Software Requirements Specification

for

Fruhb

Version 1.0 approved

Prepared by

Ethan Heverly, Nicholas Jones, Shawn Butler, Tate Smith

Submitted for the requirements of

CSE 4214/6214 Intro to SE

1/23/2023

# Introduction

## Purpose

The purpose of this document is to describe our service, Fruhb. This SRS will explain the features of the service, the interfaces of the service, and the constraints underlying the service. This service is developed in accordance with the requirements provided by Reference [2]. This document discusses the systems and subsystems of Version 1.0. Thus, future versions of these systems will be released on updated SRS documents.

## Document Conventions

This SRS uses a tier section format, with major section titles written in bold and a font size of eighteen. The title for each section follows the format **‘W. Title’**, in which *W* represents the section number, and *Title* is the title of the section. Subsections follow a similar format - **‘W.X. Title.’** The *W* represents the section number, *X* represents the subsection number, and *Title* represents the title of the subsection. Smaller subsections may have the format **‘W.X.Y. Title’** or **‘W.X.Y.Z. Title’**, in which *W* represents the section number, *X* represents the subsection number, *Y* represents a subsection within a subsection, *Z* represents a subsection of the previous two subsections, and *Title* represents the title of the lowest priority subsection. All requirements inherit priority of the high-level requirements unless otherwise stated.

## Intended Audience and Reading

This document is intended for Charan Gudla, professor for class CSE 4214/6214 Intro to SE, teaching assistant Abhilash Kanduri, and anyone interested in learning more about the Fruhb service. When reading this SRS, the recommended starting point is ‘Section 2: Overall Description,’ which provides an overview of our service’s functions and constraints. ‘Section 3: System Features’ elaborates on the features present in the Fruhb, Version 1.0, service. For developers curious about specific system features, consulting Sections 3, 4, and 5 is recommended.

## Product Scope

Fruhb is an e-commerce store that will be run on a local host server. Fruhb is designed with simplicity in mind, where only the most important features will be used. Fruhb draws from grocery stores like Kroger to provide both a family-friendly platform and an easy user experience. Fruhb strives to keep user data safe and secure while logging in and out of our local host server. In this way, users can be reassured that their private information will not be exposed to hackers, and users can safely go and purchase their exotic fruits.

## References

1. Kanduri, A. Introduction to Software Spring 2024 Lab Guidelines, 6 Feb. 2024, [42146214\_LabGuideline-Spring-2024.pdf: CSE 4214/6214 Intro to SE Spring 2024 (msstate.edu)](https://canvas.msstate.edu/courses/121638/files/9987266?module_item_id=2640422)
2. Kanduri, A. Project Topic: An E-commerce Platform, 6 Feb. 2024, [Project Topic.pdf: CSE 4214/6214 Intro to SE Spring 2024 (msstate.edu)](https://canvas.msstate.edu/courses/121638/files/9824917?module_item_id=2535543)

# Overall Description

## Product Perspective

Fruhb is an e-commerce web application for exotic fruits. Its users include consumers and sellers, both of which have support. Fruhb is a new, self-contained product that should be able to run on any modern web browser, which means there are no hardware requirements and limited software requirements.

## Product Functions

* Account Management
  + A customer can be either a member or a non-member.
  + Each member has a validated email address and a password
    - Passwords have the following restrictions
      * At least eight characters in length
      * At least one uppercase letter
      * At least one lowercase letter
      * At least one number
      * At least one special character
  + Member information is stored in a secure database
    - Password hashes are stored
  + Members can edit saved payment and billing information.
* Catalog
  + Both available and unavailable products will be listed
  + Unavailable products will be properly labeled
  + An item’s popularity will be recorded and shown on the home page
  + A product will be reserved when in a customer's cart
  + Managers can update stock quantities
  + Sellers and managers get emailed updates on item quantities
* Cart Management and Order History
  + Fruits can be added to or removed from a cart
  + A cart item’s quantity can be changed
  + Transactions and their relevant data will be recorded
  + Guest Checkout
    - Collect a 16-digit credit card number
    - Collect billing information
    - Collect shipping information
  + Member Checkout
    - Option to use saved payment methods or collect payment information
    - Option to use saved billing information or collect billing information
    - Option to use saved shipping information or collect shipping information
    - Can view previously ordered items

## User Classes and Characteristics

* Consumer
  + Members - Can use all the features mentioned in section 3, excluding sections 3.1.9
  + Guests - Can use all the features mentioned in section 3, excluding sections 3.1.7
* Admin - Has access to the source code and databases stored in GitHub

The consumer class and its subclasses are essential in getting this application to run. The admin class is a catch-all class to avoid a seller class, developer class, product manager class, and a more appropriate admin class.

## Operating Environment

Fruhb will run as a local host server and its interface will be a web browser. The underlying operating system isn’t relevant since the backend will be made from Python and noSQL, which are supported on most operating systems.

## Design and Implementation Constraints

Language: Python with Flask, HTML, CSS, JavaScript

* Database: NoSQL, PyMongo
* Server: Local Machine
* Security:
  + Hashed user and password storage
  + Limited login attempts

Constraints:

* Must run on Google Chrome and Firefox 18
* User must authenticate their self at login

# System Features

## Detailed Feature Design List

### Priority Key

|  |  |
| --- | --- |
| **Priority** | **Description** |
| Mandatory (M) | Mandatory for the release. Cannot release without this. |
| Important (I) | Very important for the release. The system can technically function without this, but the project will greatly suffer if it is not released with this. |
| Wanted (W) | It would be a very good asset to the customer. It is preferable that this feature is also implemented but not required. |
| Future (F) | Future releases can have this feature implemented, but it is not a priority for the current build being worked on. |
| Dismissed (D) | Has been suggested but dismissed. This feature will never be implemented. |

### Top Bar

**Description:** The top bar will be displayed on most pages of the website and will have many links the user can access as they go about their activities on the website.

#### Req-1: User Button(M)

**Description:** A button is displayed that will link the user to the login page if the user is not logged in. If the user is logged in, then the button will be replaced with the user’s username.

**Interaction:**

* Case 1 (if the user is not logged in): The user will click the button and be taken to the login page.
* Case 2 (if the user is logged in): The user will see their username displayed where the login button was.

#### Req-2: Search Bar(M)

**Description:** A search bar that will allow the user to search through the catalog of fruits and vegetables by providing a key in the text box.

**Interaction:** There will be a text box where the user will enter the name of the item they are searching for and a button to click after the text is entered to initiate the search. Clicking the search button will direct the user to the search page with the list of relevant items corresponding to what the user typed in the search bar.

#### Req-3: Home Button: (M)

**Description:** There will always be a button on the screen that the user can click to go back to the home page.

**Interaction:** The user will click the button and be redirected to the home page.

#### Req-4: Settings Button(M)

**Description:** A button that takes the user to the settings page.

**Interaction:** When the button is clicked, the user will be redirected to the settings page.

#### Req-5: Cart Button (M)

**Description:** A button that takes the user to the cart page.

**Interaction:** When the user clicks the button, they are directed to the cart page.

### Home Page

#### Req-6: Featured Items (I)

**Description**: Display highlighted fruits and vegetables on the home page of the website and allow the user to add the item to their cart directly from there.

**Interaction**: The user will see the featured items on the front page with the price displayed next to them, as well as add the item name and a ‘+’ button. When the user clicks the ‘+’ button, the item will be added straight to the user’s cart.

#### Req-7: Search Fruits Button(W)

**Description:** A button will be displayed on the home page that will send the user to the Search Page, with fruits being what is searched for.

**Interaction:** The user clicks the button and gets sent to the Search page with all of the catalog’s fruits being displayed.

#### Req-8: Search Vegetables Button(W)

**Description:** A button will be displayed on the home page that will send the user to the Search Page with vegetables being searched for.

**Interaction:** The user clicks the button and gets sent to the Search page with all the catalog’s fruits being displayed.

### Cart Page

#### Req-9: Proceed to Payment Button(M)

**Description:** A button that takes the user to the payment information page, assuming the user has items in their cart.

**Interaction:** The user will click on the Proceed to Payment button and be linked to the payment information page.

**Error Handling:** If the user does not have any items in their cart, then this button will not be present. Instead, it will be replaced with a message instructing the user to add an item to their cart first.

#### Req-10: Remove Item from Cart Button(M)

**Description:** A button that will remove a specified item from their cart when the button is pressed. The button will not decrease the quantity of the item; it will just remove the item from the cart.

**Interaction:** The user will see this ‘X’ button next to the item in their cart and they will click on the button to remove the item from the cart.

#### Req-11: Increment and Decrement Item Quantity Buttons(M)

**Description:** There will be two buttons next to each other. One button will increase the quantity of an item that they have in their cart, and the other button will decrease the quantity of that item. If the item quantity reaches 0, then the item will be removed from the user’s cart.

**Interaction:** The user will see a ‘+’ and ‘-’ button next to an item in their cart. If they press the ‘+’ button, then they will see that they will be purchasing 1 more of that item. If they press the ‘-’ button, then they will be purchasing one less of that item. If their item quantity reaches zero, the item will disappear from the cart.

#### Req-12: Display Items to Purchase(M)

**Description:** The items that the user is purchasing will be displayed as a list on the user’s screen, along with the quantity of the items.

**Interaction:** The user will enter the Cart Page and be greeted with all of the items they have added to the cart in a list-style fashion.

**Error Handling:** If the user does not have any items in their cart, then a message will be displayed instead of instructing the user to add items to their cart.

### Settings Page

#### Req-13: Log Out Button(M)

**Description:** A button that logs the user out of their account, assuming they are signed into an account. It also returns the user to the home page.

**Interaction:** The user will see the Log Out button displayed on the screen and click it to be logged out of their account and returned to the home page.

**Error Handling:** If the user is not logged into an account, the log-out button will not be displayed.

#### Req-14: Dark Mode Switch(W)

**Description:** A switch that changes the color scheme of a website to have darker colors. When the switch is off, the colors on the screen will be displayed normally. Off is the default for the switch.

**Interaction:** The user will see the switch on the screen and can toggle it on or off to change the web pages’ colors accordingly.

#### Req-15: Account Information(M)

**Description:** The information for the user’s current account will be displayed on the screen. This information includes the user’s username, email address, and shipping address.

**Interaction:** The user will see their username, email address, and shipping address displayed on the settings page.

#### Req-16: Edit Account Information Buttons(M)

**Description:** These are buttons that are located next to the user’s password, email address, and shipping address on the settings page. The user can click the buttons to change their account information corresponding to the user data that the button is labeled as.

**Interaction:** The user will see a change password, change email address, and change shipping address buttons on the settings page. If they click one of the buttons, then a popup will appear that will ask the user to enter their new account information.

### Search Page

#### Req-17: Display Items Relevant to Search(M)

**Description:** Items that are relevant to the user’s search will be displayed on the search page in a list-type format. This will include a picture of the item, a description of the item, the name of the item, and the price of the item. If the user does not provide a search key (they do not type anything in the search text box), then the page will display the full catalog of items on the screen instead.

**Interaction:**

* Case 1 (the user provides a search key): The user will be directed to the search page and a list of items relevant to their search will be displayed on the screen, assuming they provided a key in the search text block.
* Case 2 (the user does not provide a search key): The user will be directed to the search page with the full catalog of items being displayed before them.

**Error Handling:** If the user’s search key is not relevant to any items within the catalog, then the page will not display any of the catalog’s items. Rather, the page will display an error message instructing the user to try a different phrase.

#### Req-18: Add Item to Cart Button(M)

**Description:** There will be a button next to every item currently displayed on the screen that will add the item to the user’s cart if they do not already have that item in their cart. If the user already has the specified item in their cart upon pressing the button, then the but will increment the quantity of that item by one in the user’s cart. The button looks like a ‘+’ sign.

**Interaction:**

* Case 1 (The user does not have a certain item in their cart): Upon pressing the button, the user will have the item added to their cart with a set quantity of one.
* Case 2 (The user already has the item in their cart): Upon pressing the button, the quantity of the item being pressed will be incremented by one in the user’s cart.

#### Req-19: Filtering with Tags(W)

**Description:** There will be tags displayed on the screen that take the form of properties that certain items may contain. Next to the tags are checkmark boxes that, when the user clicks on them, toggle the tag on the screen display changes to only show the user items with those specific tags. Tags to implement would be 'Fruit,' 'Vegetable,' 'Canned,' and 'Organic.'

**Interaction:** The user will toggle tags on or off via checkmark boxes. The search results displayed on the screen will be filtered to require the tags that are toggled on.

**Error Handling:** If the tags combined with the search key for the item do not generate any relevant results, then a message will be displayed to the user.

### Order History Page

#### Req-20: Display User Order History(M)

**Description:** Display the user’s previous orders in a list-style fashion. It will display the items the user has previously ordered and how much money was spent on individual orders.

**Interaction:** The user will be able to scroll down the screen to see all their previous orders.

**Error Handling:** If the user has never ordered anything, then the page will display a message letting the user know that.

### Login Page

#### Req-21: Login Username/Email Text Field (M)

**Description:** A field where the user can type in the username or email address corresponding to their account.

**Interaction:** The user can click on the text box and type inside of it.

#### Req-22: Login Password Field (M)

**Description:** A field where the user can type in their account’s password. The password will be censored while typing it, and there will be a show password button that the user can click on to view the password being entered.

**Interaction:** The user can click on the text box and type their password inside of it. The password is hidden by black dots while the user is typing it. If the user wishes to view what they are typing, they can click a button next to the textbox that displays their password.

#### Req-23: Continue Button(M)

**Description:** A button that will log the user into their account based on the credentials provided. After the user is logged into their account, they will be directed to the home page. If the credentials do not match the credentials of any user in the database, then an error message will be displayed to the user stating that the username/email or password is incorrect.

**Interaction:**

* Case 1 (the user enters valid credentials): The user will be logged into their account and sent straight to the home page.
* Case 2 (the user enters invalid credentials): The user will not be logged in and an error message will be displayed to the user stating that their username/email or password is incorrect.

**Error Handling:** If the user clicks the continue button and a box is left unfilled, then a unique error message will be displayed telling the user they are missing a text field.

#### Req-24: Register Link(M)

**Description:** A link near the bottom of the login page that sends the user to the registration page.

**Interaction:** The user clicks on the link and gets sent to the registration page.

#### Req-25: Forgot Password Button(I)

**Description:** A link near the bottom of the login page will send the user an email address, giving the user a new temporary password to get into their account.

**Interaction:** The user will click the button to reset their password, and then they will get a popup message telling the user that an email has been sent to their inbox. The user can then go to their inbox and get the temporary password needed to get into their account.

**Error Handling:** The temporary password will expire within a few minutes if the email is being sent. The user must get into their account before the expiration, or they will need to get another email sent.

### Register Account Page

#### Req-26: Username Field (M)

**Description:** A field where the user can type in the username for their account.

**Interaction:** The user can click on the text box and type inside of it.

**Error Handling:** If the user tries to finish registering but the username matches an existing username in the database, then an error message will be displayed to the user to get them to change the name.

#### Req-27: Set Email Field (M)

**Description:** A field where the user can type in their email for their account.

**Interaction:** The user can click on the text box and type inside of it.

**Error Handling:** If the email address is invalid and the user tries to finish registration, then an error message will be displayed asking the user to change the email address.

#### Req-29: Set Password Field (M)

**Description:** A field where the user can type in their account’s projected password. The password will be censored while typing it and there will be a show password button that the user can click on to view the password being entered.

**Interaction:** The user can click on the text box and type their password inside of it. The password is hidden by black dots while the user is typing it. If the user wishes to view what they are typing, they can click a button next to the textbox that displays their password.

#### Req-30: Confirm Password Field (M)

**Description:** A field where the user will enter their projected password again to confirm that the password is correct. It will be censored unless the user chooses to click the view button to see it.

**Interaction:** The user can click on the text box and type their password inside of it. The password is hidden by black dots while the user is typing it. If the user wishes to view what they are typing, they can click a button next to the textbox that displays their password.

**Error Handling:** If the user submits the form and the text in this field does not match the text in the password field, then an error message will be displayed stating that they are not the same.

#### Req-31: Create Account Button(M)

**Description:** A button that will create the user’s account, assuming all the fields have been correctly filled out. After the account is created, the user will be sent to the home screen and logged into their new account. If the account fails to be created, then an error message will be displayed, and the account will not be created.

**Interaction:**

* Case 1 (the user enters valid credentials): The user’s new account will be created, and the user will be logged into it and sent to the home screen.
* Case 2 (the user enters invalid credentials): The user will not be logged in and an error message corresponding to the invalid credentials will be displayed.

**Error Handling:** If the user clicks the create account button and a box is left unfilled, then a unique error message will be displayed telling the user they are missing a text field.

### Payment Information Page

#### Req-32: Payment Information Text Fields: (M)

**Description:** There will be individual text fields responsible for getting the user’s payment information. The text fields required are for the debit card number, CSC number, and cardholder’s name.

**Interaction:** The user will enter the Payment Information page and be able to enter their card information into these text boxes.

**Error Handling:** If any of the text fields have incorrect payment information, a message will be displayed letting the user know.

#### Req-33: Order Confirmation Button: (M)

**Description:** A button that will make the purchase happen with the card information. After the button is pressed, the user will be sent to the order history page.

**Interaction:** The user will see and press the order confirmation button after they have provided all of their payment information.

**Error Handling:** If any of the fields are left blank or if any of the information is invalid, then the button will not proceed with the purchase and an error message will be displayed.

## Functional Requirements

### User Account/Registration/Login:

* The user must be able to create an account with a username, password, email address, and shipping address.
* The user must be able to login to their account.
* The system must be able to verify whether the user is trying to register with a valid email address.
* The system must be able to enforce minimum password complexity for registration security.
* The system must not allow a user to create multiple accounts with the same email address and/or username.
* The user must have a way to edit their account information, such as changing their email address, shipping address, or password.
* The system must be able to send a temporary password to the user in case the user forgets their password.
* If the user receives a temporary password, the system needs to make sure that the password can only be used within a few minutes of it getting sent.
* The user must have a way to delete their account if they choose.
* The user must have a way to log out of their account.

### Browsing Catalog

* The user must be able to enter a search key into a search text field and be shown relevant results to their search.
* The user must be able to apply tags to their search to further narrow down the items in the search’s catalog.
* The user must be able to browse the entire catalog of items from the database.

### Order History/Cart/Checkout

* The user must be able to add items to their cart.
* The user must be able to remove items from their cart.
* The user must be able to edit the quantity of any item in their cart.
* The user must be able to checkout with their cart (purchase their items).
* The user must have a way to enter in their payment information upon checking out. This includes a field for card number, CSC number, cardholder name, and zip code.
* The system must be able to recognize whether the user has entered valid payment information or not.
* The user must be able to confirm their shipping address before checkout.
* The user must be able to view all their previous purchases, including what they bought and how much they spent.

## User Stories:

### User Story 1: As a user, I would like to be able to register a new Fruhb account using my email address and my shipping address.

* The user starts on the Home Page of the website, not logged into an account.
* The user clicks on the profile icon on the top bar section.
* The user clicks on the “Login” button.
* The user is redirected to the Login page.
* The user clicks on the “Register a new account” button.
* The user is redirected to the Register Page.
* The user fills out the text boxes: username, password, confirm password, shipping address, and zip code.
* The user clicks the “Create Account” button.
* If the user enters valid credentials, they are redirected to the Home Page and are now logged into their new account.
* *Error Scenario:**If the user enters a poor password, invalid email address, or nonunique username, then the user will be displayed an error message.*

### User Story 2: As a user, I would like to be able to login to my Fruhb account.

* The user starts on the Home Page when not logged into their account.
* The user clicks the profile icon on the top bar of the screen.
* The user clicks on the “Login” button and is directed to the Login Page.
* The user types in their username/email address into the first text field and then their password into the second text field.
* If the information is correct, the user is logged into their account and sent to the home screen.
* *Error Scenario: If the user enters credentials that are incorrect, then an error message will be displayed to the user instead of them being logged in. If the user’s account information is incorrect a certain number of times, then the account will be locked.*

### User Story 3: As a user, I would like to be able to log out of my Fruhb account.

* The user starts on the home page and is logged into their account.
* The user clicks the gear icon at the top bar of the page and is sent to the Settings Page.
* On the settings page, the user clicks the “Log Out” button.
* The user is redirected to the home screen and is no longer logged into their account.

### User Story 4: As a user, I would like to be able to search for a specific item that I want and add that item to my cart.

* The user starts on the Home Page of the website.
* The user clicks on the search box located on the top bar of the screen.
* The user types in the name of the item they are searching for.
* The user clicks the search button and is directed to the search page.
* *Error Scenario:* *If the user’s search key doesn’t correspond to any item in the database, then the user will be given a message asking the user to try another item.*
* If the user types a valid search word, then the user will be shown related items to what they have searched for.
* The user finds the item they want, and they press the ‘+’ button next to their item of choice to add that item to their cart.

### User Story 5: As a user, I would like to be able to edit the quantity of items in my cart as well as remove an item from my cart.

* The user starts on the Home Page of the website (Logged into their account).
* The user clicks on the cart icon at the top bar on the screen and is taken to the cart page.
* On the cart page, the user can see the items they have added to their cart.
* *Alternate Scenario: If the user hasn’t added any items to their cart yet, then the user will not have anything displayed on the cart page except for a message telling the user to add an item to their cart first.*
* Besides any item in the user’s cart, there will be a text box representing the amount of the given item the user has in their cart. The user can edit the text box to change the quantity.
* The user clicks the ‘-’ button above any given item to completely remove the item from their cart.

### User Story 6: As a user, I would like to be able to purchase the items that are in my cart.

* The user starts on the Home Page of the website (Logged into their account).
* The user clicks on the cart icon at the top bar on the screen and is taken to the cart page.
* The user will click on the “Proceed to Payment” button and be sent to the payment information page.
* On the new page, the user will enter their payment information into the given text fields.
* The user will click on the “Purchase” button and be sent to the order history page, where the user can see the purchase they just made at the top of the page.
* *Error Scenario: The user will get an error message if their payment information is either invalid or if their payment can’t be completed for any reason.*

### User Story 7: As a user, I want to be able to access the settings page to edit my account information.

* The user starts on the Home Page of the website (Logged into their account).
* The user clicks on the gear icon at the top bar on the screen and is taken to the settings page.
* The user sees their account information with buttons next to them that say “edit.”
* The user edits their account by clicking one of these buttons and typing their new information into the popup box.
* *Error Scenario: The user tries to change their username, password, or email into something that is invalid based on each criterion. This will result in an error message displaying.*

### User Story 8: As a user, I want to view my order history from my previous transactions.

* The user starts on the Home Page of the website (Logged into their account).
* The user clicks on the “Order History” button on the front page of the website and is redirected to the order history page.
* The user can see a list of their previous transactions on the order history page, and they can scroll down and view older transactions. The user can also directly see exactly what they bought from those transactions and how much they spent.
* *Alternate Scenario: If the user has not made any transactions with the account they are currently using, then nothing will display on the page except a message letting the user know that their history is empty.*

### User Story 9: As a user, I would like to be able to reset my password if I forget it.

* The user starts on the Home Page of the website (Logged into their account).
* The user clicks on the gear icon at the top bar on the screen and is taken to the settings page.
* The user clicks on the “Log Out” button, and then they are logged out of their account while being redirected to the home page.

### User Story 10: As a user, I want to have the option to delete my account if I choose.

* The user starts on the Home Page of the website (Logged into their account).
* The user clicks on the gear icon at the top bar on the screen and is taken to the settings page.
* The user clicks on the “Delete Account” button and is then shown a message asking them to reenter their account password, followed by a text box.
* When the user reenters their password, they are redirected to the home page after their account has successfully been deleted.
* *Error Scenario: If the user does not enter the correct password, the user will be displayed an error message. Also, they will not be allowed to delete their account.*

# Other Nonfunctional Requirements

## Performance Requirements

**4.1.1** **Response Time**

The application must maintain a responsive user experience. Basic user interactions, such as clicking buttons or entering text, should yield a response time of less than one second. During peak operations, the response time should not exceed three seconds.

**4.1.2** **Simultaneous Users**

The application must handle high demand efficiently, providing users with a seamless experience even during peak periods. The system should reliably support up to 200 simultaneous users with, at most, a minor performance drop.

**4.1.3** **Data Retrieval**

Backend operations, such as database queries, may take a bit longer. From a user's perspective, these operations should be seamless, completing within three seconds during normal operations and ten seconds during peak operations.

**4.1.4** **Database Synchronization**

Inventory information must be viewed in real time. The stock of each item visible on the webpage should synchronize with the database, updating at least once every thirty seconds.

## Security Requirements

**4.2.1** **Password Requirements**

Passwords will comply with the following constraints:

* a length of 8 to 16 characters
* At least one uppercase letter,
* At least one lowercase letter
* At least one number
* At least one special character.

**4.2.2** **Password Lockout**

For security, users will be locked out of their accounts after ten consecutive incorrect password entries. Regaining access requires a password reset through a message sent to the registered email address.

**4.2.3** **Data Protection/Encryption**

All user data must undergo encryption employing industry-standard protocols. Passwords should be hashed using the SHA-256 algorithm.

## Software Quality Attributes

**4.3.1** **Scalability**

The application's infrastructure must be modular and adaptable to handle increased demand. It should seamlessly run on various servers and clouds.

**4.3.2** **Interoperability**

The application must function correctly on popular web browsers (Chrome, Edge, Firefox) and devices (desktop, mobile) without compromising ease of use, interactivity, or performance.

**4.3.3** **Ease of Use**

The application should provide an intuitive user interface, allowing users to navigate without external guidance and ensuring a straightforward and efficient user experience.

**4.3.4** **Availability**

The application should exhibit 99% availability, with automated maintenance scheduled during periods of low activity.

# Other Requirements

# **5.1** **External Technologies**

**5.1.1** **Database Management System**

The application will use the MongoDB database management system and will interact with it using the PyMongo Python distribution.

**5.1.2** **Web Framework**

The application will be developed using the Flask web framework.

**5.2** **Supported Languages**

Fruhb will exclusively support American English.

Appendix A: Glossary

* MongoDB is a nonrelational, document-based database management system.
* PyMongo is a Python module for interacting with MongoDB.
* Flask is a lightweight web framework for developing web applications using Python.

Appendix B: Analysis Models

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