The Naive Baker
Group 08

SRS

Project By:

202001062 - Boricha Vinal

202001089 - Priyank Pitliya

202001094 - Deep Rakhasiya

202001099 - Raj Aditya

202001100 - Shobit Verma

202001103 - Dhruv Prajapati

202001108 - Nikhil Jethanandani

202001110 - Vihar Shah (Group Representative)

202001116 - Gaurav Shah

Table of Contents

Problem Statement

Imagine a world where your kitchen helps you decide what to cook using your pantry and preferences. Build a project to let you explore new recipes with ingredients you already have at hand. Enter your pantry quickly using autocomplete and machine generated ingredient suggestions, then find out what you can cook! Use the filters to limit the amount of time you have, or to select the particular ingredients you feel like cooking with today.

1. Introduction

1.1) Purpose

1) The purpose of this naive baker software is to help a section of people who have available ingredients on hand and are in a confusion about what dish should be made out of it.
2) Also the basic purpose of this software is to save the time and effort of the user to make a dish.
3) This will also help in reducing the food wastage problem.
4) This software will help all those people who want an answer to all those questions easily and Share their talents related to cooking with the world on the same platform. Here users can easily search for recipes with different search methods mentioned. Users can find the recipe based on the ingredients available in the kitchen and cook the best out of it.

1.2) Intended Audience and Reading Suggestions

The intended audience for this document include software developers, testers, project managers, chefs, users who wish to cook and save time. a as well as the other stakeholders who are involved in the development, design and implementation of this system. This document will ensure that all the people involved

have a clear, concise and shared understanding about the system requirements, product goals and its deliverables.

1.3) Scope

1) Helpful for the people who have dietary restrictions or any dietary requirements which could either have been prescribed by any fitness coach or due to any medical condition.
2) Could be boon for the cooks at home for whom the whole process of making a delicious dish would be simplified.
3) It could be very helpful for the beginners and inexperienced cooks

2) Overall description

2.1) Product perspective

1) Purpose: The purpose of the project is to develop a baking product that satisfies a specific need or desire in the market. For example, the product may be designed to offer a healthier alternative to traditional baked goods, or it may be geared towards a specific target audience, such as vegan or gluten-free consumers.

2) Features: The features of the product should be clearly defined and should include a detailed description of the ingredients, the baking process, and the final product. The features should also include any unique selling points that differentiate the product from other similar products on the market.

3) User Experience: The user experience is an important aspect of any product, and a baking project is no exception. The product should be designed with the end user in mind, and should be easy to use and enjoyable to consume. This may include considerations such as packaging, serving suggestions, and flavor profiles**.**

2.2) Product functions

As the system is designed to be able to fulfill the requirements of different users of the 'The Naive Baker', it should have the following features:

1) User Friendly Interface : The interface of the web application should be minimalistic, simple and easy to use and should be functional enough to provide detailed information in a clean manner all while being efficient.

2) Registered user Management: The web application should provide the premium user a platform through which they can search different recipes and view their own dashboard and personal details.

3) Premium user Management: The web application should provide the premium users to upload their own recipes which would be visible to all the users.

4) Security and Privacy: As the system is being used by multiple users with different authority levels it should be safe and robust enough to distinguish between the roles of different users. Also it should be secure in terms of holding personal and sensitive information.

2.3) User class and characteristics

The system has 3 major users with each having different functionalities and authoritative levels:

1) Premium users:

    a) Registered in the 'The Naive Baker' system.
    b) Needs a simple and intuitive system to make cooking easier.
    c) Use the application to search various recipes based on different filters.
    d) May use the system to upload their own recipes.
    e) May use the system to provide the feedback.

2) Non-Premium users:

    a) Registered in the 'The Naive Baker' system.
    b) Needs a simple and intuitive system to make cooking easier.

c) Use the application to search various recipes based on different filters.
d) May upgrade to Premium version

2.4) Operating environment

The following were used for making 'The Naive Baker' system:

1) NoSQL database
2) Platform: React/CSS/Node JS
3) OS independent platforms

3) Specific Requirements

3.1) External Interfaces

1) VS Code:
   IDE to help us connect to github and maintain the project files while working on it.
2) GitHub:
   A version control system to help us maintain the project while collaborating with the entire team.
3) React:
   React is a JS framework for front-end. It is helpful in creating fast single page applications for small-to-medium sized projects. It also provides support for excellent code readability.
4) MongoDB:
   Since NoSql is required, MongoDB is the best option available. The Document Data Model of MongoDB is a powerful way to store and retrieve data. Also the MongoDB is available in many major public clouds.MongoDB provides a good user experience and also provides a good scalable architecture
5) Chrome DevTools:
   It provides us the tool to inspect and fix the errors in our code.It helps us analyze the performance of our webpage. It helps us to check the responsiveness of the page on different devices and browsers.
6) Bootstrap:
   Since bootstrap contains basic boilerplate code, it becomes easier to code quickly and save time. It helps us to make our website responsive.
7) CodePen:
   It provides the developers with functions to test and run our code (ex. HTML, CSS, JavaScript, etc.) directly without uploading the website on the browser. 2 Group 08 - Naive Baker IT 314 Lab 04
8) Node.js:

Node.js is a back-end JavaScript runtime environment which is used for allowing developers to code applications and also to run them simultaneously.

3.2) Functions

1) USER LOGIN

The user would first have to login into the system or create an account if (s)he doesn't have any. To create an account, the user has to enter their email id, password and a unique username. After creating an account, the user can log in. After logging into the system, the user can see their profile or dashboard.

2) SEARCH FILTER

a. Duration: The user can search dishes and recipes based on the time required for making a particular dish.
b. Ingredients: The user can search for dishes and recipes based on available ingredients. The dishes with the most matching ingredients are returned.
c. Type of Cuisine: The user can select the type of cuisine that (s)he wishes to have. These may include continental, Indian, Mexican, regional etc.
d. Health Requirements: The user can select or choose a particular dish or recipe based on a range of calorie intake.
e. Ratings: a way to filter based on the average ratings by the various users.
f. Meal: The user can filter out what meal (s)he wants from breakfast, lunch, snacks and dinner.

3) ADDITION OF NEW DISHES

Users can add a new dish or recipe to the database if they wish to through a form.

4) RECOMMENDATIONS AND ‘DOTD’

The user sees recommendations on the home page based on their past searches. A “Dish of The Day” is also available for the user, which is selected randomly.

5) RECIPE TUTORIAL

A user can see the step-by-step instructions on how the cook the item, which the chef of that item provides. Alternatively, an embedded video link can also be provided.

6) PREMIUM USER

A user can search for recipes a limited number of times, after which they need to buy a subscription to the website to get unlimited access.

7) DISH RATING

The user would have the feature to rate dishes based on their experience. Each dish would thus have its rating, which would be helpful while searching/filtering.

8) VEG/NON-VEG/VEGAN

The user would have the option to filter out all the options or dishes available.

3.3) Performance Requirements

1. The software should be able to handle multiple users simultaneously, so the server should be compatible with heavy traffic.
2. Software should respond quickly to user queries.
3. Software should work with minimum lag time and access the database in the shortest time possible.
4. The software should have high throughput to ensure that sound events are recognized and logged in real-time.

3.4) Software System Attributes

1. Reliability: The software should be reliable and provide accurate results based on user needs

2. Availability:
   1. The software should be easy to access and should be interactive.
   2. The software should have a straightforward user interface.
   3. Recipe queries latency should be minimum.
   4. User registration and login should be easy and quick.

3. Security:
   1. The user's data should be kept secure for privacy reasons.
   2. The software should be safe from unauthorized access, and no third party should be able to interfere with the user data.

4. Maintainability:
   1. The system should be able to quickly restore to operation status if any repair actions have been performed.
   2. The system should be easy to repair and maintenance activities can be performed easily.

5. Portability
    1. The software should be able to operate on other operating systems other than the system on which it was developed.
    2. Should be able to yield the same results on a different environment.

4) System Features

4.1) Use cases:

The different use cases used in the use case diagram are

1. Sign Up/Sign in
2. Search Recipe
    - 2.1. Search by Ratings
    - 2.2. Search by Meals
    - 2.3. Search by Health Requirements
    - 2.4. Search by Cuisine
    - 2.5. Search by Ingredients
    - 2.6. Search by Duration
3. Dashboard
    - 3.1. Check watch history
    - 3.2. Saved Recipe
    - 3.3. Clean watch history
4. See your recipes
    - 4.1. Add recipe
    - 4.2. Edit recipe
    - 4.3. Delete recipe
5. Check database
6. About us Page
    - 6.1. Feedback

4.2) Use case scenario:

1) SIGN-UP

Name: Register
Actors: Non premium users.
Goal: To register the new users visiting our website so that we can store and handle their data.

Pre-Condition: The system does not have the information regarding the users trying to access the website.
Trigger: The Register option registers the new users to access the system.
Actual Flow:

1)The user enters the email address and password.
2) The user enters the confirm password.
3) The user confirms registration by clicking on submit.

Alternate Flow :

1)If the confirm password does not match with password entered, user is prompted to re-enter the password

Post-Condition: After the completion of this user case, we successfully register the new users.

2) SIGN-IN

Name:SignIn
Actors: Both Premium and Non Premium users and admin.
Goal: To allow already registered users to access our system.
Pre-Condition: The user should already be registered.
Trigger: The SignIn option gives the user the option to access different functionalities of the system.
Actual Flow :

1)The user enters the email address and password.
2) The user confirms registration by clicking on submit.

Alternate Flow :

1) If the entered email and password are not valid then the user is prompted to re enter the details.

Post-Condition: The user successfully Signs In on our website.

3) SEARCH

Name: Search Recipe
Actors: Users divided in two categories as Prime User and Non-prime user as per their subscription.
Goals: Users can search according to different filters for multiple recipes.
Pre-Condition: Users must register themselves before using search recipes .
Trigger: When a user wants to search for a particular recipe.
Actual Flow of Search:

1) User will first select the filter.
2) Users will have to type the keyword for a particular filter.
3) Users will be recommended a number of recipes according to their selected filter.
4) User will now have to select his preferable recipe from the recipes shown.

Alternate Flow of Search:

1.1 If The user is unable to select a filter.
1.2 If the entered keyword does not exist.

Different methods/filters to search:
1.Duration: The user can search dishes and recipes based on the time required for making a particular dish.
2.Ingredients: The user can search for dishes and recipes based on available ingredients. The dishes with the most matching ingredients are returned.
3.Type of Cuisine: The user can select the type of cuisine that (s)he wishes to have. These may include continental, Indian, Mexican, regional etc.
4.Health Requirements: The user can select or choose a particular dish or recipe based on a range of calorie intake.
5.Ratings: a way to filter based on the average ratings by the various users.

6.Meal: The user can filter out what meal (s)he wants from breakfast, lunch, snacks and dinner.
Post-Condition: After the completion of the user case the user will be able to see multiple recipes according to given keywords.

4) See your recipe

Name : Recipe management
Actors : Premium user, Admin
Goals : This allows the actor to make changes in the database of the recipes.
Pre condition : The user should be logged in with a premium account or the admin account.
Description : In this use case, the actor can edit/delete his own recipe or add a new one in the database.
Trigger : When the actor clicks to see his/her recipes uploaded on the website.
Related use cases : Edit Recipe, Add recipe, Delete recipe

Actual Flow :

1) The user will see your recipe option to perform the functionality.

2) The system will show all the recipes uploaded by the user.

3)User will then perform the functions that they wish, such as

Delete Recipe :

3.1) The user will be able to view all the uploaded recipes.

3.2) Delete option would be available beside each recipe.

3.3) The user would select the options and the recipe would be deleted from the user's database.

In the same way, we can have the actual flow of other 2 functionalities
Post Condition : The user will see the changes made in the recipe database.

5) Dashboard

Name:Dashboard
Actors: Users, admin
Goals: To carry all the information regarding the user and admin has the control on it
Pre-Condition : User must login/ sign in before accessing the dashboard.
Trigger: When the user clicks on the profile icon placed at the top right corner.
Related use cases : Check watch history, Clear watch history, Saved recipes
Actual Flow :

1. The user will have to click on the profile icon to display the dashboard panel.
2. The user will click on the functions provided inside the dashboard.

Watch History :

2.1. The user will be able to see all the recent recipes it visited.

Delete Watch History :

2.1. The user will select the recipes to delete from the history database.
2.2. Users will then confirm the deletion of the recipes .

Saved Recipe :

2.1. User will click on the bookmark icon on the dashboard.
2.2. From there the user can access all the saved recipes on his account.

Post Condition: The user will be able to see the parameters and data of the functions provided in the Dashboard.

6) About Us page

Name: About Us Page

Actors: Users, admin
Goals: To provide information about the help desk which will help to resolve the technical issues.
Trigger: When the user encounters an error which he has no idea about.
Actual Flow:

1. The user encounters a problem.
2. The user selects About Us Page.
3. The user can also provide feedback about the site.

Post Condition: The user will be able to see information regarding the About Us Page and possibly resolve his/her issue.

4.3) User stories

As a user (recipe-seeker), I want to find recipes which include my ingredient list, so that I can cook from available ingredients in my kitchen.

As a user (recipe-seeker), I want to find recipes of various cuisines, so that I try different dishes.

As a user (chef), I want to upload my recipes for others, so that they can follow my recipes for different dishes.

As a user (recipe-seeker), I want to create my personal account, so that I can see my personalized feed and dashboard features

As a user (recipe-seeker), I want to find recipes which can be prepared in a particular time so that I can manage my time properly.

As a user (chef), I want to check the trending recipes and new steps through how a particular dish could be prepared.

4.4) Use case response/stimuli

1) Stimulus: User wants to Login into the system
   Response: System displays a login screen with inputs for username and password and after the user enters the details. The system validates the details and lets the user enter the system.

2) Stimulus: User wants to Sign up into the system
   Response: System displays a sign-up screen with inputs for email-id, username and password and after the user enters the details the system. The profile is created by the system and it lets the user enter the system.

3) Stimulus: User wants to view dashboard
   Response: The user would have to perform task 1 and then on clicking on the dashboard option, the personal information would be displayed.

4) Stimulus: User wants to update the password
   Response: The user would have to open the dashboard and then click on 'Reset Password' button which would open a new page which would ask the user for inputs like new password and confirm password.

5) Stimulus: User wants to search a recipe
   Response: The user would select the search option from navbar and then the user would be given the option to search a recipe by typing the name or using various filters like cuisine, time, meal etc.

6) Stimulus: User wants to upload a recipe
   Response: The user would select the dashboard option from the navbar and then the user would click on 'upload your recipe' option and then a new page would open up which would ask the user for inputs such as Recipe name, type, description, steps etc.

7) Stimulus: User has forgotten a password
   Response: While logging in, the user would select the 'forgot password' option and then would be redirected to a new page which would ask the user input for email-id. THe user will then receive a mail to reset the password.

8) Stimulus: User wants to see step by step tutorial of a Recipe
   Response: The user would have to click on the Recipe card which would redirect it to a new page where the required details would be given.

9) Stimulus: User wants to visit the social media pages of 'The Naive Baker'
   Response: The user would have to click on the social media icons present at the footer which would redirect it to respective social media pages such as Instagram, twitter and facebook.

4.5) Non-functional requirements:

1) PERFORMANCE:

    1. The software should be able to handle multiple users simultaneously, so the server should be compatible with heavy traffic.
    2. Software should respond quickly to user queries.
    3. Software should work with minimum lag time and access the database in the shortest time possible.

4. The software should have high throughput to ensure that events are recognized and logged in real-time

2) SCALABILITY

1. Since the number of users can increase soon, it should be scalable.
2. Performance should be degraded while implementing scalability.

3) SECURITY,PRIVACY AND DATA INTEGRITY

1. The user's data should be kept secure for privacy reasons.
2. The software should be safe from unauthorized access, and no third party should be able to interfere with the user data.

4) ACCESSIBILITY AND USABILITY

1. The software should be easy to access and should be interactive.
2. The software should have a straightforward user interface.
3. Recipe queries latency should be minimum.
4. User registration and login should be easy and quick.

5) RELIABILITY

1. The software should be reliable and provide accurate results based on user needs.