

**SCHOOL OF BUSINESS AND ECONOMICS**

**DEPARTMENT OF BUSINESS TECHNOLOGY**

**SYSTEM ENGINEERING**

**Project Name:**

**G.Kings Recipe recommendations system**

**Student name: SHYAKA Crispin**

**Reg No: 222004852**

**Submitted to: Dr. BUGINGO Emmanuel**

**Date: 10/7 2023**

# Table of contents

Contents

[Table of contents 2](#_Toc137559274)

[List of Figures 4](#_Toc137559275)

[List of tables 5](#_Toc137559276)

[1. Introduction 6](#_Toc137559277)

[2. Statement of the problem 7](#_Toc137559278)

[2.1 Description of the existing system 7](#_Toc137559279)

[2.2 Problems of the current system 7](#_Toc137559280)

[2.3 How the proposed system will work 7](#_Toc137559281)

[2.4 Describe other alternatives 7](#_Toc137559282)

[3. Software requirement 7](#_Toc137559283)

[3.1 User Requirements 7](#_Toc137559284)

[3.1.1 Functional requirement 7](#_Toc137559285)

[3.1.2 Non-function requirement 7](#_Toc137559286)

[Portability requirement 7](#_Toc137559287)

[Reliability requirement 7](#_Toc137559288)

[Usability requirement 7](#_Toc137559289)

[Space requirement 7](#_Toc137559290)

[Organization requirement 7](#_Toc137559291)

[Implementation requirement 7](#_Toc137559292)

[External environment requirement 7](#_Toc137559293)

[Privacy requirement 7](#_Toc137559294)

[Safety requirement 7](#_Toc137559295)

[3.2 System requirements 7](#_Toc137559296)

[3.2.1 Minimum End-user Hardware Requirements 7](#_Toc137559297)

[3.2.2 Minimum End-user Software Requirements 7](#_Toc137559298)

[3.3 Software specifications 7](#_Toc137559299)

[**4.** DATA FLOW DIAGRAM (LEVEL 0, LEVEL 1**)** 8](#_Toc137559300)

[4.1 DFD LEVEL 0 8](#_Toc137559301)

[4.2 DFD LEVEL 1 9](#_Toc137559302)

[5. Feasibility study 9](#_Toc137559303)

[5.1 Technical Feasibility 9](#_Toc137559304)

[5.2 Financial Feasibility 9](#_Toc137559305)

[5.3 Market Feasibility 10](#_Toc137559306)

[5.4 Economic Feasibility 10](#_Toc137559307)

[5.5 Legal and Regulatory Feasibility 10](#_Toc137559308)

[5.6 Operational Feasibility 10](#_Toc137559309)

[6. Description of the selected Process Model 10](#_Toc137559310)

[6.1 Cause of your selection 10](#_Toc137559311)

[7. Conclusion 10](#_Toc137559312)

[8. References 10](#_Toc137559313)

# List of Figures

[Figure1.Types of software requirements 8](#_Toc140416920)

[Figure2 types of software 18](#_Toc140416922)

[Figure3 dfd (level0) 19](#_Toc140416923)

[Figure4 dfd(level1) of the system 20](#_Toc140416924)

# List of tables

[Table1 software and hardware requirements for the system 15](#_Toc140416541)

[Table2 different browser to used 17](#_Toc140416544)

# 1.Introduction

# Are you tired of looking at Aton of the ingredients and feeling lost what to cook, Are you ready to turn your ingredients into an adventure of cooking, are you tired of staring at your ingredients looting in the fridge, wondering what to cook on your next meal? Do you find yourself in the same field preparing the same dishes continuously, looking for a lot of culinary inspirations or are you looking for a way of cooking delicious meal using the only ingredients you have on hand? Say hello to the our recipe recommendation system because No more worries our G.Kings recipe recommendation system got you covered and is here to turn your existing ingredients into an amazing well looking and designed dish and a culinary masterpiece than what you’ve even seen before and it is here to transform your available ingredients into a delightful and amazing recipe and it is here to transform the way you cook, you serve as well as the way you eat

# G.Kings- system is a recipe recommendation system that helps you to become a maestro in culinary activities with the ingredients that you only have on your hand. No more stresses in searching recipes or your ingredients wastage.

# 2.Statement of the problem

## 2.1 Description of the existing system

The existing system was that which allowed the user to search for the recipe and provides the variety of existing recipe recommendations and the user has to choose what is suitable for him regarding the ingredients he has

## 2.2 Problems of the current system

The existing system didn’t put in consideration the user’singredients which resulted into the consumption of a lot of time as the user had to choose among a variety of recipes that have been provided.

## 2.3 How G.Kings- system will work

1. Ingredients inserting:

Simply input your ingredients that you have on your hand either those which are fresh or those that are pantry staples it doesn’t matter.

1. Analysis:

Our system algorithm will analyze your provided ingredients, considering there compatibility, texture as well as their flavors.

1. Recipe provision:

in the a matter of moment, our system will immediately provide the list of recipe option that precise the match of your ingredients that you’ve provided above and allows you to choose your perfect recipe

## 2.3.4 Describe other alternatives

For the other alternatives we may have the following

**Collaborative filtering with matrix:**

This approach would recommend the recipes basing on the preference and the choices for the similar users. And it will require the creation of a matrix to find the patterns within the users and the items so use to suggest the recipe

**Content based filtering:**

Within this approach the recipes will be recommended basing on the similarities among the ingredients provided by the user and those on the lists of the existing recipes

**Hybrid approach :**

By here these approaches (content based filtering and collaborative filtering) can be combined such they can be used to provide more accurate and a lot of recipe recommendation diversity.

## 3. Software requirement

The software requirements for G.Kings system are the specific functionalities and capabilities that the software must provide to meet the system’s objective that comes to outline the behaviors and the features that the software should provide.

The software requirements defines the core functionalities and features of this system aiming to deliver a user friendly, efficient and reliable platform for the recipe recommendations and based on the users ingredients provided . Some of the requirements of software are:

System must enable the users to input the available ingredients for recipe recommendations, and the system must process the user provided ingredients to identify the relevant recipes. The system after must handle the ingredient variations to improve accuracy

System must generate recipe recommendations based on the provided ingredients. It must consider the user preference, such as the restrictions; the system should offer multiple recipe options with the details instructions regarding to that recipe.

Responsive user interface: this system will be able as it will be having a user friendly and responsible interface accessible across multiple devices.

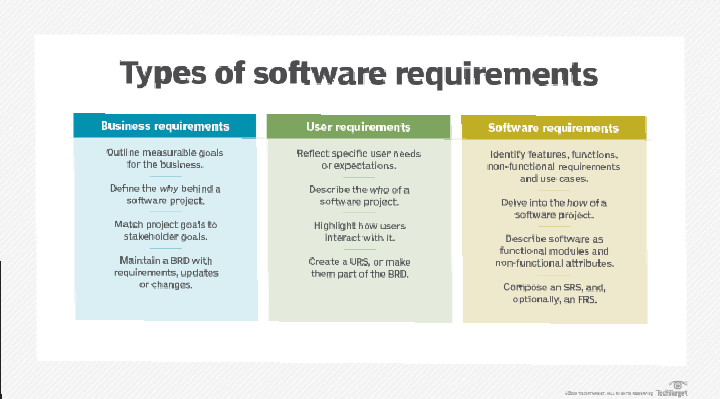


Figure0‑1.Types of software requirements

## 3.1. User Requirements

The user requirements describes the expectations and the needs of the users ensuring that the system caters to their preference by understanding the user requirements is way of building a system that is able to meet the users expectations and encourages its adaption within the users

By addressing these requirements our system can deliver a satisfying and personalized experience to its users, increasing user engagement and retention .some of the user requirements for this system are:

User friendly interface: an interface which is clear and easy to interpret by any kind of user

Ingredient input flexibility: system should accommodate or host various ways the user can express his ingredients

Accurate recipe recommendations:

Recipe variety

Enough information detailed on the recipe

Rating and reviews on the system’s workings and recipe recommendations provided by this system.

Fast response time: this system must be able to provide quick response on the ingredient inputs and recipe searches in other words it must minimize the waiting time

### 3.1.1 Functional requirement

**Ingredient input:** user should be allowed and able to insert the list of the ingredients within a user-friendly interface

**Recommendation of the recipe:** the system should be able to analyze and provide the list of all relevant recipes that are resulting from those ingredients provided by the user

**Sorting:** the user should have the option of sorting the list of recipes according to some criteria like: dietary preference, cooking time

**Special meal:** the system should be able to help those with the wish of special diet and be able to get the suitable recipe by considering the restrictions of the diet.

**Recipe details:** each of the recommended recipe should be detailed with the information of including ingredients, time of cooking , size to be served and the step by step preparation of the dish

**User registration:** the system should allow the users to register and be able to create their profile such that they can save their recipes on the platform

**User feedback and rating:**  the system should allow the users to rate the recipes, view ratings and comments of the other users and be to provide a feedback on the participation of the system

**Recipe sharing:** the users should have the ability to share their favorite recipe on the social media platforms

**Substitution of the ingredients:**  once some of the ingredients are not available within the recipe the system should suggest the suitable substitute to satisfy the users preferences

### 3.1.2 Non-function requirement

### Portability requirement

This system should be designed such that it can work across various devices and operating systems so as to ensure the portability and accessibility for the users easily this system focuses on ensuring that the it can be easily deployed and run on different platforms and environments without significant modifications portability allows the system to be flexible and adoptable and hosting options and user preferences, which also help the administrators of the system. Here are some portability requirements for the system:

Browser compatibility: this system’s user interface should be compatible with popular web browsers such that it will be easy for the users to access it

Mobile device support: the system should responsive and mobile friendly such that those users with mobile device can user it and it is able to adopt different screen size and resolutions

Data portability: this will be able to allow the user to import and export the functionalities of it, and allow the users to transfer their saved recipes or preferences between different platforms of the system

Cross platform compatibility: this system will be able to operate under multiple operating systems which allows the users to access and use it regardless of what kind of operating system they are using

### Reliability requirement

The will have minimal downtime and it will maintain data to ensure that no unexpected failures or disruptions and users can rely on it whenever they need recipe recommendations system instead of seeking for other solutions, reliability is crucial to maintain the user trust and satisfaction. Some of the reliability requirements are:

Data integrity and back: the system should ensure the integrity and accuracy of the user data, preventing data corruption and regular backups should be performed to fight against the data loss and enable data recovery if needed.

Continuous maintenance: system maintenance should be applied to identify and resolve issues proactively and ensuring system reliability and stability

Performance optimization: the system will be designed such that it can handle the user loads efficiently, and ensuring smooth performance during the peak usage periods.

System availability: system must accessible and available to the users for the well acceptable time limits which should aim at high accessibility and availability with a minimum planned maintenance and unplanned outages

Monitoring: the system should be equipped with the tools in charge of monitoring the system health and performance matrices and provide notifications once there have been a situation to the administrators in case of errors of other performance degradations.

### Usability requirement

The user interface should be visual appealing and user-friendly and interacting with the system such that it will be easy for any user to study and explore it without the need of other extra skills and knowledge to use it and it should enable seamless navigations. The usability requirement of this system focus on ensuring that the system is intuitive, and meets the needs of its intended users. It enhances user satisfaction and maximizes the system’s effectiveness. Here are some of the usability requirements that the system must meet:

Clear design: the system should follow the clear patterns and maintain the uniform layout as well as color scheme throughout its application

Responsive and fast performance: this system should be responding promptly to the users and have a minimized loading time and delays in displaying the recipe recommendations and other search results

Help and user guide: the system should have a well prepared help documentation and user guides to help the new user to know how they user and access the system

Accessibility: this system should be designed with accessibility standards in mind ensuring that the users with disabilities can access and user this system

This includes the support assistance technologies like screen navigation and screen reader

Multi language support: the system should be able to support multiple languages allowing the users to use their language of preference in the system for a global user base

Mobile responsiveness: the system’s user interface should be optimized for mobile devices, providing s seamless and user friendly experience on smart phones and tables

User feedback and surveys: the system should provide a mechanism for the users to provide feedback, report issues, and participate in surveys or usability testing. User feedback should be actively collected and incorporated into system improvements.

### Space requirement

The space requirements for this system refer to the storage and infrastructure needs of the system, either physical or digital space required to run and maintain the system; The system’s database and storage should be scalable such that it can accommodate the ever growing collection of recipes and the entire user’s data inserted in the system, the space requirements for this system are:

Database storage: the recipe database will require space to store all the recipe information including the ingredients, instruction and user preferences and this will depend on the number of recipes and the level of details for each recipe

Backup storage: some level of storage space should be left for regular backups of the system’s data to ensure that the data can be recovered in case of system failures and data loss.

And this storage will depend frequently on the size and amount of data being backed up

Infrastructure space: this is the space required to host those equipments like servers, and other hardware components necessary for the system to operate. And this will depend on the scale of the system

File storage: system may require space to store additional files such as images or media related to recipes .the file storage requirements will depend on the size and number of files being stored

Server storage: the system will require server space to store the application code, user data, and other necessary files, and this storage will depend on the size of the recipe database and the expected system growth

System logs: space should be allocated with storage logs, error logs, and audit trails which can be used in case of troubleshooting, system monitoring and for security purpose. The amount of space required will depend on the logging level and duration for which the logs are retained

### Organization requirement

The organization requirements of G.Kings system are the essential to ensure the successful development, implementation and maintenance of the recipe recommendation platform.

These requirements focus on the organizational structure, resources, and processes needed to support the system and some of the organizational requirements:

Skilled development team: the organization will need the experts and experienced development team with software development, machine learning and database management. And working with recipe database to ensure the successful implementation of the G.Kings system

Project management: the organization is responsible for setting the timeline for the development and ensures timely delivery of the system and efficient resource management

System maintenance: the organization is supposed to allocate the resources for the ongoing maintenance, bug fixing, and user support to ensure the system operates smoothly and properly.

Team training: the organization is required to provide some training to its team such that they are able to provide the necessary database skills such that the system will be able to satisfy its users

### Implementation requirement

The implementation requirement for this system will involve the technical aspects and steps which are necessary to bring the system to operate or to life. These are the essential requirements for building a functional and reliable recipe recommendation system

Some of the key requirements are:

Database management: creation of database to store the recipe data, user profile and user interaction. you have to choose the database management system that is able to handle the data requirements of the system

User interface design: you have to develop a user friendly interface to enable users to input ingredients, view recipe recommendations and interact with the system efficiently.

Recipe details display: design a clear and understandable layout to display the details of the recipe, including the ingredients, cooking instructions and preparation time

Testing and quality assurance: conduct through testing, including unit testing,user acceptance testing to identify and fix bugs and ensure system reliability

### External environment requirement

The external requirements of this system refer to the external factors and all dependences that can influence its availability, performance and success. Understanding and addressing these requirements are crucial for ensuring that the system operates effectively in the broader context it is intended to serve. Here are some of the external requirements:

User devices: the system should be compatible with a wide range of device such as desktops, laptops, tablets, smart phones and web browsers to cater diverse user preferences

Data sources: the system should have reliable access to recipe database or sources to provide accurate and up to date recipe recommendations to users.

Competitive landscape: analyzing the competitive landscape will help the system identity opportunities for differentiation and improvement to attract and retain users

Marketing: successful deployment of the system may involve a marketing and outreach efforts attract and engage potential users

Copyright consideration: the system should respect copyright laws and licensing agreements when dealing with recipes and other copyrighted content

User demographics and language: system should be designed to cater to the language preference and culinary interests of the target user demographics

External services: if the system users externals services of analytics , it should ensure seamless reliability of these services

Geographical consideration: if this system targets specific regions it should consider the culture or regional cooking preference in its recipe recommendations such that the users feel considered within the system.

### Privacy requirement

Privacy requirements are crucial for ensuring the protection of the user data and maintaining user trust within the system these privacy requirements helps the system to build a trustworthy and privacy conscious platform, fostering user confidence and promoting responsible data handling practices. Some of these requirements are:

Data minimization: the collection and retail for only the minimum necessary user data required for the function of our system and avoid unnecessary data collection.

Data access control: implement access controls to ensure that only the users and other authorized personnel can have access and handle the user data.

Data deletion: the system must provide the users with option to delete their accounts and associated data from the system once they feel like they no longer need to use that account

Data encryption: all the user data including the personal information and other inputs should be encrypted during transmission and storage to prevent the access from unauthorized personnel.

Secure user authentication: implement secure user authentication methods such as passwords and multi factor authentication to protect user accounts from being used by the unauthorized users

Privacy policy: the system should have a comprehensive and easily accessible privacy policy that clearly outlines how users’ data is collected.

User account verification: implement the mechanism to verify the authenticity of the user account to prevent fake activities such as unauthorized logins

Cookie and tracking policies: if the system uses cookies or tracking mechanisms, inform users about their usage and obtain consent where required by the law

### Safety requirement

These are essential to ensure that users can confidently use the platform without encountering any harm or risks. Safety considerations go beyond data privacy and include aspects related to user well being and the prevention of physical or mental harm, these requirements makes the system promote the safe cooking practices, protect the users from potential harm and promotes user experience that encourages user trust and engagement. Here are some of the safety requirements:

Limitation of liability: the is including the disclaimer that the system provides recipe recommendations for informational purpose only and that users should exercise caution when preparing unfamiliar dishes

Prompt support: offer responsive support to address the safety concerns, or report promptly

Accessibility: ensure that the system is accessible to users with disabilities, hosting the needs for a safe and inclusive user experience

Community reporting: allow the users to report potentially harmful contents or inappropriate recipe provided to them, and establish a system to promptly address such reports.

Recipe accuracy: regularly review and validate recipe details to ensure they are accurate, well tested, and safe such that they will not be harmful to the users.

User input validation: implement input validation to prevent users from entering the wrong and incorrect information or those which are harmful

Allergens and restrictions: the system should clearly highlight allergens and dietary restrictions in the recipe recommendations to prevent users from accessing recipes that may pose health risks.

Safety warnings: including safety warnings and cautions in recipes that involve hazardous cooking techniques or some ingredients

## System requirements

These requirements encompass the technical specifications and capabilities necessary for its successful implementation and operations. These ensure that the system can efficiently handle user request, provide accurate recipe recommendations, and deliver a seamless user experience. These requirements help the system to operate effectively, provide accurate recipe recommendations, and deliver a seamless and user friendly experience to its users. Some of these requirements are:

Internet connectivity: a stable and high speed internet connection is necessary for the system to be accessible to users

Data sources: - ingredients database

-access to reliable recipe database

Machine learning and AI algorithms

User interface: -intuitive design

-responsiveness

Security measure: - data encryption

- Secure user authentication

- Access control

Performance optimization

Documentation

### Testing and quality assurance

### Table1 software and hardware requirements for the system

|  |  |
| --- | --- |
| Software requirements | Hardware requirements |
| 1. HTML 2. CSS 3. Mysql 4. PHP | 1. Server 2. Internet facility 3. Computers 4. mobiles |

### 3.2.1 Minimum End-user Hardware Requirements

The minimum end user requirements for this system depend on the platform and deployment method chosen. Since this system is a web based application, users primarily need a device with an internet connection and a web browser to access the system. Therefore the general minimum requirements for the end users are:

1. Desktop or laptop:

Operating system: windows, marcOS, or Linux

Processor: Intel Core i3 or equivalent

Memory(RAM): 4GB or higher

Storage

Internet connection: higher speed internet connection

1. Mobile devices:

Operating system: IOS or android

Memory:2GB or higher

Storage: sufficient to install and run a web browser

Internet connection: cellular data or Wi-Fi

And are general guidelines the actual hardware may vary regarding on the complexity of the system and the size of recipe database as well as the performance optimization implemented

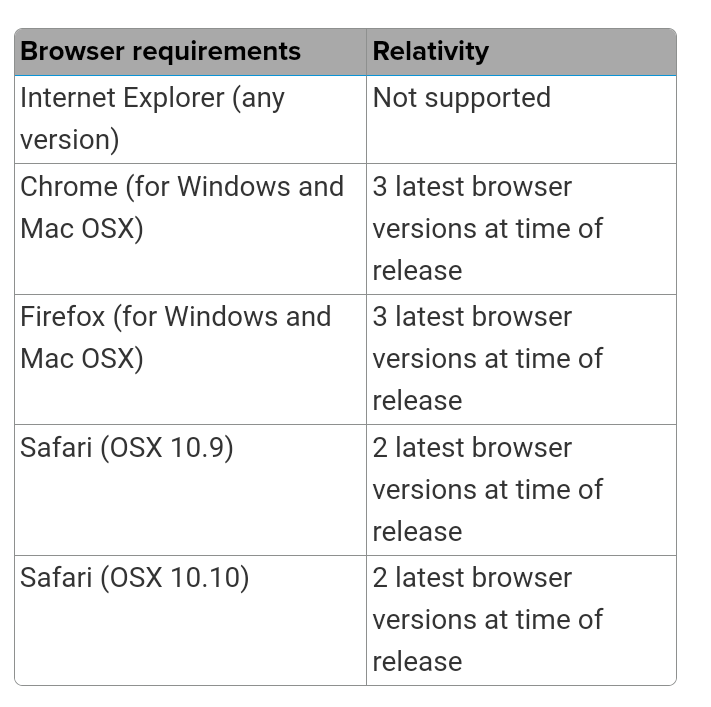
### 

### Minimum End-user Software Requirements

The minimum end user software requirements of the system primarily revolve around having a compatible web browser. Since is web based, users need a modern web browser to access and interact with the system application the minimum software requirement are:

Web browser

### Table2 different browser to used



In addition users are recommended to keep their browsers updated to ensure that security, and they can access the new features of the system once it is updated

Also the system needs to be checked if is able and compatible with the assistance technologies used by individuals with some kind of disabilities

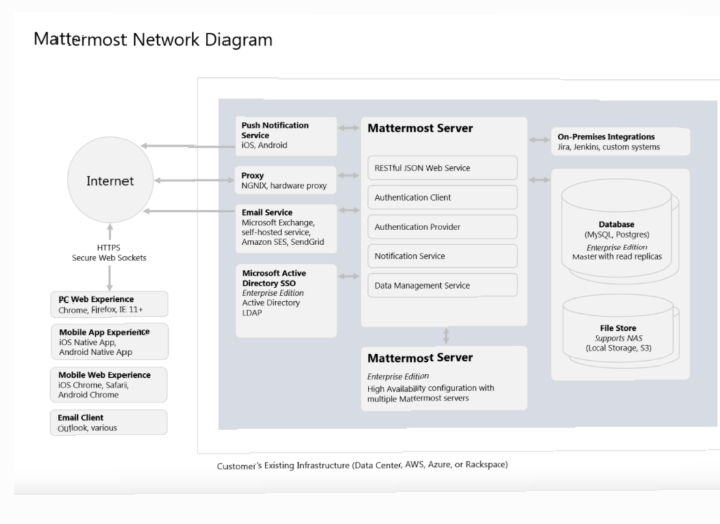


Figure2 types of software

Software specifications

The software specifications for this system provide detailed information about the software components, modules, and technologies required to build and operate the system. These specifications ensure that the software elements are well defined, and aligned with the system requirement some of these specifications for the system are:

Programming language:

HTML/CSS: for building the user interface and frontend components

Web framework:

Flask: this web framework for developing the server side application, managing routes and handling user request

Database management system:

PostgreSQL: a powerful open source relational database for storing recipe data, user profile

User interface design:

HTML/CSS: for designing the layout and styling of the web pages

JavaScript framework: for dynamic and interactive user interfaces.

Search and filtering:

Elastic search: for fast and efficient search functionality

Redis: for caching and optimizing search result

Testing:

Unit testing framework: for testing individual components

Integration testing: to test interactions between different modules

Deployment

Docker: for containerization, allowing easier deployment and scalability

Web server: for serving the web application

Data validation and sanitization:

Validation libraries: for validating user inputs and preventing potential security issues

Table.2different browser to use

# 4.DATA FLOW DIAGRAM (LEVEL 0, LEVEL 1**)**

4.1 DFD LEVEL 0

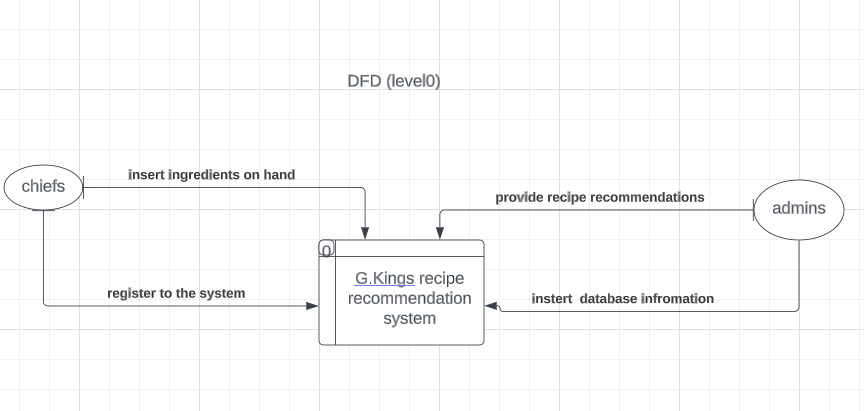


Figure3 dfd (level0)

This figure defines the boundary between the system, or part of a system, and its environment, showing the entities that interact with it, this diagram is a high level view of a system.

## 4.2 DFD LEVEL 1

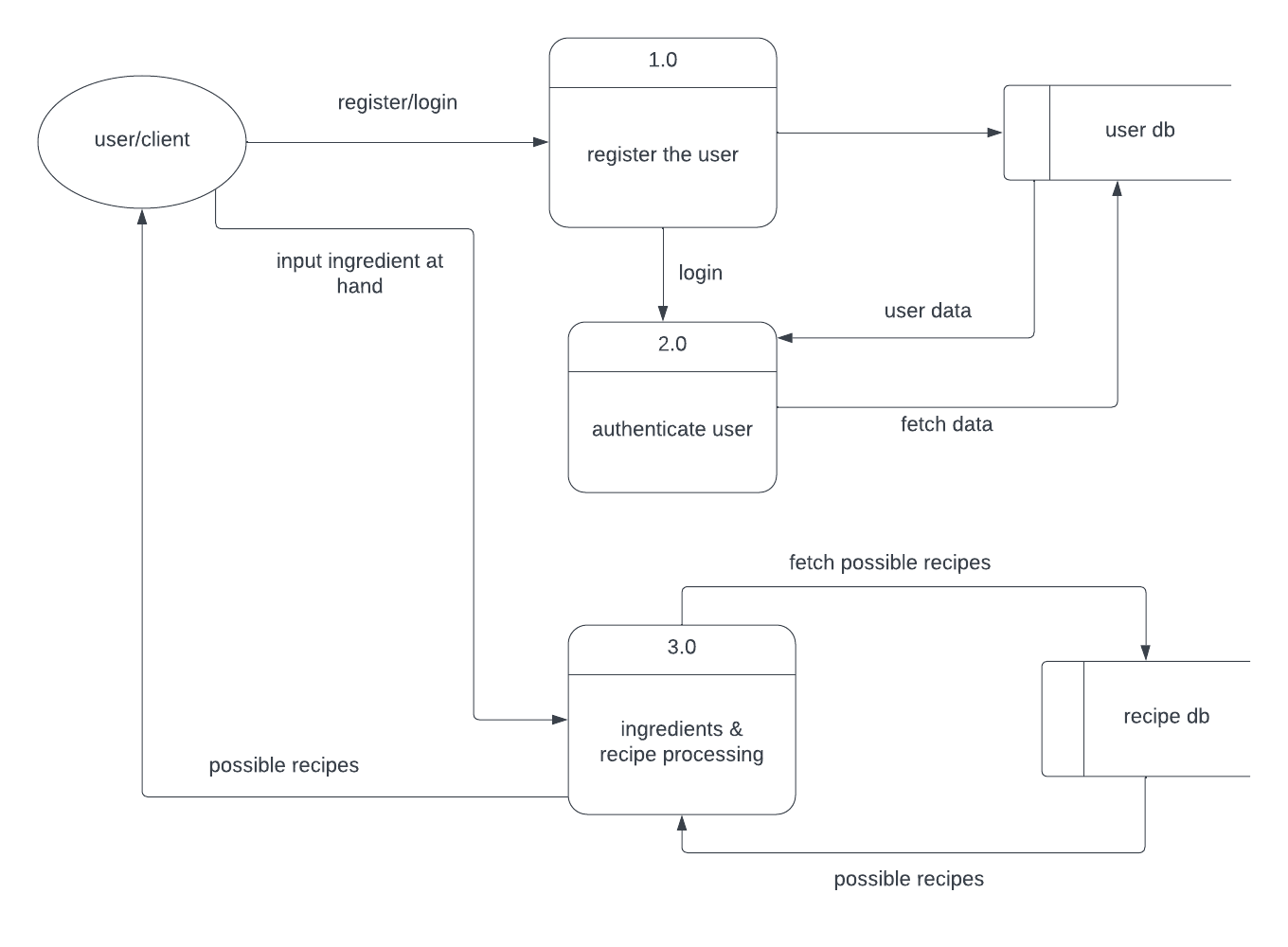


Figure4 dfd(level1) of the system

This diagram shows the whole system is represented as a single process. A level 1 DFD notates each of the main sub-processes that together form the complete system.

# 5. Feasibility study

5.1 Technical Feasibility

The success of the system implemented from technical perspective depends on several factors that are going to be assessed below:

1. Technical requirements: the technical requirements of our system including software specification hardware compatibility needed to be carefully considered and met and once the system meets these identified requirements it will be a strong foundation for success
2. Availability of resources: sufficient resources including skilled development team, project managers, and other personnel who should be allocated to handle the development and system maintenance will generate a success
3. Adequate infrastructures: the availability of infrastructure like servers, networking equipments, storages are essential for smooth functioning of the system hence the success of our system
4. Expertise and knowledge: availability of the experts and knowledge in frameworks, programming language, and tools is vital for the successful implementation of our system

By meeting these aspects this will guarantee our system’s technical prospective success

## 5.2 Financial Feasibility

The financial viability of the system depend on various factors, and it is important to conduct the financial planning as a way of determining the financial viability for the system. Here are some of the factors that determine the financial viability of a system:

Revenue streaming: before you have to identify the potential revenue sources for the system. this may include features like advertising

Estimated cost: you have to check if all the costs associated with developing, maintaining and all other system activities. Carefully you have to analyze and quantify these costs to get a clear understanding of the investment required

Return on investment: this is a measure of the profitability of an investment relative to its costs. You have to compare your costs and project revenue over a specific period of time which helps to know the financial standard of your system

Profitability analysis: perform a profitability by subtracting the estimated costs from the project generated revenue by considering the factors like gross profit, net profit and an analyze the profitability of different revenue outcomes

## 5.3 Market Feasibility

This feasibility refers to the potential of the system to succeed and generate demand within the targeted market and this involves evaluating some factors which results into a definitive assessment. The factors to be considered in evaluating the market feasibility are:

Market demand: according to the analyze current demand for the recipe recommendation systems it clearly show that here is availability of these system at the market

Competitive landscape: according to the existing competitors it clearly show that the system will be successful as those for the competitors lack some feature like data authentication which will lead to the success of this system

Size of the market: this system will have a wide market size as there is a lot of users demanding to easily have the recipes without requiring more skills

## 5.4 Economic Feasibility

Assessing the economic feasibility of this system we involved considering some factors such as its impact on the local economy, job creation. While a comprehensive analysis requires detailed local economic data and specific context these considerations for evaluating the economic feasibility are

Local economy impact: this system can increase the economic activities in the region and it is also an economic activity too and it can contribute to the local ecosystem,

Job creation: this system is economic viable as it provides jobs to those programmers and many other for its implementation and maintenance

Income generation: this system is able to generate income both to the users and the developer as the developers gain salaries and the users are able to do their task resulting to the income they can gain after the task completion

Multiple effect this system will generate multiple activities and it will act as the multiplier to the economy as it is able to simulate spending from those needing to use it, it will also simulate investment to those willing to develop it wider, and simulate entrepreneurship to those delivering it to the users

## 5.5 Legal and Regulatory Feasibility

.this system will meet legal and regulatory feasibility as is compliance with the applicable laws, regulations and standards. Here are some suggestions to consider when conducting the legal and regulatory feasibility assessment:

Identify relevant laws and regulations: this system had to be fulfilling with the legal frameworks such as data protection, privacy and any specific regulations related to the food and recipe or nutrition information

Property rights: the system also fits with the intellectual property rights like copyright, patent, licenses and trademark that are supposed to be unique for each system

Consumer protection: the consumers are protected by the law as there is transparency and fairness practice within the system

## 5.6 Operational Feasibility

This will assess if our system can be integrated into the existing system and assess whether the system can be deployed and maintained such that it can be used in a practical manner

Infrastructure requirement: this system needs servers and networking equipments, storages are essential to make sure that the system is practical

Operational processes: the systems’ operations requires other processes via browsers so it can interact with the existing system

# 6.Description of the selected Process Model

Based on the characteristics and the requirements of the G.Kings system, the suitable process model is Agile. It emphasizes the flexibility, collaboration, and incremental delivery, making them suited for the systems that require frequent iterations, adaptability to changing the requirements. Here are the descriptions of the agile process model and how it can be applied to develop our system:

1. Requirement gathering: the project team collaboratively gathers and analyzes user requirements of the system. These requirements are prioritized based on the importance and their values.
2. Sprint planning: the development tasks are organized into short time frames called sprint, which last 1-4weeks. in a sprint planning the team selects a set of prioritized requirements and break them into manageable tasks to be completed within that sprint
3. Development of the sprint: this involves the designing of the user interface, developing the backend functionality and conducting unit test
4. Daily stand meeting: meanings are help to keep team up dated the progress and each team member shares what they have accomplished and the plans they have on the tasks
5. Testing and sprint review : as they progress development this goes with a frequent testing and the review on the sprint where the team showcases the completed features and gather the feedback from the users as well as the stake holders
6. Iterative development: the development continues with subsequent sprint therefore each building upon the work completed in the previous iteration
7. Release: once the number of the sprint and outcomes becomes satisfying there is a release candidate prepared for deployment, including system tasting and users acceptance testing conducted to ensure the quality of the release
8. Maintenance: after deployment the system enters the maintenance phase, where the bugs are fixed security updates and other improvements based on the users’ feedback

## 6.1Cause of your selection

The system is based on several factors that make it suitable choice The main causes that makes agile development model suitable for the system are:

User centricity: the system is focused on providing recipe recommendations to user. Agile methods emphasize user involvement and feedback throughout the development

Flexibility and adaptability: agile allows for flexibility and adaptability to changing requirements, system may require frequent adjustment and refinements and based on user feedback

Collaboration: agile emphasize collaboration and regular communication among team members, stakeholders, and end users

Risk management: agile’s iterative and incremental approach helps mitigate risks by breaking the development process into smaller iteration; issues can be overall project risks.

# 7.Conclusion

In conclusion, the G.Kings-System is a recipe recommendation system that allows users to input their available ingredients and receive personalized recipe suggestions. Throughout the discussion, we have covered various aspects related to the system, including functional and non-functional requirements, organizational considerations, implementation requirements, external environment factors, privacy and safety requirements, software specifications, user requirements, portability, reliability, usability, and space requirements. The economic feasibility of the system involves evaluating factors such as cost-benefit analysis, investment requirements, potential revenues, return on investment, and profitability. By considering these factors, it is possible to assess the financial viability of the system and determine its potential for generating sustainable revenue and returns. Legal and regulatory feasibility requires compliance with applicable laws, regulations, and standards. It involves identifying relevant legal frameworks, engaging legal expertise, and ensuring adherence to privacy, data protection, intellectual property, consumer protection, and other relevant regulations Operational feasibility involves assessing the system's integration capabilities, infrastructure requirements, resource availability, training needs, operational processes, scalability, maintenance, and support. Ensuring seamless integration, effective utilization of resources, and user adoption are crucial for the successful operation of the system. Overall, the G.Kings-System shows promise as a user-friendly recipe recommendation solution that addresses the needs of users seeking recipe suggestions based on available ingredients. Conducting further analysis, market research, and considering specific business objectives and constraints will provide a more comprehensive understanding of the system's viability and potential for success.

# 8. References

[ISO/IEC/IEE 29148-201](C:\\Users\\user\\AppData\\Local\\Microsoft\\Windows\\INetCache\\IE\\4TGTZQ3W\\crispin)

[www.rdb.gov.rw](http://www.rdb.gov.rw)

[www.alecsoft.com/blog/business analysis/](http://www.alecsoft.com/blog/business%20analysis/)

**Section II: database**

-- phpMyAdmin SQL Dump

-- version 5.2.1

-- https://www.phpmyadmin.net/

--

-- Host: 127.0.0.1

-- Generation Time: Sep 16, 2023 at 12:56 AM

-- Server version: 10.4.28-MariaDB

-- PHP Version: 8.2.4

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `crispin\_shyaka\_22204852`

--

DELIMITER $$

--

-- Procedures

--

CREATE DEFINER=`root`@`localhost` PROCEDURE `deleteuser` (`p\_user\_id` INT) begin

delete from users

where

user\_id = p\_user\_id;

end$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `displayall` () begin

select \* from users;

select \* from ingredients;

select \* from ratings;

end$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `insertuser` (IN `p\_first\_name` VARCHAR(20), IN `p\_emal` VARCHAR(100), IN `p\_paword` VARCHAR(8)) begin

insert into users (first\_name,emal,paword)

values (p\_first\_name,p\_emal,p\_paword);

end$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `updatedat` (IN `p\_user\_id` INT, IN `p\_new\_name` VARCHAR(25), IN `p\_new\_cooking\_level` VARCHAR(50)) begin

update users

set first\_name = p\_new\_name,

cooking\_level = p\_new\_cooking\_level

where user\_id = p\_user\_id;

end$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `updatedata` (IN `p\_user\_id` INT, IN `p\_new\_name` VARCHAR(25), IN `p\_new\_cooking\_level` VARCHAR(50)) begin

update users

set first\_name = new\_name,

cooking\_level = new\_cookng\_level

where user\_id = p\_user\_id;

end$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `updatedatas` (IN `p\_user\_id` INT, IN `p\_new\_name` VARCHAR(25), IN `p\_new\_cooking\_level` VARCHAR(50)) begin

update users

set first\_name = p\_new\_name,

cooking\_level = p\_new\_cookng\_level

where user\_id = p\_user\_id;

end$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `userwithmostrecipe` () begin

select u.first\_name,u.emal

from users u

join(

select user\_id, count(\*)

as savedrecipecount

from saved\_recipes

group by user\_id

order by savedrecipecount desc

limit 1

)

as subquery

on u.user\_id = subquery.user\_id;

end$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `userwithmostrecipes` () begin

select u.first\_name,u.emal

from user u

join(

select user\_id, count(\*)

as savedrecipecount

from saved\_recipes

group by user\_id

order by savedrecipecount desc

limit 1

)

as subquery

on u.user\_id = subquery.user\_id;

end$$

DELIMITER ;

-- --------------------------------------------------------

--

-- Stand-in structure for view `ingredentsview`

-- (See below for the actual view)

--

CREATE TABLE `ingredentsview` (

`ingredent\_id` int(11)

,`name` varchar(15)

,`category` varchar(10)

,`nutrition\_informaton` varchar(50)

,`avalability` varchar(18)

);

-- --------------------------------------------------------

--

-- Table structure for table `ingredients`

--

CREATE TABLE `ingredients` (

`ingredent\_id` int(11) NOT NULL,

`name` varchar(15) NOT NULL,

`category` varchar(10) DEFAULT NULL,

`nutrition\_informaton` varchar(50) DEFAULT NULL,

`avalability` varchar(18) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `ingredients`

--

INSERT INTO `ingredients` (`ingredent\_id`, `name`, `category`, `nutrition\_informaton`, `avalability`) VALUES

(1, 'Onion', 'carbo', NULL, 'yes'),

(2, 'pillaw\_masala', 'spices', 'vitaminA', 'yes'),

(3, 'tea\_masala', 'spicies', 'vitamins', 'yes');

--

-- Triggers `ingredients`

--

DELIMITER $$

CREATE TRIGGER `updateingredients` AFTER UPDATE ON `ingredients` FOR EACH ROW begin

insert into ingredientlog(ingredent\_id,action,timestamp)

values (new.ingredent\_id,'user\_updated',now());

end

$$

DELIMITER ;

-- --------------------------------------------------------

--

-- Stand-in structure for view `insertusers`

-- (See below for the actual view)

--

CREATE TABLE `insertusers` (

`user\_id` int(11)

,`first\_name` varchar(20)

,`last\_name` varchar(15)

,`emal` varchar(30)

,`paword` varchar(8)

,`preferrences` varchar(20)

,`cooking\_level` varchar(10)

);

-- --------------------------------------------------------

--

-- Table structure for table `ratings`

--

CREATE TABLE `ratings` (

`rating\_id` int(11) NOT NULL,

`recipe\_id` int(11) DEFAULT NULL,

`user\_id` int(11) DEFAULT NULL,

`ratng\_value` int(11) DEFAULT NULL,

`revew` varchar(10) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `ratings`

--

INSERT INTO `ratings` (`rating\_id`, `recipe\_id`, `user\_id`, `ratng\_value`, `revew`) VALUES

(1, 1, 2, 5, 'awesome');

-- --------------------------------------------------------

--

-- Table structure for table `recipelog`

--

CREATE TABLE `recipelog` (

`log\_id` int(11) NOT NULL,

`recipe\_id` int(11) DEFAULT NULL,

`action` varchar(20) DEFAULT NULL,

`timestamp` varchar(50) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

-- --------------------------------------------------------

--

-- Table structure for table `recipes`

--

CREATE TABLE `recipes` (

`recipe\_id` int(11) NOT NULL,

`title` varchar(10) DEFAULT NULL,

`ingredient\_lst` varchar(100) DEFAULT NULL,

`prepqraton\_time` int(11) DEFAULT NULL,

`cooking\_time` int(11) DEFAULT NULL,

`nutritional\_info` varchar(100) DEFAULT NULL,

`user\_rating` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `recipes`

--

INSERT INTO `recipes` (`recipe\_id`, `title`, `ingredient\_lst`, `prepqraton\_time`, `cooking\_time`, `nutritional\_info`, `user\_rating`) VALUES

(1, 'breakfast', 'meat\_onions\_eggs', NULL, 1, NULL, 20);

--

-- Triggers `recipes`

--

DELIMITER $$

CREATE TRIGGER `insertafterrecipe` AFTER INSERT ON `recipes` FOR EACH ROW Begin

Insert into recipelog(recipe\_id,action,timestamp)

Values (new.recipe\_id,'recipe\_added',now());

End

$$

DELIMITER ;

-- --------------------------------------------------------

--

-- Stand-in structure for view `recipestodelete`

-- (See below for the actual view)

--

CREATE TABLE `recipestodelete` (

`save\_id` int(11)

,`user\_id` int(11)

,`save\_data` varchar(100)

);

-- --------------------------------------------------------

--

-- Table structure for table `saved\_recipes`

--

CREATE TABLE `saved\_recipes` (

`save\_id` int(11) NOT NULL,

`user\_id` int(11) DEFAULT NULL,

`recipe\_id` int(11) DEFAULT NULL,

`save\_data` varchar(100) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `saved\_recipes`

--

INSERT INTO `saved\_recipes` (`save\_id`, `user\_id`, `recipe\_id`, `save\_data`) VALUES

(1, 1, 1, 'breakfast\_great\_recipe');

-- --------------------------------------------------------

--

-- Stand-in structure for view `userinsert`

-- (See below for the actual view)

--

CREATE TABLE `userinsert` (

`user\_id` binary(0)

,`frist\_name` char(0)

,`last\_name` char(0)

,`emal` char(0)

,`paword` char(0)

,`preferrence` char(0)

,`cooking\_level` char(0)

);

-- --------------------------------------------------------

--

-- Table structure for table `userlog`

--

CREATE TABLE `userlog` (

`log\_id` int(11) NOT NULL,

`user\_id` int(11) DEFAULT NULL,

`action` varchar(50) DEFAULT NULL,

`timestamp` varchar(10) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `userlog`

--

INSERT INTO `userlog` (`log\_id`, `user\_id`, `action`, `timestamp`) VALUES

(1, 6, 'user added', '2023-09-09'),

(2, 4, 'user\_updated', '2023-09-09');

-- --------------------------------------------------------

--

-- Table structure for table `users`

--

CREATE TABLE `users` (

`user\_id` int(11) NOT NULL,

`first\_name` varchar(20) NOT NULL,

`last\_name` varchar(15) NOT NULL,

`emal` varchar(30) DEFAULT NULL,

`paword` varchar(8) DEFAULT NULL,

`preferrences` varchar(20) DEFAULT NULL,

`cooking\_level` varchar(10) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `users`

--

INSERT INTO `users` (`user\_id`, `first\_name`, `last\_name`, `emal`, `paword`, `preferrences`, `cooking\_level`) VALUES

(1, 'karangwa', 'umutoni', 'ritaumutogmail.com', 'rita@123', 'snacks', 'bad'),

(2, 'junior', 'johnson', 'junjo@gmail.com', 'jun123', 'sou', 'worst'),

(3, 'jean', 'shima', '12@gmail.com', 'jean12', 'spicy', 'bad'),

(4, 'cristelly', 'manzi', 'jeff@gmail.com', 'manzi12', 'drinks', 'yammy'),

(5, 'john\_doe', '', 'john256@gmail.com', 'secure', NULL, NULL),

(6, 'jack', '', 'jac@gmal.com', 'cured', NULL, NULL);

--

-- Triggers `users`

--

DELIMITER $$

CREATE TRIGGER `deleteuser` AFTER DELETE ON `users` FOR EACH ROW begin

insert into userlog(user\_id,action,timestamp)

values (old.user\_id,'user\_deleted',now());

end

$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER `insertafter` AFTER INSERT ON `users` FOR EACH ROW begin

insert into userlog(user\_id,action,timestamp)

values(new.user\_id,'user added',now());

end

$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER `updateusers` AFTER UPDATE ON `users` FOR EACH ROW begin

insert into userlog(user\_id,action,timestamp)

values (new.user\_id,'user\_updated',now());

end

$$

DELIMITER ;

-- --------------------------------------------------------

--

-- Stand-in structure for view `usersavedrecipes`

-- (See below for the actual view)

--

CREATE TABLE `usersavedrecipes` (

`user\_id` int(11)

,`first\_name` varchar(20)

,`save\_id` int(11)

,`savedrecipetitle` varchar(10)

);

-- --------------------------------------------------------

--

-- Stand-in structure for view `usersingredients`

-- (See below for the actual view)

--

CREATE TABLE `usersingredients` (

`first\_name` varchar(20)

,`last\_name` varchar(15)

,`ingredent\_id` int(11)

,`name` varchar(15)

);

-- --------------------------------------------------------

--

-- Table structure for table `user\_login`

--

CREATE TABLE `user\_login` (

`login\_id` int(11) NOT NULL,

`user\_id` int(11) DEFAULT NULL,

`password` varchar(8) DEFAULT NULL,

`action\_type` varchar(20) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `user\_login`

--

INSERT INTO `user\_login` (`login\_id`, `user\_id`, `password`, `action\_type`) VALUES

(1, 1, 'rita@123', 'chief');

-- --------------------------------------------------------

--

-- Structure for view `ingredentsview`

--

DROP TABLE IF EXISTS `ingredentsview`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `ingredentsview` AS SELECT `ingredients`.`ingredent\_id` AS `ingredent\_id`, `ingredients`.`name` AS `name`, `ingredients`.`category` AS `category`, `ingredients`.`nutrition\_informaton` AS `nutrition\_informaton`, `ingredients`.`avalability` AS `avalability` FROM `ingredients` ;

-- --------------------------------------------------------

--

-- Structure for view `insertusers`

--

DROP TABLE IF EXISTS `insertusers`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `insertusers` AS SELECT `users`.`user\_id` AS `user\_id`, `users`.`first\_name` AS `first\_name`, `users`.`last\_name` AS `last\_name`, `users`.`emal` AS `emal`, `users`.`paword` AS `paword`, `users`.`preferrences` AS `preferrences`, `users`.`cooking\_level` AS `cooking\_level` FROM `users` ;

-- --------------------------------------------------------

--

-- Structure for view `recipestodelete`

--

DROP TABLE IF EXISTS `recipestodelete`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `recipestodelete` AS SELECT `saved\_recipes`.`save\_id` AS `save\_id`, `saved\_recipes`.`user\_id` AS `user\_id`, `saved\_recipes`.`save\_data` AS `save\_data` FROM `saved\_recipes` WHERE `saved\_recipes`.`user\_id` = 3 ;

-- --------------------------------------------------------

--

-- Structure for view `userinsert`

--

DROP TABLE IF EXISTS `userinsert`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `userinsert` AS SELECT NULL AS `user\_id`, '' AS `frist\_name`, '' AS `last\_name`, '' AS `emal`, '' AS `paword`, '' AS `preferrence`, '' AS `cooking\_level` ;

-- --------------------------------------------------------

--

-- Structure for view `usersavedrecipes`

--

DROP TABLE IF EXISTS `usersavedrecipes`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `usersavedrecipes` AS SELECT `u`.`user\_id` AS `user\_id`, `u`.`first\_name` AS `first\_name`, `sr`.`save\_id` AS `save\_id`, `r`.`title` AS `savedrecipetitle` FROM ((`users` `u` join `saved\_recipes` `sr` on(`u`.`user\_id` = `sr`.`save\_id`)) join `recipes` `r` on(`sr`.`save\_id` = `r`.`recipe\_id`)) ;

-- --------------------------------------------------------

--

-- Structure for view `usersingredients`

--

DROP TABLE IF EXISTS `usersingredients`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `usersingredients` AS SELECT `users`.`first\_name` AS `first\_name`, `users`.`last\_name` AS `last\_name`, `ingredients`.`ingredent\_id` AS `ingredent\_id`, `ingredients`.`name` AS `name` FROM (`ingredients` join `users` on(`users`.`user\_id` = `ingredients`.`ingredent\_id`)) ;

--

-- Indexes for dumped tables

--

--

-- Indexes for table `ingredients`

--

ALTER TABLE `ingredients`

ADD PRIMARY KEY (`ingredent\_id`);

--

-- Indexes for table `ratings`

--

ALTER TABLE `ratings`

ADD PRIMARY KEY (`rating\_id`),

ADD KEY `recipe\_id` (`recipe\_id`),

ADD KEY `user\_id` (`user\_id`);

--

-- Indexes for table `recipelog`

--

ALTER TABLE `recipelog`

ADD PRIMARY KEY (`log\_id`),

ADD KEY `recipe\_id` (`recipe\_id`);

--

-- Indexes for table `recipes`

--

ALTER TABLE `recipes`

ADD PRIMARY KEY (`recipe\_id`);

--

-- Indexes for table `saved\_recipes`

--

ALTER TABLE `saved\_recipes`

ADD PRIMARY KEY (`save\_id`),

ADD KEY `user\_id` (`user\_id`),

ADD KEY `recipe\_id` (`recipe\_id`);

--

-- Indexes for table `userlog`

--

ALTER TABLE `userlog`

ADD PRIMARY KEY (`log\_id`),

ADD KEY `user\_id` (`user\_id`);

--

-- Indexes for table `users`

--

ALTER TABLE `users`

ADD PRIMARY KEY (`user\_id`);

--

-- Indexes for table `user\_login`

--

ALTER TABLE `user\_login`

ADD PRIMARY KEY (`login\_id`),

ADD KEY `user\_id` (`user\_id`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `ingredients`

--

ALTER TABLE `ingredients`

MODIFY `ingredent\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;

--

-- AUTO\_INCREMENT for table `recipelog`

--

ALTER TABLE `recipelog`

MODIFY `log\_id` int(11) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT for table `recipes`

--

ALTER TABLE `recipes`

MODIFY `recipe\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT for table `saved\_recipes`

--

ALTER TABLE `saved\_recipes`

MODIFY `save\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT for table `userlog`

--

ALTER TABLE `userlog`

MODIFY `log\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT for table `users`

--

ALTER TABLE `users`

MODIFY `user\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=7;

--

-- AUTO\_INCREMENT for table `user\_login`

--

ALTER TABLE `user\_login`

MODIFY `login\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- Constraints for dumped tables

--

--

-- Constraints for table `ratings`

--

ALTER TABLE `ratings`

ADD CONSTRAINT `ratings\_ibfk\_1` FOREIGN KEY (`recipe\_id`) REFERENCES `recipes` (`recipe\_id`),

ADD CONSTRAINT `ratings\_ibfk\_2` FOREIGN KEY (`user\_id`) REFERENCES `users` (`user\_id`);

--

-- Constraints for table `recipelog`

--

ALTER TABLE `recipelog`

ADD CONSTRAINT `recipelog\_ibfk\_1` FOREIGN KEY (`recipe\_id`) REFERENCES `recipes` (`recipe\_id`);

--

-- Constraints for table `saved\_recipes`

--

ALTER TABLE `saved\_recipes`

ADD CONSTRAINT `saved\_recipes\_ibfk\_1` FOREIGN KEY (`user\_id`) REFERENCES `users` (`user\_id`),

ADD CONSTRAINT `saved\_recipes\_ibfk\_2` FOREIGN KEY (`recipe\_id`) REFERENCES `recipes` (`recipe\_id`);

--

-- Constraints for table `userlog`

--

ALTER TABLE `userlog`

ADD CONSTRAINT `userlog\_ibfk\_1` FOREIGN KEY (`user\_id`) REFERENCES `users` (`user\_id`);

--

-- Constraints for table `user\_login`

--

ALTER TABLE `user\_login`

ADD CONSTRAINT `user\_login\_ibfk\_1` FOREIGN KEY (`user\_id`) REFERENCES `users` (`user\_id`);

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

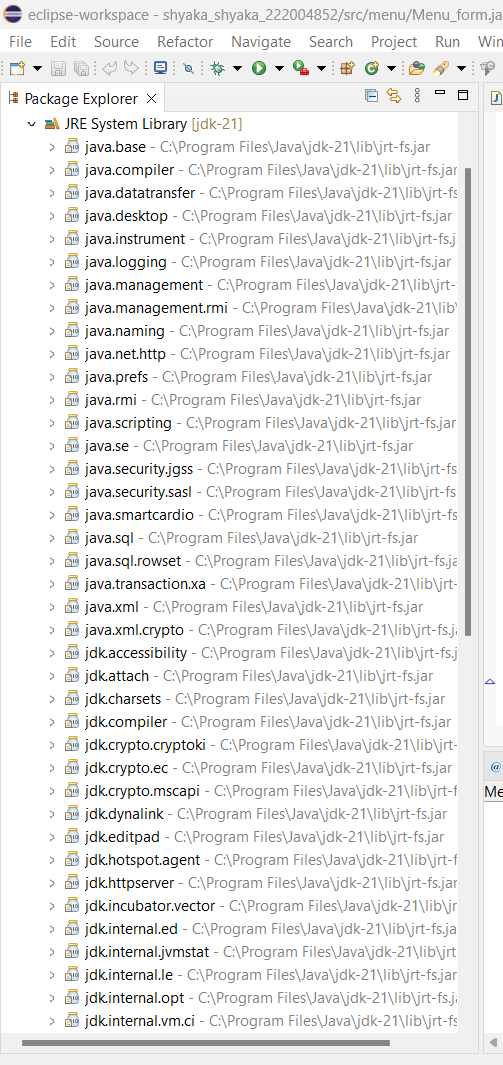
/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

**Section III: java programming**

**In** this section we are going to see how our gkings recipe recommendation system will be working according to the forms we have

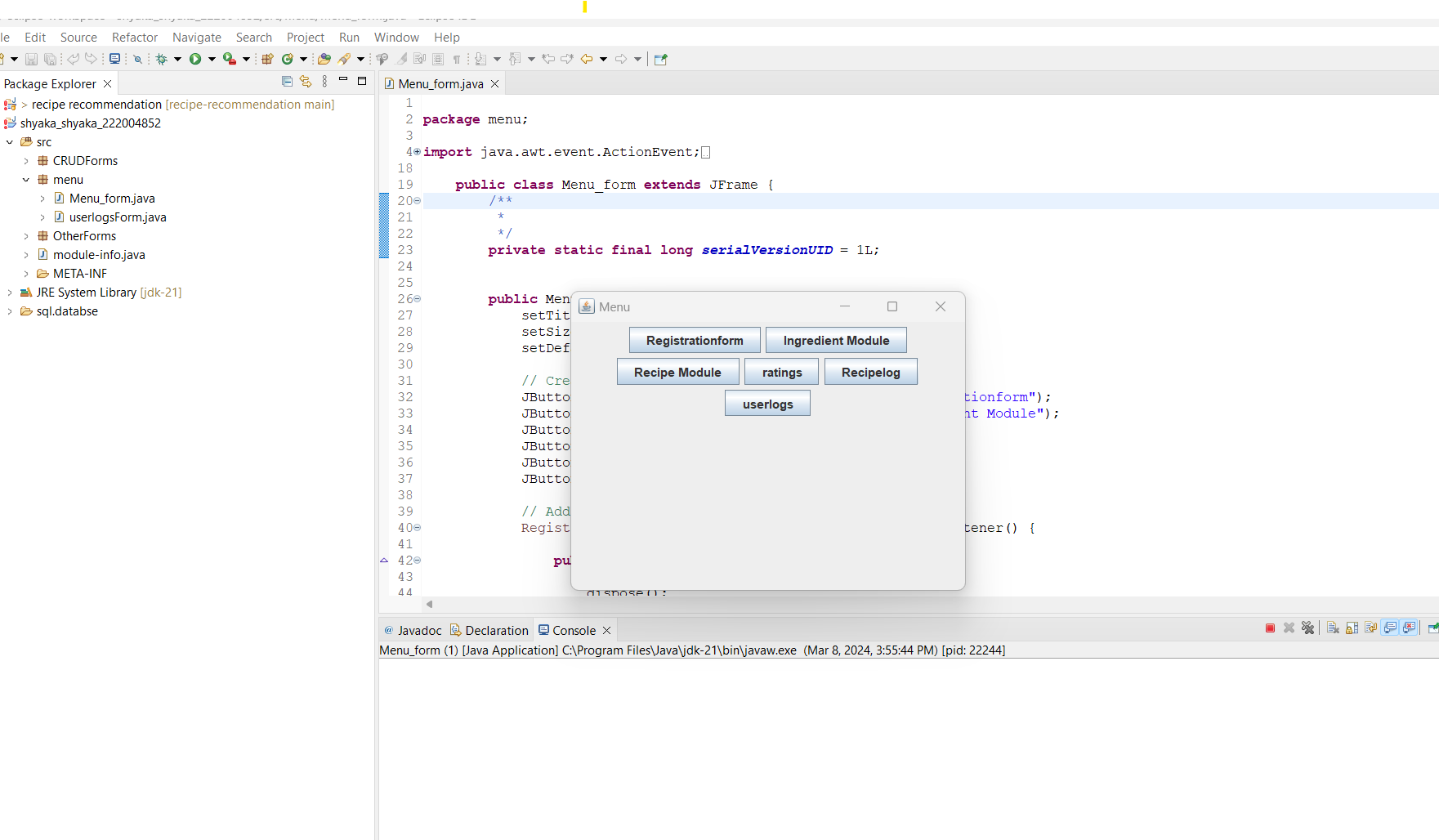
Lets begin with the jars these are also known as java archives : It's a file format based on the popular ZIP file format and is used for aggregating many files into one. Although JAR can be used as a general archiving tool, the primary motivation for its development was so that Java applets and their requisite components.



Mysql connectors: these are standard based drivers that are responsible for connecting java eclipse with mysql database

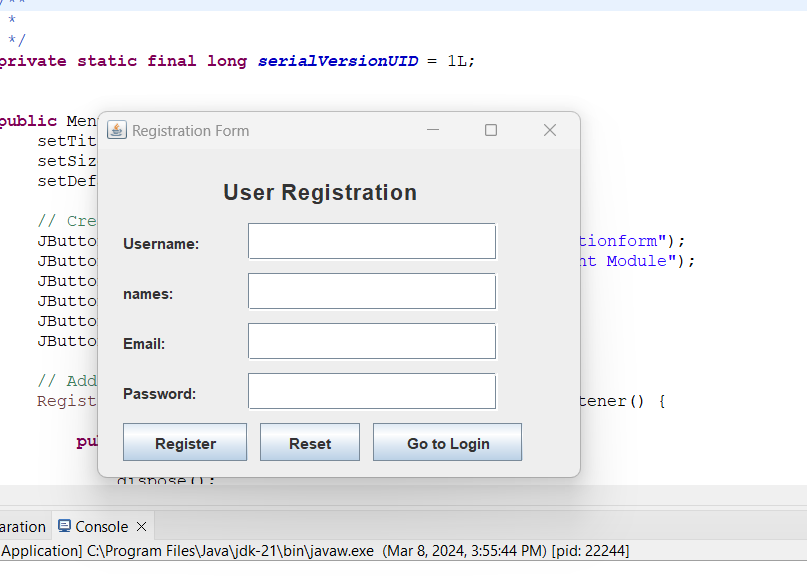
**FORMS AND DESCRIPTION**

On our system we will be having a menu form displaying the available tables in our system

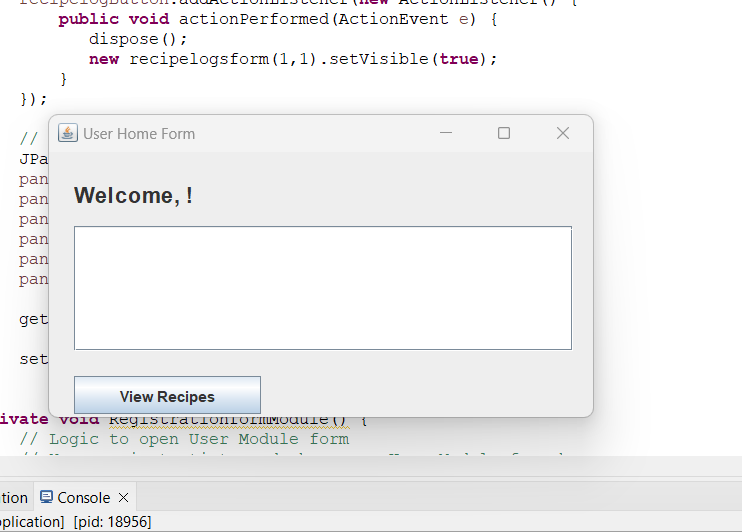


When you click on registration button it will open a new form : the user registration

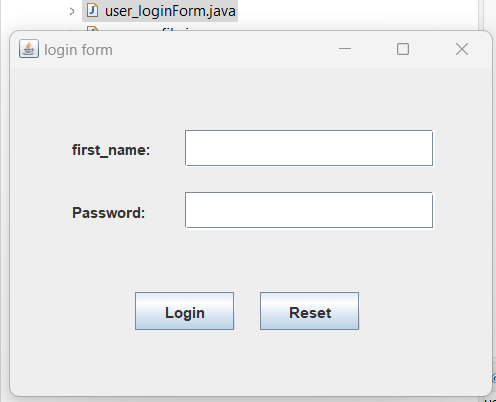
This form will allow the user to register into our system by inserting the username names email and password



Once he click on register button he will be registered and once he click on reset button the data inserted will be erased and when you click on go to login this will take the user to the userlog form that allows the users to view the recipe as they click on view recipes button

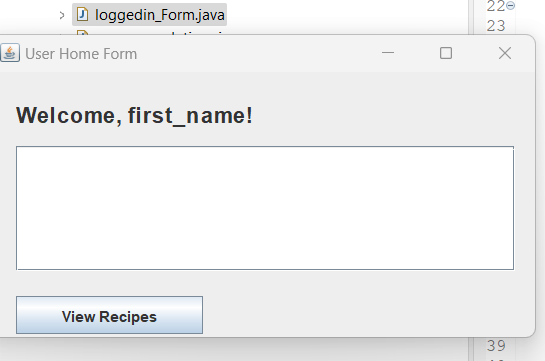


**User login form**

****

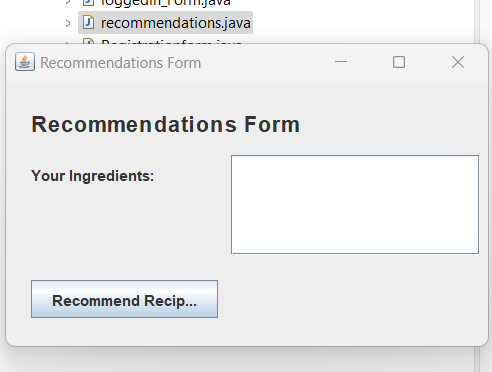
This form allow the user to login through the system and once they insert the requested data they and click on the login button they will be redirected to the:

**User loggedin form**

****

this form allows the user to insert there ingredients they have and they will receive the corresponding recipes after clicking the view recipes button

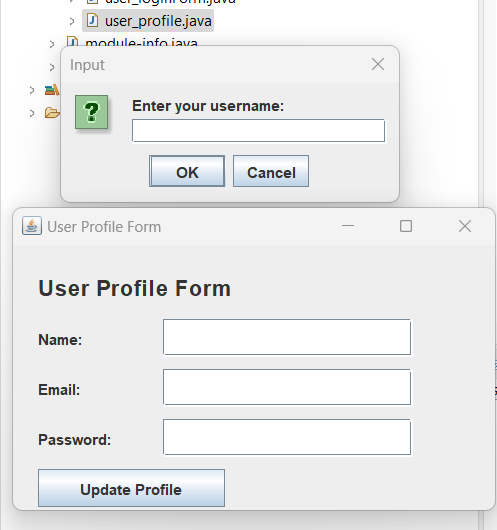
**RECOMMENDATIONS FORM**

****

This form will allow the user to be recommended a recipe that base on the ingredients that are inserted in the system and after clicking the recommend recipe button

**USER PROFILE**

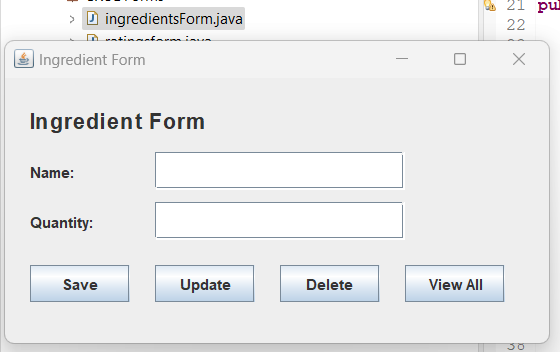
this is the form that allow se user to update his information and it opens after it verifies that the user is valid by requesting the username before opening the official form



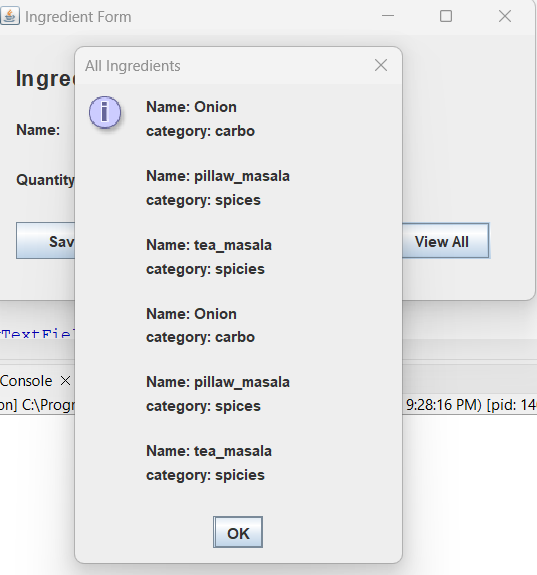
After the user inserts the first name in the username textfield provided before which will open this user profile form

**INGREDIENTS FORM**

This is the form that provides the ingredients available in our database

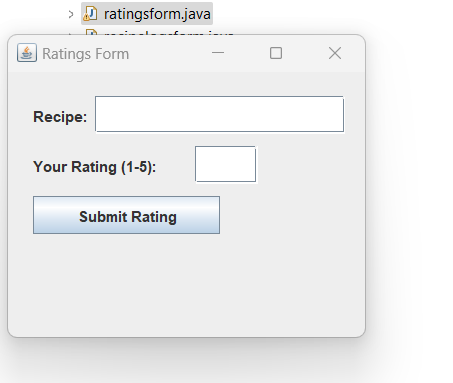


This form will allow the insert of new ingredients when the user click on save button or to update the quantity of the ingredients by clicking the update button



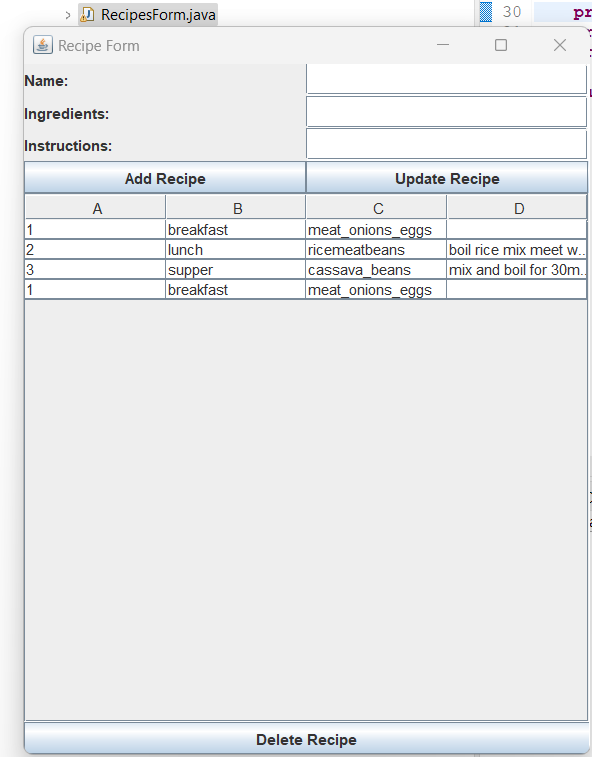
This is the result of clicking the view all button which pops up all the available ingredients in our system

**RATINGS FORM**

****

This form allows the user to provide a certain rating to the recipe that was recommended as they can rate it to 5 as they insert in the recipe id

**RECIPES FORM**

****

This form provides the user to insert the new recipes as they click the add button or they can update the recipe data as they click on the update button and the user can delete the recipe as they click on the recipe of his wish and then click on delete button

Reference

1. www.eclipse.org>windowbuilider
2. Java Performance: The Definite Guide First Edition (Scott Oaks, 2017)