

IT 314 - Project

Naive Baker

Group 08

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 Content

1. Introduction	2
1.1. Purpose	2
1.2. Scope	2
1.3. Definitions, Acronyms, and Abbreviations	2
1.4. References	2
1.5. Overview	2
2. Overall Description	2
2.1. Product Perspective	2
2.1.1. System Interfaces	2
2.1.2. Interfaces	2
2.1.3. Hardware Interfaces	2
2.1.4. Software Interfaces	2
2.1.5. Communication Interfaces	2
2.1.6. Memory Constraints	2
2.1.7. Operations	2
2.1.8. Site Adaption Requirements	2
2.2. Product Functions	2
2.3. User Characteristics	2
2.4. Constraints	2
2.5. Assumptions and Dependencies	2
2.6. Apportioning of Requirements	2
3. Specific Requirements	2
3.1. External Interfaces	2
3.2. Functions	2
3.3. Performance Requirements	2
3.4. Logical Database Requirements	2
3.5. Design Constraints	2
3.5.1. Standards Compliance	2
3.6. Software System Attributes	2
3.6.1. Reliability	2
3.6.2. Availability	2
3.6.3. Security	2
3.6.4. Maintainability	2
3.6.5. Portability	2
3.7. Organization of Specific Requirements	2
3.7.1. System Mode	2
3.7.2. User Class	2

3.7.3. <i>Objects</i>	2
3.7.4. <i>Features</i>	2
3.7.5. <i>Stimulus</i>	2
3.7.6. <i>Response</i>	2
3.7.7. <i>Functional Hierarchy</i>	2
4. <i>Change Management Process</i>	2
5. <i>Documents Approval</i>	2
6. <i>Supporting Information</i>	2

1. Introduction

1.1. Purpose

1. The purpose of this naïve baker software is to help a section of people who have available ingredients on their hand and are in a confusion that what dish should be made out of it.
2. Also the basic purpose of this software to save the time and effort of the user to making a dish.
3. This will also help in reducing the food wastage problem.

1.2. 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 bon 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

1.3. Definitions, Acronyms, and Abbreviations

1.4. References

1.5. Overview

2. Overall Description

2.1. Product Perspective

- 2.1.1. System Interfaces
- 2.1.2. Interfaces
- 2.1.3. Hardware Interfaces
- 2.1.4. Software Interfaces
- 2.1.5. Communication Interfaces
- 2.1.6. Memory Constraints
- 2.1.7. Operations
- 2.1.8. Site Adaption Requirements
- 2.2. Product Functions

2.3. User Characteristics

5

2.4. Constraints

2.5. Assumptions and Dependencies

2.6. Apportioning of Requirements

3. Specific Requirements

3.1. External Interfaces

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

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.

2) ADDITION OF NEW DISHES Users can add a new dish or recipe to the database if they wish to through a form.

3) 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.

4) RECIPE TUTORIAL

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

5) 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.

6) 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.

7) 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. Logical Database Requirements

3.5. Design Constraints

3.5.1. Standards Compliance

3.6. Software System Attributes

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

- 3.6.2. 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.

- 3.6.3. 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.

- 3.6.4. 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.

3.6.5. 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.

a. Organization of Specific Requirements

- i. System Mode
- ii. User Class
- iii. Objects
- iv. Features
- v. Stimulus
- vi. Response
- vii. Functional Hierarchy

3. Change Management Process

4. Documents Approval

5. Supporting Information