A red and black logo

Description automatically generated

**SECD 2613-15 SYSTEM ANALYSIS AND DESIGN**

**2023/2024 – SEMESTER 2**

**PHASE 2**

**RECIPE MANAGER**

**FACULTY OF MJIIT**

**Section 16**

**Supervised by: Dr Amy Hamidah**

**GROUP 1**

|  |  |
| --- | --- |
| **NAME** | **MATRIC ID** |
| HASAN AMMAR ABDULAZIZ AL-TALIB | A23MJ0018 |
| Mohamed G. S. Elmasharawi | A23MJ0009 |
| MAAROF SAQR YOUSEF HASAN | A23MJ4006 |
| ELHASSEN KHATTARY | A23MJ4002 |

**TABLE OF CONTENT**

1.0 OVERVIEW OF THE PROJECT**3**

**2.0 PROBLEM STATEMENT 3**

**3.0 PROPOSED SOLUTIONS 4**

**4.0 CURRENT BUSINESS PROCESS / WORKFLOW 5**

**5.0 LOGICAL DFD (AS-IS) 8**

**6.0 SYSTEM ANALYSIS AND SPECIGICATION 21**

**7.0 PHYSICAL SYSTEM DESIGN 34**

**8.0 SYSTEM WIREFRAME ( I/O DESIGN ) 41**

**9.0 SUMMARY OF THE PROPOSED SYSTEM 42**

1.0 OVERVIEW OF THE PROJECT

The Recipe Manager project is designed to be a simple application that lets you store and find the recipes you like. Due to increased interactions at home and the different recipes available online, there is a need for an application that can help manage the activities in the kitchen. This app will allow the user to create, modify, remove, and look for recipes using certain ingredients or type. The best and efficient way of achieving this is to make the use of recipes manageable, and boost the process of cooking while at the same time minimizing fuss of maintaining different recipes.

2.0 PROBLEM STATEMENT

In the present world, recipes are found on different media sources – In books, on papers, on the Internet, and in various mobile applications. This fragmentation leads to several issues:

Efficiency in Recipe Management: The Number one pain created by the current Social Recipe platform is that whenever a user requires a particular recipe, he often spends a lot of time searching for that recipe in vain.

Difficulty in Customization: Cooking has become tiresome every time a person is forced to change one recipe to another because some ingredients are not available, or one does not want to use some particular cooking ingredient.

Limited Sharing Capabilities: When people want to share recipes with others, especially friends and family, the process is conducted in a traditional way and can be quite a tedious and lengthy one since no integration with other social networks is possible in this case.

Inconsistent User Experience: The tools, if existent, may not necessarily have user-friendly interfaces and, therefore, the organization of recipes may prove unreasonably cumbersome.

Security Concerns: The sharing of personal recipes makes other people having access to the accounts or platforms where the information is being stored creating insecurity.

Scalability Issues: As the number of recipes increases in those applications, many current solutions fail to accommodate them, or they lack versatile options to sort information.

These issues underpin the necessity for an optimal solution that would consolidate the management of recipes into one.

3.0 PROPOSED SOLUTIONS

**1. Centralized Recipe Management:** This assists in the easy arrangement of all recipes in the same place to enable each recipe to be easily accessed when required?

**2. Efficient Recipe Editing:** One of the main advantages is that it is possible to add new recipes in this book and also do modifications to the current recipes in the book and also one can have an option of deleting those recipes which are not wanted thus making the recipe book ever updating to the current.

**3. Advanced Search Functionality:** Functionality can be done through the search with ingredients and other keywords and categories, hence making it easier for users to pick an appropriate recipe for certain occasions.

**4. User-Friendly Interface:** The approach to the application developing, which is considered nowadays, is oriented on the usage, and so the application can be easily used even if the person does not have great computer skills.

**5. Social Sharing Features:** The connection with the most widespread social platforms allows them to share their most loved dishes with friends and other dear ones, thus making the cooking more sociable.

**6. Data Security and Privacy:** Consumers, while utilizing the Recipe Manager, enjoy higher levels of protection for their data, which are shielded from disclosure to other users.

**7. Scalability and Adaptability:** Remember that in the given case the developed application is aimed at recipes’ storage and management, however, the further application’s creating can be based on its users’ responses and the number of recipes.

**4.0 CURRENT BUSINESS PROCESS / WORKFLOW**

This paper also details the various methods and strategies that were used in the information-gathering process for Recipe Manager: Following are the various techniques and instrumentations that have been employed in the collection of this crucial information.

**4.1 METHODS USED**

To gather relevant information, we employed the following methods:

**Interviews:** With the help of interviews carried out on home cooks, specialized cooks, nutritionist and potential consumers some quantitative data were collected as well as qualitative data in the form of interviews**.**

**Surveys and Questionnaires:** circulated to a more extensive population to gather an aggregate of numerical information and user preferences concerning contemporary recipe management solutions and preferred characteristics.

**Observations:** Used in the current practices to assess the current activities or usage rates in the homes as well as in the hotel and restaurant business.

**4.2 SUMMARY FROM METHODS USED**

The following are the methods used and the summary that has been derived from the research.

Interviews: Essentials that arose from interviews with home cooks and persons in the culinary line include the following:

**Customization Requirements:**

Specifically, it is necessary in relation to the problem of the absence of a ready-made platform to note the following points, which were mentioned by the participants: the ability to use various media formats for creating and editing recipes.

**Security and Privacy Concerns:** Concerns for data security and data privacy were prevalent, some in need of adhering to the GDPR or General Data Protection Regulation.

Example: A Target audience A target group consisted of professional chefs, one of whom raised an issue of the confidential nature of recipes and the need to implement protective measures to prevent unauthorized users from accessing such information.

**Surveys and Questionnaires:** Details were collected using a questionnaire which was administered through an online survey; 170 potential users responded to the survey out of which 200 were contacted hence giving us a response rate of 85%. Specifically, the emphasis was on current user experiences with existing platforms as well as what they would like to have in them. Key findings included:

**User-Friendly Interface:**

Specifically, the overwhelming majority of the surveyed audience – 78% – reported that ease of use was essential to the success of the platform.

**Advanced Features:**

Respondents wanted newer features and had an interest of making enhancements such as: Search, Categorization, and easy Edit capabilities had an interest of by 80% of the respondents.

**Example:** As one home cook stated, “Currently I use my hand to support my tool and the model I currently work with is not easy to use; it requires more interfaces that make it easier to switch through the recipes without having to be trained. ”

**Observations:**

Observations in various kitchen environments revealed several insights:

**Efficiency Challenges:**

* + Traditional methods of managing recipes, such as handwritten notes or disparate digital files, are often cumbersome and time-consuming.
  + Users struggle with organizing recipes into coherent categories, which makes retrieval difficult and inefficient.
  + The lack of standardization across various recipe formats (e.g., text files, images, PDFs) leads to additional time spent on formatting and categorization.
  + **Example:** In one observed home kitchen, a user spent significant time searching through a cluttered drawer of handwritten recipes, highlighting the inefficiency of manual organization. Another instance involved a professional chef who found it cumbersome to convert various recipe formats into a unified digital system, indicating the need for seamless integration and categorization features.

1. **Remote Access Needs:**
   * The necessity for a platform that supports remote access was evident, enabling users to manage and access recipes from different locations, whether they are at home, in a professional kitchen, or traveling.
   * Users expressed the need for synchronization across devices, ensuring that any changes made on one device are reflected on all others.
   * Remote access also facilitates collaboration among users, such as sharing recipes with family members or colleagues in a professional setting.
   * **Example:** Observations in a culinary school showed that instructors required a way to access and update recipes from different classrooms and even from home. Another scenario involved a traveling chef who needed to access his recipe database on-the-go, underscoring the importance of a cloud-based solution that offers reliable and secure remote access.

**5.0 LOGICAL DFD**

**5.1 Existing business process map Comparative Business Process (AS-IS)**

**Scenarios and Workflow**

There are several scenarios in the current business process for managing recipes and all of these depict how users are disadvantaged and faced with a lot of problems. These include:

**Scenario 1: Multiplex Mess Recipe Storage**

* **Workflow:** Recipes are stored in many different forms and places including hand-written notes, a physical cookbook collected or bookmarked websites, and electronically on different devices.
* **Challenges:**
  + Managing and finding recipes is a nightmare.
  + It is tedious to search for a specific recipe in the traditional way.
  + This means that there is no centralized system to create all the recipes for use in the company.

**Scenario 2: Improper editing and sorting**

* **Workflow:** Currently, raw and compiled recipes go through manual post-processing and tagging with basic tools such as word processors or folders.
* **Challenges:**
  + There are some drawbacks of using this design: little flexibility for editing and categorizing.
  + Time-consuming and error-prone process.
  + They also identify inconsistent categorization as another route to further disorganization.

**Scenario 3: Lack of proper Search Engine features**

* Workflow: People try to find recipes using information relevant to them from different sources by a word of mouth.
* **Challenges:**
  + Lack of efficient and less time consuming search strategies for information is another key issue.
  + The problem of sorting recipes depending on certain criteria such as certain type of ingredient or banned ones, specific type of food preferences etc. .
  + Some of the users reported dissatisfaction caused by the absence of new search options.

**Scenario 4: Remote access and synchronization**

One of the key areas that are not properly implemented in the current system is the lack of remote access and synchronization.

* **Workflow:** Users open recipes from multiple platforms, and there is no primary hub they go via.
* **Challenges:**
  + No consistent update of recipes in recipe management systems.
  + One of the disadvantages of using cloud storage is that the changes made in the cloud are not reflected in all the devices immediately.
  + Issues, for example, of cooking or sharing recipes with other people at different locations.

**Functional Requirements (AS-IS)**

**Input:**

* A recipe can include information about ingredients, steps, categories they are under and any media relevant to it.
* Based on the collected data: User preferences and the text of search queries.

**Process:**

* Storing and categorizing recipes.
* Making recipe adjustments and modifications of recipe information.
* Procedure of selecting and sorting options depending on the client’s preferences.
* Another factor involve in affecting backup strategies is synchronizing data across devices for remote access.

**Output:**

* Together it makes for an organized and accessible collection of recipes.
* Important information to display when a user performs a search and the recipes that are related to it.
* They include, recipe update notifications, and activity notifications from the community.

**Non-Functional Requirements (AS-IS)**

**Performance:**

* There is no delay in any section or tool of the program, and transitions seem smooth and fast.
* Flexibility in managing big recipe databases.

**Control:**

* Authorization by individual identification of users in order to provide access to the system.
* Protection of the personal information that a company or an organization handles.
* Backup solutions and data preservation protocols: Backup as an activity that requires implementation on a regular basis.

**Usability:**

* Intuitive and user-friendly interface.
* The usability for common users and the web-design that will not require any additional skills from the users.

**Scalability:**

* Scalability to accommodate additional features or additional traffic from users as might become necessary in the future.
* Design for modularity to extend it to embrace other technologies and services.

**5.2. DFD (AS-IS) System Logical Overview**

1. **Context Diagram:**

Shown below is the use case diagram of the Recipe Manager system that shows user interaction with the system and the primary input and output data that is exchanged between the user and the system in terms of recipes and search queries.

* Key Interactions:
  + Users: Enter the recipes, set the search terms, and then output the search results.
  + System: An input process that stores recipe data and provides an output to the given queries entered by the user.

1. Diagram 0:
   * Recipe Storage: The procedure of sorting and selection of the recipes with the help of users.
   * Recipe Editing: Any changes or enhancements regarding the considered recipe details.
   * Recipe Search: As it will be defined below, the process of searching and selection of the recipes according to the criteria set by the user.
   * Data Synchronization: Coordinating the data of recipes through the means of synchronization in order to enable access from all kinds of devices and constantly updating the recipe information.

**Child Diagrams:**

* Search Algorithm: Explaining how filtering and retrieval of recipes by queries occur, the process in detail.
* Data Encryption and Security: This section explains how the data behind recipes and the users are protected from unauthorized access and potential misuse.
* Synchronization Process: Explaining the manner in which data ensures synchronization when used on various devices.

**Link to Information Gathering**

* Efficiency Challenges: The ineffectiveness in the traditional management of recipes has informed recommendations for the improvement of the work processes and search functions.
* Remote Access Needs: Due to the envisioned need to access and work on recipes regardless of the location, the system integrates the concept of remote access and synchronization.
* Customization and Security Concerns: Both the functional and non-functional requirements of the platform, therefore, bear the need for flexible recipe creation and security as key considerations in meeting the expectations of the users.

**5.3 NON-FUNCTIONAL REQUIREMENT (PERFORMANCE AND**

**CONTROL)**

Performance Requirements

* Response Time:

The identified objective involves fulfilling up to 90% of queries within 2 seconds of a search performed in the system concerning recipes. Moreover, the system should then guarantee that all page transitions are done within a half a second.

* Scalability:

This means that concurrent usage will not significantly affect the desired performance of the application and should therefore support up to 10,000 users.

To achieve this, the database of the systems shall contain up to 1 million recipes in such a way that the search performance of the application shall not degrade more than 5%.

* Throughput:

During the business hours of the day, it shall be able to handle and store at least 1000 new recipes per minute.

* Resource Utilization:

The system shall not operate at maximum CPU utilization of more than 70% under the working load and not more than 90% during the peak hours.

Maximum permitted operating memory usage shall be 75 percent of the total capacity when operating in normal mode.

* Data Transfer Rate:

The system shall be capable of handling data transfer rates not exceeding 500 MB per second where data uploads and downloads are massive.

**Control Requirements**

* Authentication and Authorization:

The system shall have adequate security measures through the use of RBAC so that only authorized personnel can interact with the management functions.

Essentially, ALL users have to authenticate in a secure way, preferably, OAuth 2. 0 or Multi-Factor Authentication (MFA) is a combination of two or more security measures used for verifying the identity of a user with the purpose to enhance the protection against unauthorized persons.

* Data Integrity:

Data integrity shall be maintained for recipe data in transmission and storage by checksums incorporated into the system.

They will need to perform regular data backups to ensure data is not lost, they will need to set the recovery time objective (RTO) at 4 hours and recovery point objective (RPO) at 30 minutes.

* Audit and Logging:

It shall be useful to include the time stamp for all activities done in the system, such as creating, editing, or erasing a recipe, among others.

Records should be stored for at least 1 year and be kept at a secured location for audit purposes.

* Compliance:

The system must align with standard legal requirements concerning personal information processing such as GDPR, CCPA, among others, to protect the privacy of users and their data.

Annual compliance audits are to be conducted as a part of compliance reporting on a regular basis.

**5.4 LOGICAL DFD AS-IS SYSTEM (CONTEXT DIAGRAM, DIAGRAM 0, CHILD)**

* **Context Diagram:**

**A diagram of a software company

Description automatically generated**

* **Diagram 0:**

**A diagram of a software company

Description automatically generated**

**Completion:**

**A diagram of a software application

Description automatically generated**

**Child Diagrams:**

**1.**

A diagram of a data flow

Description automatically generated

**2.**

**A diagram of a flowchart

Description automatically generated**

**3.A diagram of a software system

Description automatically generated**

**4.**

**A diagram of a software company

Description automatically generated with medium confidence**

**5.**

**A diagram of a process

Description automatically generated6.A diagram of a data flow

Description automatically generated**

**6.0 System Analysis and Specification:**

**1. Introduction:**

Food Recipe Manager is a system with an objective to let its users manage recipes, create shopping lists, and suggest meals according to the users’ preferences, this system is designed to make the management of meals easier so that the user can easily schedule the culinary operations.

**2. Objectives:**

* User-Friendly Interface: Ensure that the application offers basic and easy to navigate graphics for using the recipes database and the shopping list.
* Recipe Management: Allow the users to create, modify, and delete recipes and let them search for recipes.
* Personalized Recommendations: Provide meals that could be suggested based on the users’ profile and their previous actions.
* Shopping List Generation: Able to create shopping lists for the selected recipes.
* Integration with External Databases: Complete recipes from other databases are to be included into the recipe list.

**3. Scope:**

* User profile management
* Recipe management
* Shopping list generation
* Recommendations for meals
* Integration with external recipe databases for additional recipes

**4. Functional Requirements:**

4.1 Manage User Profiles:

* Create Profile: Users should be able to add a new profile for their preferences.
* Update Profile: User can carry out a profile update.
* Delete Profile: It gives the authority to the users to delete their profile and every information that is linked to it.

4.2 Manage Recipes:

* Add Recipe: It can be also used to store new recipes that have been found.
* Update Recipe: It also allows the users to modify existing recipes.
* Delete Recipe: As for the final user, concerning the operation of the web site, the following points should be noted: Users can delete recipes from their collection.
* Fetch Recipe: The system also has the capability to search recipes from an offsite recipe database.

4.3 Generate Shopping List:

* Create Shopping List: Create a shopping list of the different products required depending on the chosen recipes.
* Update Shopping List: It includes the capacity for the users to modify the pattern of the provided shopping list (append or remove items).
* Delete Shopping List: Perhaps one of the most noticeable observations is that a user can delete a shopping list.

4.4 Provide Recommendations:

* Create Shopping List: Create a shopping list of the different products required depending on the chosen recipes.
* Update Shopping List: It includes the capacity for the users to modify the pattern of the provided shopping list (append or remove items).
* Delete Shopping List: Perhaps one of the most noticeable observations is that a user can delete a shopping list.

**5. Non-Functional Requirements**:

* Performance: The system should be able to support numerous users at a go, at slightly reduced efficiency as compared to a smaller scale system.
* Usability: Third, the development of the interface should be as easy as possible and understandable for a non IT-specialist.
* Reliability: Pursue the data integrity and availability with little disruption time.
* Security: The program should safeguard user information and provide appropriate access to the users’ account, recipes, and other information.
* Scalability: Through features such as the use of a database, should be expandable, as the number of users increases, and the number of recipes inserted.

**6.1 Logical DFD TO-BE system (Context Diagram, Diagram 0, Child):**

Context Diagram:

A diagram of a food recipe system

Description automatically generated

Diagram 0:

A diagram of a software company

Description automatically generated

Child Diagram:

A diagram of a user profile

Description automatically generated

**6.2 Process Specification (based on Logical DFD TO-BE):**

**1. Manage User Profiles (1.0)**

1.1 Create User Profile:

The intended use of this process is for a new user to create a profile and input their details and preferences. The process starts with the collection of input information by the user including name, email, password, and choice of diets. The system then checks the input entered by the user and conforms to check if all the fields that are supposed to be filled with data have been filled by the user or not and whether the data entered is correct or not. If the validation is completed, the profile data is stored in the User Data Store. Such prerequisites are the user to whom this process will be offered is not registered in the system. The post condition, therefore, of this process is that a new user profile is created and saved in the system.

1.2 Update User Profile:

This activity enables the existing users to change or fill in some of their details in their profiles. The user enters the changes in the profile information which can be the modifying of the diet or changing of the contact information. It checks on the new data that has been entered in the system so that one can keep on confirming that all is well. If the data validation is successful, then the new profile data created in the client tier are written into the User Data Store. Web system preconditions are user logged in and user have profile. The postcondition is that the user profile stored in the system is moderated with the new information that is given.

1.3 Delete User Profile:

This activity enables the existing users to change or fill in some of their details in their profiles. The user enters the changes in the profile information which can be the modifying of the diet or changing of the contact information. It checks on the new data that has been entered in the system so that one can keep on confirming that all is well. If the data validation is successful, then the new profile data created in the client tier are written into the User Data Store. Web system preconditions are user logged in and user have profile. The postcondition is that the user profile stored in the system is moderated with the new information that is given.

**2. Manage Recipes (2.0)**

2.1 Add Recipe:

This process lets users add new recipes in the collection. The user enters quantitative information about the recipe: products list, steps in preparation and cooking, and the nutritional values. The system checks the recipe data for compliance with the specific information that is required and for correctness. Subsequently, the recipe is stored in the Recipe Data Store after validation is provided on the recipe. The given process is only possible provided that the user is logged in the system. The postcondition is that new recipe has been saved in the collection of the user and filed in the system.

2.2 Update Recipe:

This serves the purpose of letting the users modify recipes that are already in their database. The user provides new information regarding the recipe like new additions or deletions, to the existing recipe. The system checks the updated data both for its compliance with the pre-set standards as well as the various requirements. The modified recipe is the stored in the Recipe Data Store if the validation is successful. Logically, preconditions are the user being logged in and the recipe existing in the system. The postcondition is that the key details of a recipe located in the Recipe Data Store are modified with the new information

2.3 Delete Recipe:

This process enables one to remove recipes from a list of stored recipes. The process begins with the user inquiring about the removal of a given recipe. However, to eliminate the chances of accidental deletion, the system prompts the operation to confirm the deletion request. On confirmation, the system deletes the recipe from the Recipe Data Store of the given system. The precondition to this is that the user must authentic and the receipt must exist in the database. The post condition is that the recipe is dumped permanently from the customer list of the system.

2.4 Fetch Recipe:

The rationale of this process is to allow the system to obtain recipes from an external recipe basis. The user enters a recipe query and this is passed to the external recipe database by the system. The operation external DB passes the recipe data to the Recipe Data Store and reveals the recipe to the user. The precondition is that the user must be logged in but it easily overcame with the context of the application. The postcondition is that after the described actions the recipe data is ready to be given to the user and is also saved in the system.

**3. Generate Shopping List (3.0)**

3.1 Create Shopping List:

This process creates a shopping list according to the chosen recipes. In products that allow for multiple potential meals to be created, the user chooses which of these recipes should be created for the current shopping list. The next step in the flow of this system entails the generation of a list of ingredients necessary for the chosen recipes. This compiled shopping list is stored and saved under the Shopping List Data Store then shown to the convenience of the user. The prerequisite for this process is the fact that the user must be authorized and has made choice of recipes. The postcondition is that there is a new list of stores generated and saved in the system to be used by the user.

3.2 Update Shopping List:

This process enables the user to modify an existing shopping list. The user modifies the shopping list through the input of the new and deletions of items that exist on the list. It ensures that the updates entered are formulated correctly and that they are reasonable. If the validation is positive, the new shopping list as updated is stored into the Shopping List Data Store. Such conditions are the availability of the shopping list and the user being logged in to her/his account. The postcondition is that the attribute shopping list is modified with the new details entered by the user.

**3.3 Delete Shopping List:**

This process enables a user to erase a shopping list. The user deletes a certain shopping list, and the system asks him if he really wants to delete that component or list. On confirmation, it removes the shopping list from the Shopping List Data Store to ensure the list is no longer available. The precondition here is that the user must be logged in and that the shopping list that the user is trying to modify must be present. The postcondition is that the shopping list is deleted from the system permanently so that there is no stored information that could be retrieved finally.

**4. Provide Recommendations (4.0)**

4.1 Generate Recommendations:

The purpose of this process is to provide personalized meal recommendations based on user preferences and past behavior. The system analyzes the user's preferences and past recipe data to generate these recommendations. The recommendations are then displayed to the user, helping them make informed decisions about their meals. The precondition is that the user must be logged in. The postcondition is that the user receives personalized meal recommendations tailored to their preferences and dietary needs.

4.2 Provide Nutritional Information:

This entails that information on nutritional content is incorporated in the food choices highlighted whenever this is possible. The nutritional data for the recipes being recommended is fetched from the Recipe Data Store of the system. This information is then fed into the recommendations that are given to the user thus offering them vital information concerning the calorific value of the foods that they are consuming. The precondition is the Nutritional Information must be provided on the recipes. The post condition is that the user will be recommended meals and nutritional advice in order to make the right choices.

**Partitioning:**

Recipe Manager

User Interface

Recipe Management

Recipe Display Interface

Recipe Input Interface

Recipe Search Interface

Recipe Modification

Recipe Storage

Recipe Retrieval

Auxiliary Services

User Management

Meal Planner

Nutritional Analysis

Shopping List Generator

User Profiles

User Authentication

**Diagram 0:**

**A diagram of a software company

Description automatically generated**

**Completion:**

**A diagram of a software application

Description automatically generated**

**Child Diagrams:**

**1.**

**A diagram of a data flow

Description automatically generated**

**2.**

**A diagram of a flowchart

Description automatically generated**

**3.**

**A diagram of a software system

Description automatically generated**

**y**

**4.**

**A diagram of a software company

Description automatically generated with medium confidence**

**7.0 Physical System Design**

**1 Introduction**

The details of the physical system design guarantee the use and running of the “Food Recipe Manager” as efficient, secure, and as scalable as desired. This section describes the hardware requirement and software, network as well as the architectural requirement.

**2 Hardware Requirements**

Servers: The high-performance servers are required for the database, website hosting and application processing needs.

Workstations: For administrative and analytical purposes, and other large scale computations which are vital for system maintenance and data analysis.

Storage: SSDs or NAS for Archived data of large volume recipes, users, and logs.

Network Equipment: Raised for handling the traffic in the laptop, such as the routers, switches and the firewalls.

**3 Software Requirements**

Operating Systems: For servers, Linux OS is recommended while for workstation one may use Windows OS or macOS.

Database Management System: My SQL or PostgreSQL to store recipes, users, and shopping list information.

Application Software: Server-side scripting languages applied in the middle tier (e. g. JAVA programming language, PHP), Middleware and web server software (e. g. Apache Tomcat, Nginx).

Security Software: Encryption tools, antivirus, and IDS systems are some of the examples of network security.

**4 Network Design**

Network Topology: Star topology for ease of designing and for easy implementation of the maintenance protocols.

IP Addressing: Well defined IP addressing plan.

Bandwidth Requirements: Sufficient to bear the max loads without any deterioration in functionality.

Network Security: VPN, firewall and secure transference of data.

**5 System Architecture**

Client-Server Architecture: Bayadic strongly supports modularity and scalability as key software constructs.

Cloud Integration: The utilization of the cloud services to get more storage space and back up services in case of a disaster.

Data Flow: There is the real-time update and response via the web interface.

**6 Physical Layout**

Server Room Layout: Air cooling and management, especially of cables, as well as protective arrangements.

Workstation Layout: Improve work flow and gaining access to difficult hardware.

Network Cabling: Structured cabling for reliability to perform diagnostics on the network.

**7 Redundancy and Failover**

Backup Systems: Automated backup at fixed intervals or daily, fortnightly, monthly and so on kept in offsite or cloud.

Failover Mechanisms: Switching on to back servers immediately to avoid elongated server downtime.

**8 Security Measures**

Access Control: Smart cards for main hardware, locks keys, and other more essential simultaneous equipment.

Surveillance: Cook Security in computer rooms and any area considered important.

Disaster Recovery: Plan for bringing back the delivery of services as well as data after a disaster.

**9 Scalability Considerations**

Hardware Scalability: Suggesting that many of the structural components should be ‘modularized’ so that changes could be made easily as and when required.

Network Scalability: Support for additional devices and traffic can be defined as a crucial element of the platform since it helps to provide the necessary client-server connections and ensure smooth running of the services for customers.

Storage Scalability: Other forms of storage solutions such as outsourced variants such as the cloud solutions.

**Crud Matrix:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Activity | User | Recipe Storage | Recipe Editing | Recipe Search | Social integration | Data security % privacy |
| User logon | R |  |  |  |  |  |
| User inquiry | R |  |  |  |  |  |
| Recipe creation |  | C | C |  |  |  |
| Recipe modification |  |  | U |  |  |  |
| Recipe deletion |  |  | D |  |  |  |
| Recipe Search |  |  | D | R |  |  |
| Social Sharing |  |  |  |  | C |  |
| Privacy settings update |  |  |  |  |  | U |
| Data encryption |  |  |  |  |  | C |
| User account deletion | D |  |  |  |  |  |

**Event Response table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Event** | **Trigger** | **Source** | **Activity** | **Response** |
| User logs in | User submits login details | User | Authenticate user | System verifies credentials and grants access to the user's dashboard. |
| User inquires about a recipe | User clicks on recipe link | User | Fetch recipe details | System retrieves and displays the recipe details. |
| User creates a new recipe | User submits recipe form | User | Store new recipe | System saves the new recipe in the database and confirms the creation. |
| User modifies an existing recipe | User submits updated recipe | User | Update recipe | System updates the recipe in the database and confirms the update. |
| User deletes a recipe | User confirms deletion | User | Remove recipe | System deletes the recipe from the database and confirms the deletion. |
| User searches for recipes | User enters search criteria | User | Search recipes | System searches the database and displays matching recipes. |
| User shares a recipe | User clicks share button | User | Share recipe on social media | System posts the recipe to the selected social media platform. |
| User updates privacy settings | User changes settings | User | Update privacy settings | System updates the user's privacy settings in the database. |
| Data encryption check | System operation | User | Encrypt data | System ensures that all sensitive data is encrypted. |
| User deletes their account | User confirms account deletion | User | Delete user account | System deletes the user's account and all associated data. |

**Structure chart:**

**A diagram of a company

Description automatically generated**

**System architecture:**

A diagram of a software application

Description automatically generated

**8.0 SYSTEM WIREFRAME ( I/O DESIGN )**

[**https://www.figma.com/design/iQarIItTB43bHb6XqMX8bG/SAD-RECIPE-MANAGER-PROTOTYPE?node-id=0-1&t=Zvu29JVXwjeang3l-1**](https://www.figma.com/design/iQarIItTB43bHb6XqMX8bG/SAD-RECIPE-MANAGER-PROTOTYPE?node-id=0-1&t=Zvu29JVXwjeang3l-1)

**9.0 SUMMARY OF THE PROPOSED SYSTEM**

**9.1 Recipe Manager System**

The Recipe Manager System is a complex solution whose aim is to tackle several issues which can be associated with the management of recipes originating from different sources. The instance the client becomes more inclined towards cookery and various types of recipes available around, the demand increases for the application which helps the client in managing the recipes. This system aims at enhancing the way of cooking by establishing a system that categorizes recipes which is nested, editable, searchable, and sharable.

**9.2 Key Features and Functionalities**

**Recipe Storage and Management:**

**Create, Modify, and Delete Recipes:** Users have no problem in entering new record, change any record they want and remove any record that the user does not want.

Search Functionality: The multi-stage search is integrated into the system so that, depending on the ingredients, kind of dish or any other aspect, the target recipes can be searched easily.

**User Interaction and Social Integration:** It means one has to learn how to interact with users and how such interaction enhances social integration.

**User Login and Authentication:** The concept of user authentication for right people and the privacy of personal recipes and data are untouchable to the outsiders.

**Social Sharing:** The application allows a user to directly share recipes of their desired ones on all the various social platforms.

**Customization and Flexibility:**

**Recipe Customization:** Through it, users can change the proportions in the recipes in the system relative to available stocks or the user’s default setting that has been programmed into the system.

**Shopping List Generation:** This is functional in that users can opt to have list of shopping according to type of recipes they selected they can edit /include/exclude it.

**Security and Privacy:**

**Data Encryption:** This ensures that data is protected which sometimes is sensitive and needs protection from unauthorized persons.

**Privacy Settings:** The people have the ability to make their recipe posts private or set privacy on their post hence restricting the number of people that can see certain recipes.

**Scalability and Performance:**

**Hardware and Network Scalability:** The webpage is designed in a way that it is capable of handling a thousands of users at the same time and at the same time it shall be capable of handling a data base of thousands of recipes.

**Remote Access:** The users can always be able to retrieve their recipes irrespective of the location or the device used making it more easier to use.

**9.3 Non-Functional Requirements**

**Performance:** It presently, as well as in the future, can also accommodate any 10 000 active users; however, the site is planned for a database of up to one million recipes. It is thus necessary that at least ninety percent of the level of searching should be accomplished in no less than 2 seconds, and that switching of the pages should not take more than a half of a second.

**Usability:** This means that all interfaces should be rather resistant and able to fit any profile and especially for IT oriented workers.

**Reliability:** Accordingly, it gives very high grade of data security and at the same time has positive impact on the availability, because its changes, for instance the software, are not very often required.

**Compliance:** Regarding data management and data privacy it also complies with the GDPR and CCPA legislation.

**9.4 Conclusion**

At the moment, it is possible to state that the Recipe Manager System can be rightfully referred to as rather an idea on the development of which one can still witness the rudimentary stage of creation of the idea, which will alter consumers’ perception of recipes. That is why, the application in a proper way, puts in oblivion all the troubles associated with the traditional methods of recipes usage, because the application unites all the procedures connected with them. Therefore the themes are: While experiencing relief in preparation and while cooking and the safety and the scalability and the performance of the system who guarantees that its relief assist the home ware cooks and the professional cooks.