Software Requirements Specification

for

JustEatIt

Version 0.1

Prepared by RubberDuck Team

November 14, 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Ihor Bandurovych	November 14, 2022	Initial Draft	0.1

1 Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the JustEatIt system. This document will explain the purpose and features of the system, what the system will do and the constraints under which it must operate. This document is intended for both the stakeholders and the developers of the system.

1.2 Intended Audience and Reading Suggestions

This project is a prototype for the nutrition app and it is useful for medical and fitness centers that want to assist clients with their diet management in the most efficient way and as well it is useful for the clients who will be able to manage their nutrition and food expenses.

1.3 Project Scope

This software system will provide a service for medical and fitness centers where they can manage lists of recommended dishes for their clients diet and a service for application users who can see and edit their weekly diets. The system is designed to maximize the efficiency of users' nutrition and to minimize their expenses of the ingredients for their dishes. Also, the system will be providing an intuitive and minimalistic user interface. The system contains a database with Users, Menus, Dishes, and Ingredients entities.

2 Overall Description

2.1 Product Perspective

The JustEatIt system will consist of an online registration form, an administrators' portal, a backend database to store and process user's information, ingredients, dishes and weekly menus data. This system will be used to manage the users' weekly menus and to calculate food expenses for users.

2.2 Product Features

The system will have the following major features:

1. Showing a recommended weekly menu immediately after user's registration based on the body metrics entered by the user.
2. Adding or removing dishes from the weekly menu by the user.
3. Importing of a shopping list by the user based on their weekly menu.
4. Managing of a shopping list by the user.
5. Adding or removing dishes from the favorites list by the user.
6. Adjusting the weekly menu by the system according to the list of favorite dishes of the user.
7. Displaying all available ingredients and dishes for administrators.
8. Managing list of available ingredients by the administrator.
9. Managing list of available dishes by the administrator.
10. Managing users access permissions by the administrator.

2.3 User Classes and Characteristics

There are two types of users that interact with the system: Administrators and Users. Each of these two types of users has a different use of the system so each of them has its own requirements. The Users can only use the application to enter their body metrics, create their weekly menus from existing dishes, manage their favorite dishes and form a shopping list based on the selected dishes. The Administrators shall be able to perform same actions as Users, manage users' access permissions and also manage dishes: creating new dishes, adding and removing ingredients from dishes, changing amount of ingredient in the dish, adding new ingredients to the list of available ingredients etc. They are managing the overall state of system to keep the system's integrity. The Administrators can manage the information for each dish and ingredient.

2.4 Operating Environment

Operating environment for the JustEatIt system is as listed below:

- RDBMS: PostgreSQL;
- Client/server system;
- Operating system: Unix-like (GNU/Linux, BSD family);
- Platform: Java Virtual Machine.

2.5 Assumptions and Dependencies

The system will be developed by using Java programming language. Other dependencies will also include:

- Maven
- Spring
- Hibernate
- Tomcat
- SL4FJ
- Junit
- Mockito
- Docker

The system may have an adaptive design for small screens (smartphones, tablets).

3 Requirement specifications

3.1 Functional Requirements

This section contains all functional requirements of this project. They describe how our system should work and behave, so these requirements are very important and must be implemented in full.

3.1.1 The User Functional Requirements

The list of the application user functional requirements descriptions:

1. The unregistered user shall be able to see "Log In" and "Sign Up" buttons.
2. The unregistered user shall register after filling in a registration form or log in using e-mail and password.
3. The registration form shall have the following fields: "E-mail", "Password", "Full Name", "Age", "Gender", "Height", "Weight", "Exercise Frequency".
4. The registered user shall see the page with recommended weekly menu immediately after successful registration or login.
5. The registered user shall be able to see "View Menu", "Personal Cabinet" and "Sign Out" buttons.
6. The registered user shall be able to edit personal data and change password on the "Personal Cabinet" page.
7. The registered user shall see the updated weekly menu after applying changes to their personal data.
8. The registered user shall see the page with information about the dish after selecting the dish from the weekly menu page.
9. The registered user shall see the list of favorite dishes after pressing "Favorite Dishes" button on the weekly menu page.
10. The registered user shall be able to select a dish as a favorite.
11. The registered user shall be able to remove a dish from the favorites list.
12. The registered user shall be able to change a dish in the weekly menu.
13. The registered user shall be able to select dishes which will form a shopping list after pressing "List of Ingredients" button.
14. The registered user shall see the shopping list after pressing "Form a Shopping List" button on the dishes selection page.
15. The registered user shall be able to add or remove ingredients from the shopping list.
16. The system shall update the shopping list page after the registered user applies changes to the shopping list.
17. The registered user shall be signed out after pressing "Sign Out" button.

3.1.2 The Administrator Functional Requirements

The list of the application administrator functional requirements descriptions:

1. The administrator shall be able to see "View Menu", "Add Ingredient", "Add Dish", "Edit Users", "Personal Cabinet", and "Sign Out" buttons.

2. The administrator shall be able to perform same actions as the registered user: see the recommended weekly menu page, edit personal data on the "Personal Cabinet" page, log out etc.
3. The administrator shall see a page with the "Create New Ingredient" form and ingredients list after pressing "Add Ingredient" button.
4. The "Create New Ingredient" form shall have two fields: "Ingredient Name" and "Caloric Value".
5. The system should update the page with the list of ingredients after the administrator fills the "Create New Ingredient" form and presses "Create New Ingredient" button.
6. The administrator shall be able to remove an ingredient from the list of ingredients.
7. The system shall update all dishes and all users' weekly menus dynamically after the administrator removes an ingredient.
8. The administrator shall be able to edit the ingredient's name and/or caloric value by selecting an ingredient from the list of ingredients and submitting changes to the form.
9. The system shall update all users' weekly menus dynamically after the administrator applies changes to the ingredient's name and/or caloric value.
10. The administrator shall see a page with the dishes list and "Create New Dish" form after pressing "Add Dish" button.
11. The "Create New Dish" form shall contain four fields: "Dish Name", "Link To Image", "Name and Amount of Ingredient", "Category"; and two buttons: "Add Ingredient" and "Add Category".
12. The "Create New Dish" form should show an additional field named "Name and Amount of Ingredient" after the administrator presses "Add Ingredient" in the form.
13. The "Create New Dish" form should show an additional field named "Category" after the administrator presses "Add Category" in the form.
14. The system should update the page with the list of dishes after the administrator fills the "Create New Dish" form and presses "Create New Dish" button.
15. The administrator shall be able to remove a dish from the list of dishes.
16. The system shall update all users' weekly menus dynamically after the administrator removes a dish.
17. The administrator shall be able to edit the dish's name, image link, ingredients and/or categories by selecting a dish from the list of dishes and submitting changes to the form.
18. The system shall update all users' weekly menus dynamically after the administrator applies changes to the dish's name, image link, ingredients and/or categories.
19. The administrator shall see a page with the list of existing users and two inactive buttons: "Assign Admin Rights" and "Remove Admin Rights" after pressing "Edit Users" button.
20. The administrator shall be able to upgrade selected users from the list to administrators by pressing "Assign Admin Rights".
21. The administrator shall be able to downgrade selected users from the list to registered users by pressing "Remove Admin Rights".
22. The system shall update the access privileges of affected users after the administrator makes changes to the users list.

3.2 Non-functional Requirements

This section contains all non-functional requirements of this project. They describe what the system is supposed to be, different system properties. They also concentrate on the user's expectations, so it is good to have them, although these requirements are non-mandatory.

3.2.1 Performance

The list of the performance requirements descriptions:

1. The application shall load and be usable within 1 second.
2. The system shall be able to support 1000 simultaneous users.
3. Response time for queries and updates shall be not over 100 ms for simple endpoints (e.g. displaying user information, adding dishes/ingredients) and not over 200 ms for complex endpoints (e.g. ration calculation).

3.2.2 Reliability

The list of the reliability requirements descriptions:

1. The system shall be operational even if errors have occurred.
2. The system shall have at least 80% uptime.
3. Downtime after a critical failure shall not exceed 5 hours. The average period of the downtime shall be between 2 and 5 hours.

3.2.3 Usability

The list of the usability requirements descriptions:

1. The user shall see all their changes or changes made by the administrator dynamically.
2. The interface shall be easy to master without additional hints and allow users to accomplish their goals without errors.

3.2.4 Supportability

The list of the supportability requirements descriptions:

1. The system shall be available and usable in Google Chrome, Microsoft Edge, Opera, Mozilla Firefox, and Safari browsers.

3.2.5 Maintainability

The list of the maintainability requirements descriptions:

1. The system shall be easy for testing.
2. The application shall make use of continuous integration in order to deploy features and bug fixes rapidly without facing downtimes.

3.2.6 Security

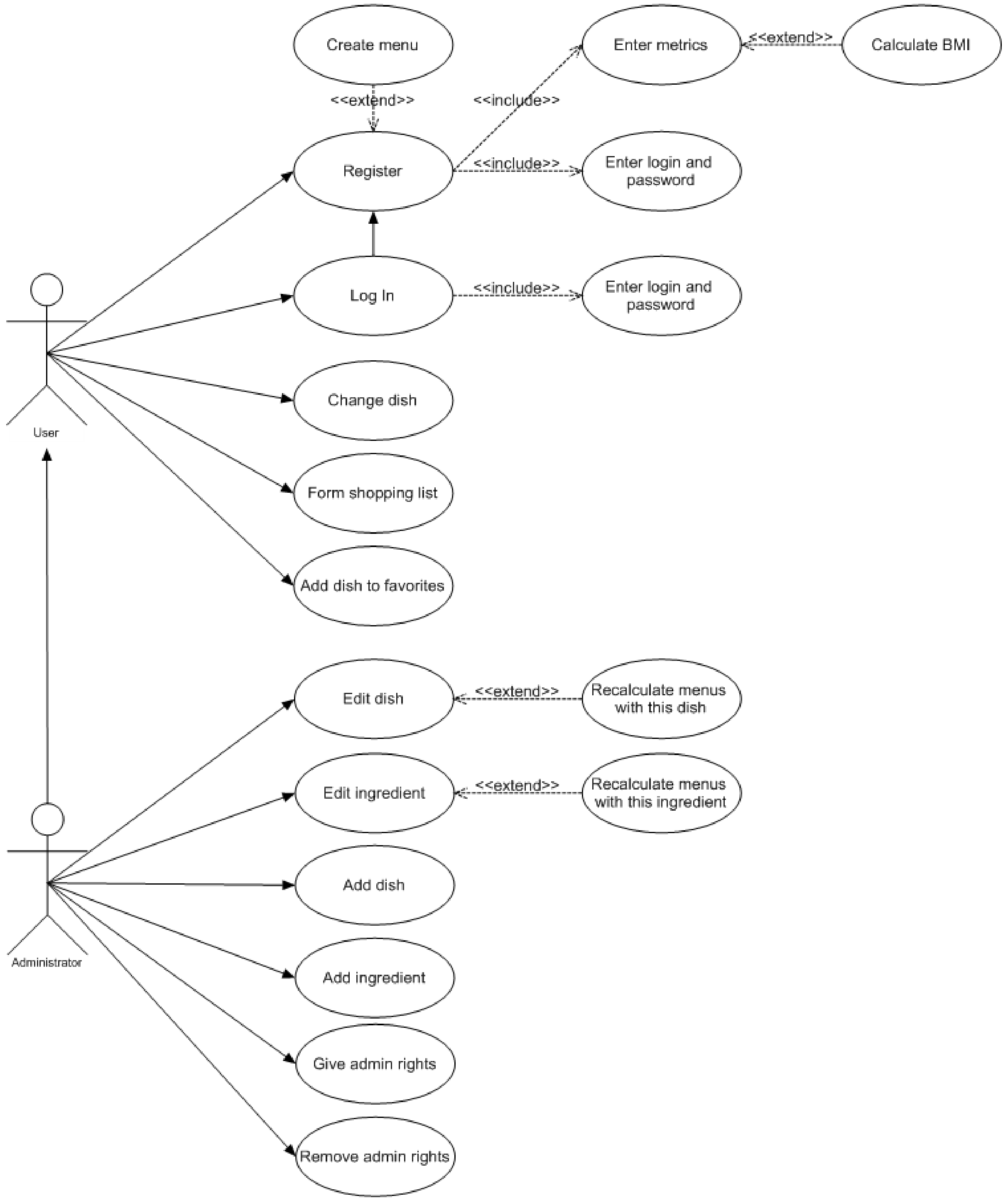
The list of the security requirements descriptions:

1. Only authorized users are allowed to make changes in the application.
2. Users’ passwords shall have the following restrictions: minimal length; minimal number of digits and letters in the password.

3. User access permissions for application may only be changed by the Administrator of the system.
4. All external communications between the server and clients must be encrypted.

4 Other Requirements

4.1 Use case diagram



This is a use case diagram of JustEatIt project. The system has two actors: User and Administrator.

4.2 User Stories and Acceptance Criteria

User story: As a guest
I want to be able to register for a service
So that I will be able to use it

Acceptance criteria:

- A guest can submit a registration form only by filling in all required fields
- The guest shall enter their e-mail, password, full name, age, gender, height, weight and exercise frequency in the registration form
- The registration page shall display an error if the guest enters invalid e-mail or password that doesn’t match the security restrictions
- User should see a page with recommended weekly diet after successful registration

User story: As a user
I want to be able to update my body metrics
So that I will get an adjusted weekly menu

Acceptance criteria:

- A user can edit his personal data on "Personal Cabinet" page
- The user shall see updated weekly menu after they change their age, gender, height, weight and/or exercise frequency and submit the changes

User story: As a user
I want to be able to change a dish in the weekly menu
So that I will have dishes in the menu that I like

Acceptance criteria:

- User can change a dish in the weekly menu by pressing "Change Dish" button near the dish on the weekly menu page
- User shall see the page with the list of possible replacements after pressing "Change Dish" button
- User shall select a dish from the list of possible replacements
- User shall see updated weekly menu page after selecting a replacement dish

User story: As a user
I want to form a shopping list
So that I will be able to plan my purchases

Acceptance criteria:

- User can import a shopping list of ingredients to buy by pressing "List of Ingredients" button on the weekly menu page
- User can add or remove an ingredient from the shopping list
- User can change amount of an ingredient in the shopping list

User story: As a user
I want to add a dish to my favorites
So that I will be able to track which dishes I liked

Acceptance criteria:

- User can add a dish to favorites by pressing "Add to Favorites" button near the dish on the weekly menu page
- User shall see the list of favorite dishes by pressing "View Favorites" button on the weekly menu page
- User can add or remove a dish from the list of favorite dishes

User story: As an administrator
I want to manage available ingredients
So that I will be able to add different ingredients to dishes

Acceptance criteria:

- The "Add Ingredient" page should contain "Create New Ingredient" form and the list of ingredients
- The administrator shall enter name of an ingredient and its caloric value in the form
- The "Add Ingredient" page should show updated list of ingredients with entered data after the administrator submits the form
- The "Add Ingredient" page should show updated list of ingredients after the administrator presses "Remove" button near an ingredient from the list
- All dishes that had a removed ingredient should be updated dynamically
- The administrator can edit a dish by selecting a dish at "Add Ingredient" page
- A page with "Edit Ingredient" form should appear after the administrator selects the dish to update
- The administrator shall change name of an ingredient and its caloric value in the form
- The administrator shall receive a message that their edits succeeded after submission of the form
- The "Add Ingredient" page should show updated list of ingredients with entered data after the administrator submits the form
- All user weekly menus that have the updated ingredient should be updated dynamically

User story: As an administrator
I want to manage available dishes
So that users will get more diverse weekly menus

Acceptance criteria:

- The "Add Dish" page should contain "Create New Dish" form and the list of dishes
- The administrator shall enter name of a dish, link to an image that shows the dish, names and amount of ingredients in the dish and categories of the dish in the form
- The "Add Dish" page should show updated list of dishes with entered data after the administrator submits the form
- The "Add Dish" page should show updated list of dishes after the administrator presses "Remove" button near a dish from the list
- All user weekly menus that had the removed dish should be updated dynamically
- The administrator can edit a dish by selecting a dish at "Add Dish" page
- A page with "Edit Dish" form should appear after the administrator selects the dish to update
- The administrator shall change name of a dish, link to an image that shows the dish, names and amount of ingredients in the dish and categories of the dish in the form
- The administrator shall receive a message that their edits succeeded after submission of the form
- The "Add Dish" page should show updated list of dishes with entered data after the administrator submits the form
- All user weekly menus that have the updated dish should be updated dynamically

User story: As an administrator
I want to promote a user to administrator role
So that more people will manage the system

Acceptance criteria:

- The "Edit Users" page should contain the list of users with their assigned roles
- The administrator shall upgrade a user to administrator role by pressing "Assign Admin Rights" button
- The administrator shall downgrade an administrator to user role by pressing "Remove Admin Rights" button
- The users affected by the change shall see their role change dynamically