



CSD-345 Software Design Lab

Recipe Blog

Team:

Pentapati Meghana	1910110269
Siri Akanksha Grandhi	1910110393
Sree Sahithi Kantamneni	1910110402

1. Problem Definition

Recipes are important because they contain the information necessary to make a dish properly. As with any set of instructions, you rely on them to give you all of the information you need. There isn't always someone nearby who has that knowledge.

Recipes also ensure standardization. If you eat a Burger King Whopper in San Francisco, fly to Boston and eat another one, they will be the same.

Recipes also help control costs. Restaurants depend on their cooks to follow recipes so their food cost stays in line. The menu price of an item is determined by costing out a recipe. Also, recipes can help improve what you've already done, if you don't like it then you can make small adjustments to that. Using a recipe helps you to create something uniformly and consistently. It gives you an idea of ingredients to shop for. It can be a way of commemorating a special event.

Cooking is a great hobby. A lot of people consider it as a mode of relaxation. The Recipe Blog Application is a software, which provides an interface for users where one can gain any recipe information in an effective way and can submit their own recipe.

Here, the information the software provides is not actual raw data directly given to the user but all the information will be classified, categorized and will be clean so that a user who wants specific information about any topic could easily navigate through the application and get the required information he wants.

Users can also submit their own recipes, which increases the efficiency of the blog.

2. Introduction

2.1 Purpose

Food blogging is a complicated interweaving of gourmet culinary interests with blog writing and photography interests. The majority of blogs feature photographs shot by the author, and some of them specialise in culinary photography. So our idea is to build a platform where users can have access to some of the major fields in the culinary sector that are getting digitalised.

2.2 Document Conventions

The document is prepared using Google Docs and has used the font type 'Georgia'. The fixed font size that has been used to type this document is 14pt with 1.15 line spacing. It has used the bold property to set the headings of the document. All pages are numbered, the numbers appear on the lower right hand corner of the page.

2.3 Intended Audience:

The audience that we are targeting are general users who are interested in trying out new recipes and want to gain knowledge on all cuisines, utilize the application's features to make a delicious meal.

2.4 Product Scope and Additional extra information:

Through the website, a user can choose from the available features that are mentioned depending on their needs and requirements. Our job is to try and minimize the extra effort a user must keep by providing them with the data and information as accurate and common as possible that is available. We expect to bring and deliver those information to the users in the mentioned features as briefly as possible.

3. Description of Project

3.1 Product Features

- Home page: Consists of almost all the data, making it user friendly. It consists of a search bar, various categories of food and an option to submit recipes by the user.
- Categories: Different types of cuisines are represented in categories.
- Explore Latest: All the latest recipes are present here.
- Explore Random: Everytime user clicks on this, a random recipe appears.
- Search Bar: Users can search a recipe using this.
- Recipes: Each recipe has its own page and all the details of recipe including selection of ingredients and reviewing the recipe are present here.
- Get ingredients: User can see the selected ingredients and their cost and can proceed to check out by giving the details of his/her address.
- Submit recipe: Users can submit their recipe using this.

3.2 User characteristics:

The application will be used by the end user to search for various recipe information which has to be simple, easy to use and less time taking.

It must have a good looking and pleasant Graphic User Interface (GUI) with light colors to look attractive.

The user should be able to access the application 24x7.

The user need not to have any specific knowledge of how the application is processing the information or how the application is operating internally.

A user who has the basic knowledge of how to operate a computer or mobile should be able to use this application with ease as he\she does not need to have any technical knowledge.

3.3 General Constraints:

- The application must be delivered by November 28th 2021.
- The application must be user friendly.
- The application is targeted to run all over the country. As a huge application it needs to face several constraints and limitations which may lag the performance.
- As the application contains huge amount of data, the classification of functions and data are some essential constraints to be considered throughout the development and use of system

3.4 Assumptions and Dependencies

The application will be focusing on providing recipe information from a large data set in a segregated manner online.

So our assumption is that the internet facility, power supply and the internet bandwidth must always be available and functioning.

The application has a dependency on the database server in which the whole data is being stored. So another assumption is that the database should be able to handle multiple requests by the users without any discrepancies.

4. Files

4.1 app.js

This file connects the project to the port and acquires all the essential attributes. This also sets the view engine.

4.2 styles.css

This file contains the basic styling required for all the pages.

4.3 recipeRoutes.js

This file contains all the routes from the homepage like exploreRecipe, submitRecipe, search, explore-latest, explore-categories, get-ingredients, explore-random.

4.4 recipeController.js

This file acquires the database and has the code that needs to be executed after going to each of the routes mentioned above. For each route mentioned above, a function would be present in this file. For example, in the case of search, it takes the input and searches for it in the database and returns the results.

4.5 database.js

By acquiring mongoose, mongodb is connected and tables are created using below schemas.

4.6 Category.js

Schema for the category table is mentioned here.

4.7 Cost.js

Schema for the cost table is mentioned here.

4.8 Recipe.js

Schema for the recipe table is mentioned here.

4.9 main.ejs

This contains the basic view of the homepage. It includes html, styling from styles.css and also bootstrap. The body is not yet defined over here.

4.10 index.ejs

This contains the body of the homepage. Categories, search, explore-latest, explore-random, search bar, recipes, submit recipe are defined here.

4.11 categories.ejs

This file shows all categories and the recipe cards on the homepage.

4.12 explore-latest.ejs

Shows the latest recipes in /explore-latest route.

4.13 explore-random.ejs

Any one of the recipes randomly is selected and shown on the screen using exploreRandom function in recipeController.js

4.14 get-ingredients.ejs

After selecting the necessary ingredients, the get-ingredients page is visible to the user where the total cost and the details of the address are available where the user can enter his/her details for the delivery.

4.15 recipe.ejs

This has the structure of the recipe page. It contains the name of the recipe, its category, ingredients, option to select ingredients and also review the recipe.

4.16 search.ejs

This page shows up when any recipe is searched using the search bar.

4.17 submit-recipe.ejs

When the user clicks on the submit recipe this page shows up. It contains all the details like email, recipe name, its image, process. After the user uploads all the details, the recipe can be seen in the latest recipes.

5. External Interface Requirements

5.1 Software Interfaces

5.1.1 Development End

- **Javascript** is used because JavaScript has become integral to the Internet experience as developers build increased interaction and complexity into their applications. Search engines, ecommerce, content management systems, responsive design, social media and phone apps would not be possible without it. Using JavaScript for this application would make tasks much simpler. Also the database connectivity and web

connectivity is much simpler in JavaScript. One would always require less effort and time to build an application using Javascript.

- **MongoDb** will be used for the purpose of storing and updating the data frequently. Using a database has many advantages as the complete data of the application can be stored in a single location which can be accessed from anywhere and anytime. The data stored in a database is much more secured

5.1.2 Client End / User End

HTML, CSS & JavaScript will be used as a part of front end development. As JavaScript can be used in both front end and back end, developing a web page using JavaScript would be a great choice as it offers a great flexibility to create a page of exact desired requirements. As UI is the most important for an application, the webpage would be styled using CSS which enables to make the page a beautiful one. We also use a few **Bootstrap classes** which expand the options for better styling of the page.

5.2 Hardware Interfaces

A web page is the most easily accessible one for any user across the world. As the world is now completely dependent on the internet, the internet is now available to almost every part of the world. So making this application as a web application, it would reach many people and it can target a huge number of users. Accessing a web page is a one click task and hence accessing the features of this application would also be a matter of one click in our hand.

As the application is a web application, for the client/user to run the following application, below are the requirements or prerequisites:

- a. A PC, Laptop, mobile, tablet, notebook or a smart TV with a

display of minimum screen resolution.

b. An active internet connection [Wi-Fi / Mobile Data / LAN / Ethernet] c. A web browser [Chrome / Firefox / Opera / Safari etc...]

6. Other non functional requirements

6.1 Software Quality Attributes

- Availability- The application shall be available all the time. There will not be any shortages in between at any time.
- Maintainability- The system shall provide the capability of backing up the data due to any crashes and it has the ability to maintain, modify data in databases. The software should be open for any new changes/requirements through feedback and should be updated accordingly.
- Correctness- A bug free software which fulfils the correct need and requirements of the users.
- Usability-The application can be used at any number of times again and again without distortion.