise technical understanding. The nonfunctional requirements and other requirements sections provide guidance on performance, safety, security, and quality attributes, which are of interest to both technical staff and project stakeholders.

A recommended sequence for reading is to start with the Introduction (Section 1) to understand the purpose and scope, proceed to the Overall Description (Section 2) for context, then review the System Features (Section 3) for detailed requirements. Readers concerned with long-term operation, maintainability, and compliance should pay special attention to Other Nonfunctional Requirements (Section 4) and Other Requirements (Section 5).

## **1.4 Product Scope**

BurnLab would provide the meal and workout plans for customers to maintain a healthy lifestyle. The plan should be adaptable based on the preferences and characteristics of the users.

## **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 vision and scope documents. Provide enough information for the reader to access a copy of each reference, including title, author, version number, date, and source or location.

## **2 Overall Description**

### **2.1 Product Perspective**

BurnLab is a self-contained website. BurnLab will be an intermediary between sellers and buyers of products. The system of this web application will use methods of interconnected relational databases. Our users will have simple-to-understand interfaces so that anyone can use the website without confusion or a lack of technical skills.

### **2.2 Product Functions**

BurnLab provides users with the ability to create personalized fitness and nutrition experiences. The platform will collect basic profile information such as age, weight, height, and fitness goals, then use this data to generate customized workout and meal plans. Users will be able to log their daily progress, track performance over time, and receive updated recommendations as they progress in their fitness journey. The system will also function as a marketplace, where sellers such as trainers and nutritionists can upload their own programs, recipes, and plans for purchase. Buyers will be able to browse available plans, compare options, and make secure purchases through the platform. Administrative functions will include user management, content moderation, and maintaining platform security. In addition, BurnLab will allow guest visitors to preview limited content to encourage registration. These core functions: personalized plan generation, progress tracking, content marketplace, and administrative oversight will define the essential services BurnLab must deliver.

### **2.3 User Classes and Characteristics**

Buyers, Sellers, Admins, are the base user class which are going to be instated initially into the software. Buyers will have an email and password requirement to enter into the webstore's homepage. They will have access to the various meal plans and/or fitness plans sold by different sellers. In addition, sellers will have the ability to add and remove different products from our Webstore. Lastly, the Admin class is a general superuser with the ability to make direct contact with users and sellers, settling disputes and fixing bugs or errors in the buyer and seller experiences.

## **2.4 Operating Environment**

BurnLab will be optimized for general web applications. Such as Chrome, Firefox, Edge, etc.

## **2.5 Design and Implementation Constraints**

Burnlab is going to be using small scale servers, and as such, it will not be able to support a large number of users, as Amazon or eBay may be able to support. Burnlab will use Django, HTML, and CSS as the main languages for this project. As for the databases, we are using Firebase to serve as the cloud-hosted database. HTTPS and TLS encryption will be used to protect user data. Using hashing to protect data from bad actors and data leaks. Lastly, sellers must provide allergen data to alert any potential buyers of items to avoid.

## 3 System Features

### 3.1 Profile Registry

#### 3.1.1 Description and priority

Allow a user to create an account using an email and password. This is high priority since it is required for any user interaction on the website.

#### 3.1.2 Stimulus/Response Sequences

##### **Profile as Admin**

- Admin navigates to “Profile” in the Admin Console.
- System loads stored details.
- Admin edits fields and saves changes.
- System validates and updates profile.
- System displays: “Profile updated successfully.”

##### **Profile Request as Seller**

- User navigates to “Apply as Seller.”
- User enters business name, address, and tax ID.
- User uploads verification documents.
- System validates required info.
- Application stored as “Pending Review.”
- Notification: “Your seller request is under review.”

##### **Seller Profile Denied – False Information**

- Admin reviews application.
- Detects false or misleading info.



- Clicks “Deny” and records reason.
- System updates status to “Denied.”
- Seller notified: “Your application was denied due to false information.”

#### **Seller Profile Denied – Business Does Not Exist**

- Admin reviews application.
- Cannot verify business existence.
- Clicks “Deny” with reason “Business not found.”
- System updates status to “Denied.”
- Seller notified: “Your application was denied because the business could not be verified.”

#### **Profile as Buyer**

- Buyer navigates to “My Profile.”
- System loads stored information.
- Buyer edits personal details.
- Buyer clicks “Save.”
- System validates and updates record.
- Confirmation: “Profile updated.”

### **3.1.3 Functional Requirements**

REQ-1: Given that both the email and the password meet required standards the system will accept a new user.

REQ-2: If there is an invalid email or password input send a message to the user.

REQ-3: Must have User, Seller, or Admin specifications.

## **3.2 Login and Logout System**

### **3.2.1 Description and priority**

The login system feature will validate the user input credentials. This will be a high priority as it will be necessary for users and sellers to get access to the main functions.

### 3.2.2 Stimulus/Response Sequences

#### Login as Seller

- Seller navigates to the BurnLab login page.
- Seller selects “Login as Seller.”
- Seller enters email and password.
- System validates the credentials.
- If valid → redirect to Seller Dashboard.
- If invalid → display: “Invalid login credentials.”

#### Login as Buyer

- Buyer navigates to the BurnLab login page.
- Buyer selects “Login as Buyer.”
- Buyer enters email and password.
- System verifies credentials.
- If valid → redirect to Homepage.
- If invalid → display: “Email or password not recognized.”

#### Login as Admin

- Admin navigates to the BurnLab login page.
- Admin selects “Login as Admin.”
- Admin enters email/username and password.
- System validates against Admin database.
- If valid → redirect to Admin Console.
- If invalid → display: “Login failed.”
- If non-admin attempts → block and redirect to Homepage.

#### Logout

- Any logged-in user (Buyer, Seller, or Admin) clicks “Logout.”
- System ends the session and clears tokens.
- User is redirected to the login page.
- If inactive for 15 minutes → system automatically logs out and displays: “Session expired.”

### 3.2.3 Functional Requirements

REQ-4: If the user inputs a valid email-password combination, they will be redirected to the homepage.

REQ-5: If the email or password is incorrect the user will be sent an error message.

## 3.3 Marketplace

### 3.3.1 Description and priority

The Marketplace is where users will view available products and add them to their cart for purchase; as such it is a high priority inclusion.

### 3.3.2 Stimulus/Response Sequences

Seller Requests to Put Product on the Market

- Seller navigates to “Add Product.”
- Enters title, description, price, and stock.
- Uploads images.
- Submits product.
- System validates input and sets status to “Pending Admin Approval.”
- Seller notified: “Your product has been submitted for review.”

Admin Approves Product

- Admin reviews pending products.
- If compliant → clicks “Approve.”
- System changes status to “Active.”
- Product published to marketplace.
- Seller notified: “Your product has been approved.”

Admin Denies Product

- Admin reviews pending product.
- Finds violation or issue.

- Clicks “Deny” and enters reason.
- System updates status to “Denied.”
- Seller notified: “Your product was denied. Reason: [admin note].”

#### Marketplace Quantity Updates

- Buyer views product listing.
- System displays stock quantity.
- Buyer adds item to cart.
- System reduces available quantity.
- If stock hits zero → mark as “Sold Out.”

#### Navigate to Marketplace Page

- User clicks “Marketplace” in navigation.
- System loads Marketplace page.
- User browses items with pagination and filters.
- User can return to Homepage at any time.

### 3.3.3 Functional Requirements

REQ-6: User can add an item to the cart from the marketplace.

REQ-7: User can move between pages of the marketplace to view more stock.

REQ-8: User can move to the shopping cart from the marketplace.

REQ-9: User can move back to the homepage from the marketplace.

## 3.4 Order History

### 3.4.1 Description and priority

Order History will include the information on all purchases and sales made through Burnlab; as such it will be listed as a high priority inclusion because it is the way any disputes will need to be handled.

### **3.4.2 Stimulus/Response Sequences**

#### **Buyer looking for receipt documentation**

- Buyer clicks on the orderID among the list of orders.
- Buyer sees a pdf for the receipt as well as an invoice.
- Buyer selects to open the receipt pdf.
- Buyer searches using the search toolbar for another product he is interested in.

#### **Seller needs to view order invoice**

- Seller clicks to a previous page until he sees the first new order of the day
- Seller finds among the listed order history multiple purchases so the seller goes through the orderIDs for processing.
- Seller opens the invoice and receipt documents for each to begin those order confirmations and their preparation.
- Seller clicks to return to the homepage once they have finished their work and leaves their computer on by accident
- Seller is automatically logged out after a period of inactivity.

### **3.4.3 Functional Requirements**

REQ-10: User can move to the order history page from the homepage.

REQ-11: User can move back to the homepage from the order history page.

REQ-12: User can see the OrderID and can click to view the invoice and receipt files.

## **3.5 Shopping Cart**

### **3.5.1 Description and priority**

Lists selected items with their unique IDs and total price. This will be listed as high priority because it is the only place buyers can remove items from their cart and as such will be needed for basic functionality.

### 3.5.2 Stimulus/Response Sequences

- User clicks to add the appropriate payment/shipping information into the appropriate boxes; now the user can type their information.
- User can click to pay receiving a unique orderID with the receipt and invoice sending the invoice also to the sellers.
- User clicks the homepage icon and goes to the homepage.

### 3.5.3 User Makes Purchase

- Buyer clicks button to purchase items in cart
- System displays price and prompts user to input card info and address
- If given invalid card info → send message to buyer asking for valid info
- If buyer tries to proceed without an address → system sends message stating an address is required
- If given valid card info and address → system does a final check if all items are in stock and card has enough money for the purchase.
- If card is short on money → system sends message declining card
- If any items are no longer in stock → buyer is informed that something went wrong
- Buyer is informed that "Your order has been placed"

### 3.5.4 Functional Requirements

REQ-13: The user would be able to check out and purchase all items in cart. They would have to input unique payment info to confirm the purchase.

REQ-14: The user would be able to remove any item from the cart.

REQ-15: The User may click back to the homepage from the shopping cart.

## 3.6 Homepage

### 3.6.1 Description and priority

The homepage is the hub to reach any page that may need to be reached, as well as it is the only page you can logout on. For that reason it is high priority.

### 3.6.2 Stimulus/Response Sequences

#### User Destination *itemize*

*Guest User clicks the login icon, going to the login page*

*User presses the marketplace icon to go to that page*

*User presses the order history icon to go to that page*

*User presses the shopping cart icon to go to that page*

*User searches for [generic item] and is taken to the item page*

#### **User Not Logged In**

- *Guest user tries to access Marketplace, error message "user must be logged in"*
- *Guest user tries to access Shopping Cart, error message "user must be logged in"*
- *Guest user tries to access Order History, error message "user must be logged in"*
- *Guest user tries to search in the search toolbar, error message "user must be logged in"*

### 3.6.3 Functional Requirements

*REQ-16: There must be an icon or some link to go to the following pages: Marketplace, Shopping Cart, Order History.*

*REQ-17: Must be able to logout of the system via an icon or link within the homepage.*

## 3.7 Search Toolbar

### 3.7.1 Description and priority

Search Toolbar isn't a necessary inclusion for the absolute functionality of the webstore, but it is something that we need to implement to make the user experience much better, especially for a vast store. So this is going to be listed as a Medium priority feature.

### 3.7.2 Stimulus/Response Sequences

#### Buyer looks for popular fitness plan

- *A new user clicks onto the search bar to search for a popular fitness plan they've heard about*
- *User types in [generic name] then clicks to start searching*
- *User finds the fitness plan and adds it to his cart.*

- *User purchases through the shopping cart and logs out.*

#### **Admin Updates Product**

- Admin receives a request to remove a product from the website from a seller.
- Admin searches an ItemID and is taken to the item page
- Admin removes ItemID from the Marketplace



## 4 Other Nonfunctional Requirements

### 4.1 Performance Requirements

- The website will load in less than 2 seconds.
- The website should be able to support up to 75 concurrent users at a time.
- The website should be able to handle 1000 registered users.
- The website should maintain a 99 percent up-time.

### 4.2 Safety Requirements

- BurnLab will display a disclaimer stating that it is not a substitute for medical or professional healthcare advice. Users are encouraged to consult a physician before starting any fitness or nutrition plan.
- An allergy and dietary restriction filter will be provided to help prevent accidental exposure to harmful ingredients.
- The system will implement validation checks to reduce errors in financial transactions and order processing, ensuring that users are not mistakenly charged.
- In case of system failures, users will be redirected to an error page with safe recovery options (e.g., retry or cancel actions) to prevent data corruption.

### 4.3 Security Requirements

- *All communications between client and server will use HTTPS with TLS encryption.*
- *User authentication will require strong passwords (minimum 8 characters, including numbers, special characters, and mixed case).*
- *Passwords will be stored using industry-standard hashing (e.g., bcrypt or SHA-256 with salt).*
- *The system will enforce automatic logout after a defined period of inactivity (e.g., 15 minutes).*

- *Sensitive data such as credit card numbers will never be stored in plain text and will comply with PCI DSS standards.*
- *Only authorized administrators will have access to seller and buyer data.*

## 4.4 Software Quality Attributes

**Usability:** *The platform will prioritize a clean and intuitive interface so new users can navigate with minimal instruction.*

**Reliability:** *BurnLab will aim for 99% uptime and include backup procedures to recover from unexpected crashes.*

**Maintainability:** *The system codebase will follow modular design and documentation standards so future developers can easily update or fix issues.*

**Scalability:** *The architecture will be designed to scale from supporting hundreds of users to thousands without significant degradation in performance.*

**Portability:** *The platform will be accessible on major browsers (Chrome, Firefox, Edge, Safari) and responsive on both desktop and mobile devices.*

**Testability:** *Features will be implemented with automated unit and integration tests to verify correctness.>*

### 4.4.1 Functional Requirements

*REQ-18: Search bar searches values using either of the following: ItemID, Nomenclature in marketplace.*

*REQ-19: Search bar searches values using the OrderID in the marketplace.*

## 5 Other Nonfunctional Requirements

### 5.1 Performance Requirements

- *The website will load in less than 2 seconds.*
- *The website should be able to support up to 75 concurrent users at a time.*
- *The website should be able to handle 1000 registered users.*
- *The website should maintain a 99 percent up-time.*

### 5.2 Safety Requirements

- *BurnLab will display a disclaimer stating that it is not a substitute for medical or professional healthcare advice. Users are encouraged to consult a physician before starting any fitness or nutrition plan.*
- *An allergy and dietary restriction filter will be provided to help prevent accidental exposure to harmful ingredients.*
- *The system will implement validation checks to reduce errors in financial transactions and order processing, ensuring that users are not mistakenly charged.*
- *In case of system failures, users will be redirected to an error page with safe recovery options (e.g., retry or cancel actions) to prevent data corruption.*

### 5.3 Security Requirements

- *All communications between client and server will use HTTPS with TLS encryption.*
- *User authentication will require strong passwords (minimum 8 characters, including numbers, special characters, and mixed case).*
- *Passwords will be stored using industry-standard hashing (e.g., bcrypt or SHA-256 with salt).*
- *The system will enforce automatic logout after a defined period of inactivity (e.g., 15 minutes).*

- *Sensitive data such as credit card numbers will never be stored in plain text and will comply with PCI DSS standards.*
- *Only authorized administrators will have access to seller and buyer data.*

## 5.4 Software Quality Attributes

**Usability:** *The platform will prioritize a clean and intuitive interface so new users can navigate with minimal instruction.*

**Reliability:** *BurnLab will aim for 99% uptime and include backup procedures to recover from unexpected crashes.*

**Maintainability:** *The system codebase will follow modular design and documentation standards so future developers can easily update or fix issues.*

**Scalability:** *The architecture will be designed to scale from supporting hundreds of users to thousands without significant degradation in performance.*

**Portability:** *The platform will be accessible on major browsers (Chrome, Firefox, Edge, Safari) and responsive on both desktop and mobile devices.*

**Testability:** *Features will be implemented with automated unit and integration tests to verify correctness.>*

## 6 Other Requirements

***Database Requirements:** The system will maintain user accounts, profiles, order histories, and product listings in a secure relational database* ***Legal Requirements:** The platform must comply with GDPR (for EU users), CCPA (for California users), and other relevant data privacy regulations.*