**Recipe Application for Tons of Food for U (T.O.F.U)**

Nilesh Shrestha, Rijan Prajapati, Yanish Singh, Ashmita Shahi

B.Sc. (Hons.) Computing, Softwarica College of IT and E-commerce, Coventry University

ST4008CEM: Computing Activity Led Learning Project 1

Giriraj Rawat

Feb 21, 2028

**Table of Contents**

[Introduction 4](#_Toc127654314)

[Aim 5](#_Toc127654315)

[Objectives 6](#_Toc127654316)

[Problem Statements 8](#_Toc127654317)

[Features 9](#_Toc127654318)

[Functional Requirements 10](#_Toc127654319)

[Non-Functional Requirements 17](#_Toc127654320)

[Scope of the Project 18](#_Toc127654321)

[Methodology 19](#_Toc127654322)

[Tools and Technologies used 22](#_Toc127654323)

[Conceptual Design 24](#_Toc127654324)

[System Architecture 25](#_Toc127654325)

[Project Plan 26](#_Toc127654326)

[Prototypes 29](#_Toc127654327)

[Developed System 32](#_Toc127654328)

[System Testing 45](#_Toc127654329)

[Version Control 49](#_Toc127654330)

[Conclusion 50](#_Toc127654331)

[References 51](#_Toc127654332)

**Table of Figures**

[Figure 1: objectives 6](#_Toc127391395)

[Figure 2: Problems that might occur and their solutions 9](#_Toc127391396)

[Figure 3: Agile methodology 18](#_Toc127391397)

[Figure 4: User stories 19](#_Toc127391398)

[Figure 5: Tools and Technologies 21](#_Toc127391399)

[Figure 6: entity relation diagram of recipe application 23](#_Toc127391400)

[Figure 7: two tier system architecture 24](#_Toc127391401)

[Figure 8: Login page 27](#_Toc127391402)

[Figure 9: login-page exception handling 28](#_Toc127391403)

[Figure 10: registration validation 30](#_Toc127391404)

[Figure 11: Recipe Application Page 31](#_Toc127391405)

[Figure 12: Code Snippet for Recipe Creation 32](#_Toc127391406)

[Figure 13: Ui Message when Recipe is successfully created 33](#_Toc127391407)

[Figure 14: Code Snippet for Fetching Recipe 34](#_Toc127391408)

[Figure 13: How the recipe is displayed after selecting a recipe and clicking on select recipe button 35](#_Toc127391409)

[Figure 15: Code Snippet for Updating recipe 35](#_Toc127391410)

[Figure 16: UI changes for Recipe after being updated 36](#_Toc127391411)

[Figure 17: Code Snippet for Recipe Deletion 37](#_Toc127391412)

[Figure 18: UI changes after a recipe has been deleted 37](#_Toc127391413)

[Figure 19: Code Snippet for searching a recipe 38](#_Toc127391414)

[Figure 20: UI result when 0 matches for recipe is found 39](#_Toc127391415)

[Figure 21: GitHub Repository 41](#_Toc127391416)

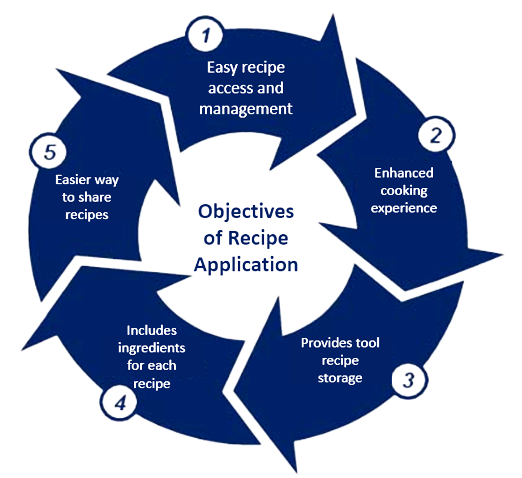
# Introduction

The recipe application is a user-friendly GUI application built with Python programming language. It caters to culinary enthusiasts by providing a simple and intuitive way to access and manage an extensive collection of recipes. The app features include searching, and creating recipes, with a user-friendly interface. Perfect for home-cooks and professional chefs, it elevates the culinary game and helps to improve cooking skills. (Developed as a school project, it demonstrates the students’ understanding of GUI and Python concepts).

# Aim

The aim of this recipe app is to provide users with a user-friendly and efficient tool to improve their cooking skills and explore new recipe ideas.

# Objectives

****

### Figure 1: objectives

The objective of creating the app is to showcase the learning of a programming language, understanding of GUI development and python programming to provide a

functional tool and designing and developing a recipe application or a recipe management system that allows user to manage a wide variety of recipes.

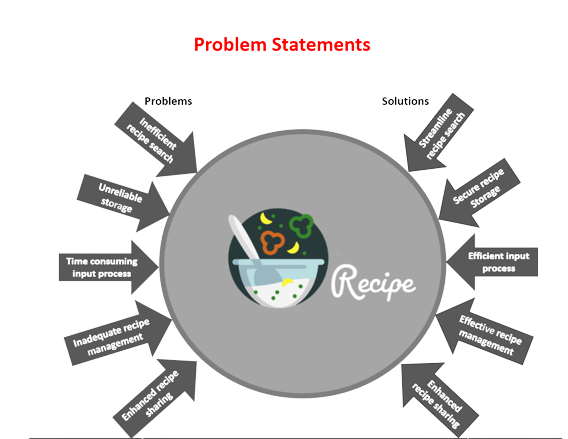
Some of the key objectives from the professional application developer point of view are:

* To provide users easy and intuitive way to access and manage wide variety of recipes and digitalize it
* To provide a tool for both home cooks and professional chefs to find, organize, and use recipes to create delicious meals.
* To enhance the cooking experience of both home cooks and professional chefs
* To develop a system that also provides the ingredients for a particular recipe
* Provides an easy way to share recipe among other users

And for the purpose of this from an academic point of view is to showcase the student’s understanding of Python programming, GUI development, Database Management System (DBMS) and algorithms by using various tools.

# Problem Statements

The methods for searching, finding and saving recipes online are outdated and unreliable, leading to difficulty in recipe management. The traditional approach of manually rewriting recipes on a paper or bookmarking can be time-consuming and prone to errors [(*Main Points of Food Recipe App Development*, n.d.).](#problem_statement)



### Figure 2: Problems that might occur and their solutions

The problems in recipe search and storage call for a modern solution. A recipe app offers effective answer, providing centralization, efficiency and functionality for recipe management by using modern tools and technologies to solve all of the above problems make it a valuable tool for home cook and culinary enthusiasts.

# Features

Some of the key features of this recipe application are:

* CRUD: Users can create, view, update and delete their recipes and its details; including title, ingredients and instructions
* User Interface: the app will have a graphical user interface (GUI) built with tkinter, making it simple and very easy to navigate and use
* Database integration: the application will store recipes and information in a database using SQLite, allowing for an efficient data management and retrieval
* Login and registration: the application will allow its user to create an account and login to personalize their recipes

These are the main features, but additional features may/may not be added in the future according to stakeholder and product owners’ needs.

# Functional Requirements

Functional requirements are the features and functions that the recipe application must have, they are as follows:

## Registration

The user registration functionality would allow a user to create a new account by entering their desired username and password along with any other required information such as their name and email address. This information will be stored in the database to be sued for authentication when the user tries to log in.

|  |  |
| --- | --- |
| Use Case Name | Register |
| Precondition | The form is displayed in the register page |
| Basic path | 1. A database is created to store all the users 2. The register page displays input fields like First name, Last name, Username, Password, Confirm-password, Email and a submit button 3. The values in input fields are necessary and must be filled. 4. The input values of username must not match the one that is already existing in the database and will, warn the user if it happens, whereas user can use the same email to create as many accounts as they want. 5. After submission, the application will send a confirmation email to verify the user. |

## Login

The user login functionality would require a secure mechanism for verifying the credentials of a user trying to log in to the application. This could be done by checking the entered username and password against a database of registered users. If the credentials match, the user would be granted access to the application. If not, the user would receive an error message indicating the login was unsuccessful.

|  |  |
| --- | --- |
| Use Case Name | Login |
| Precondition | The user must first register to the recipe application. |
| Basic path | 1. After registering an account in the application database, the user can redirect themselves to the login page 2. After logging in the user is granted access to the application |

## Recipe Creation

The creation functionality would allow a user to add a new recipe to the application by entering its name, ingredients and instructions. After the creation the application will display the name of the recipe in a list format, with options to view details of a specific recipe.

|  |  |
| --- | --- |
| Use Case Name | Recipe Creation |
| Precondition | The user must have an account and be logged in |
| Basic path | 1. After logging into the main screen, the user is able to see a form with fields, recipe title and details 2. The user cannot submit an empty form and is required to enter the necessary information to submit the form 3. The application stores the recipe in its database, confirms the recipe has been created and displays the recipe name in the list |

## Recipe Viewing

The recipe viewing functionality would allow a user to view all the recipes they have created, along with the details of each recipe such as name, ingredients, and instructions when a particular recipe is selected from the list.

|  |  |
| --- | --- |
| Use Case Name | Recipe Viewing |
| Precondition | The user must be logged in and must have created at least one recipe to access this functionality |
| Basic path | 1. User selects the “select recipe” from the list of recipes 2. The application displays the details of the selected recipes 3. The user can also choose to view another recipe |

## Recipe Updating

A user should be able update the recipe details such as changing its name or adding/removing ingredients

|  |  |
| --- | --- |
| Use Case Name | Recipe Updating |
| Precondition | The user must be logged in and must have created at least one recipe |
| Basic path | 1. User should select a recipe and then select “update recipe” button 2. The recipe details and the name will display in the same form field with the modify button enabled and create recipe button disabled 3. The user makes desired changes and submits it. 4. The application confirms the recipe has been updated. |

## Recipe Deletion

A user should be able to delete the recipe they no longer wish to keep

|  |  |
| --- | --- |
| Use Case Name | Recipe Deletion |
| Precondition | The user must be logged in and must have created at least one recipe |
| Basic path | 1. The user selects the recipe that they want to delete from the list 2. The user selects the “delete recipe” button to delete the recipe 3. The application confirms that the recipe has been deleted and displays the user message indicating it |

## Data Persistence

The data persistence functionality ensures that the information entered by the users, such as their recipes, would be stored in a database so that it can be accessed and used by the application even after the application has been closed [(Pfeil, 2010).](#Data_Persistance)

## Error Handling

The error handling functionality enables the application to handle errors gracefully, such as invalid login credentials, nonexistent recipe access, or lost database connection. An error message with relevant information or instruction is displayed to the user in these cases.

# Non-Functional Requirements

## 1. Usability

The application should be easy to understand, with a clear and intuitive user interface.

## 2. Performance

The application should be able to quickly and efficiently respond to the user inputs and requests.

## 3. Scalability

The application should be able to handle an increasing number of recipes and users as it grows.

## 4. Security

The application should protect sensitive information, such as user credentials from unauthorized access and manipulations.

## 5. Maintainability

The application should be easy to maintain and update with a well-organized codebase and documentation.

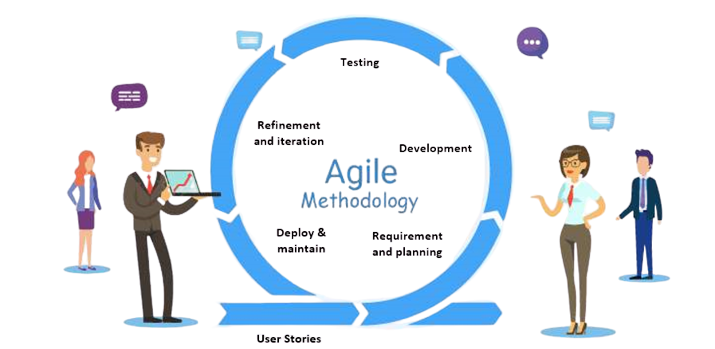
## 6. Reliability

The application should be robust and stable with minimal errors

# Scope of the Project

The scope of this project encompasses the design and implementation of a sophisticated graphical user interface (GUI) application for recipe management. The application will empower users to easily browse, search and create recipes, as well as organize their personal recipe collection with great ease. The application will be optimized for desktop platforms, with a user-centred design approach to ensure a seamless and intuitive user experience. The project will also include extensive testing and debugging to guarantee that the application meets all requirements and functions as intended. Furthermore, the project may also incorporate advanced features such as user accounts and recipe sharing. Upon completion, the project will be delivered with comprehensive user manual and technical documentation, ensuring ease of use and maintenance.

# Methodology



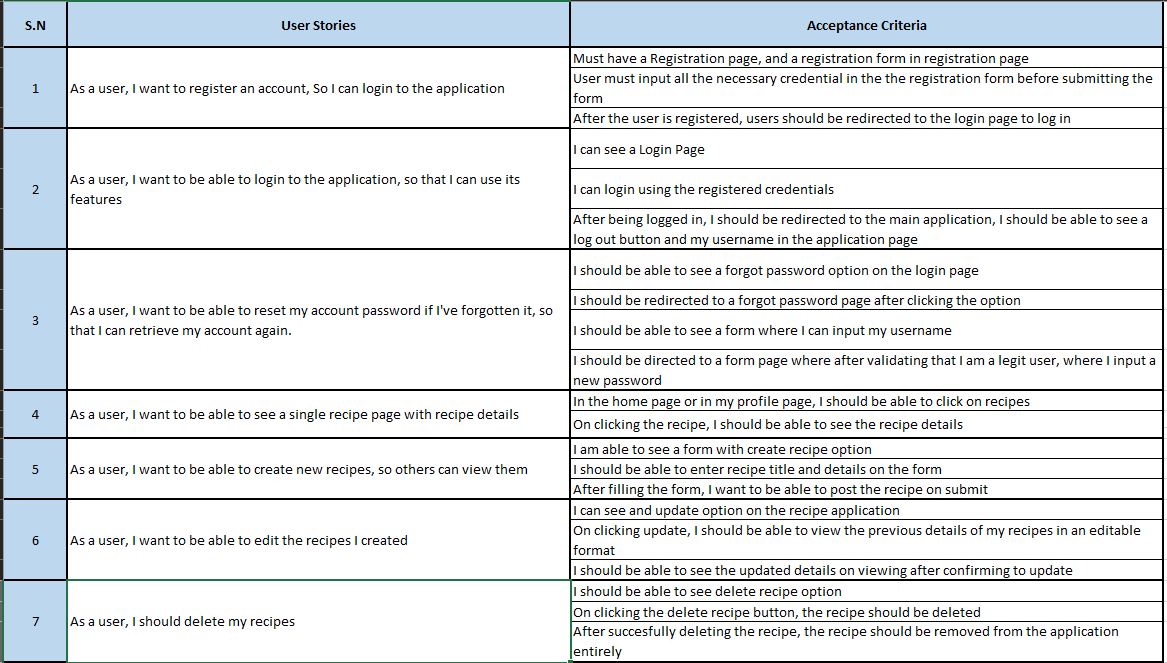
### Figure 3: Agile methodology

Agile is a flexible, iterative software development approach that prioritizes collaboration, customer satisfaction and adapting to change for success in an uncertain environment [(Agile Alliance, 2023).](#agile) Where iteration means to repeatedly develop a certain part systems functionality over a set period of time before moving onto the next one [(*Iteration*, 2022).](#iteration)

Iterative development is also a key aspect Agile methodology, it was used utilized during the development of this and how it works is given below:

## 1. User Stories

User stories describe user-desired functions, aiding developers in prioritizing features to meet the intended audience’s needs, ensuring that the application serves its purpose and offers a positive user experience. Additionally, user stories serve as the basis for system requirements.



### Figure 4: User stories

## 2. Requirements gathering and Planning

The development team engages with the client or stakeholder to gather requirements and devise a plan through user. As user stories are simple sentences that outline the desired outcome of. Requirements for initial iteration may include, login, registration or CRUD functionality, are prioritized and scoped for the first sprint and so on.

## Development

The development team executes tasks according to the plans and requirements gathered from clients during each sprint. They select tasks based on their area of expertise and focus on delivering high priority functionality, such as CRUD for recipe database, login or registration. Development is iterative, with prototypes of application delivered at end of each sprint.

## Testing

The development team evaluates the working prototype of each sprint to verify requirements and identify and potential bugs. They may also solicit feedback from the stakeholders during the testing process.

## Refinement and iteration

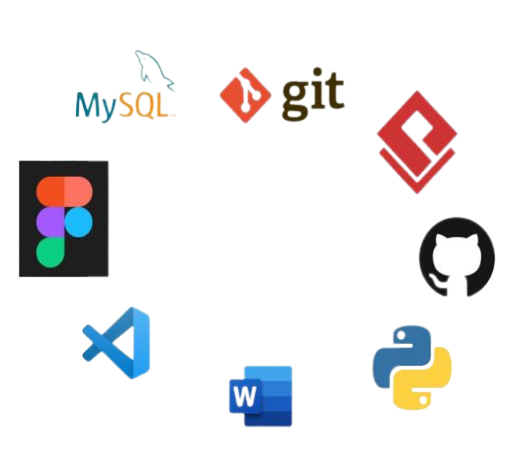
The team refines the prototypes based on testing and feedbacks and adds new features in subsequent sprints. For example, they may incorporate the forgot password or improved search functionality. This iterative development, testing and refinement is repeated until all the requirements are fulfilled and application is finalized.

## Deployment and Maintain

Once the application is complete, it is deployed for use by stakeholders and maintained by giving thorough updates in a timely manner.

Agile emphasizes iterative development and adaptation to changing requirements, resulting in high-quality software delivery.

# Tools and Technologies used

****

### Figure 5: Tools and Technologies

## 1. Python

A popular high-level programming language known for its readability and easy-to-learn syntax. Used in web development, machine learning and software development [(*What Is Python? Executive Summary*, n.d.)](#python).

## SQLite

It is a widely used, file-based, self-contained, transactional database management system (DBMS) that does not require a separate server. It is commonly used for embedded devices, data analysis, education and more [(*About SQLite*, n.d.).](#sqlite)

## Figma

A collaborative design tool for teams and individuals to create and share any high-quality application interfaces [(Husbands, 2022).](#figma)

## Git and GitHub

Git is a very popular version control system that records changes in your ongoing project repository. GitHub on the other hand is an online database that provides hosting services for Git repository [(User, 2022)](#git).

## Visual Studio Code

A free, open-source, code editor developed by Microsoft for building modern applications which supports multiple programming languages, provides intelligent code completion and debugging tools and also includes a built in Git support [(*What Is Visual Studio Code?*, n.d.).](#vscode)

## Microsoft office

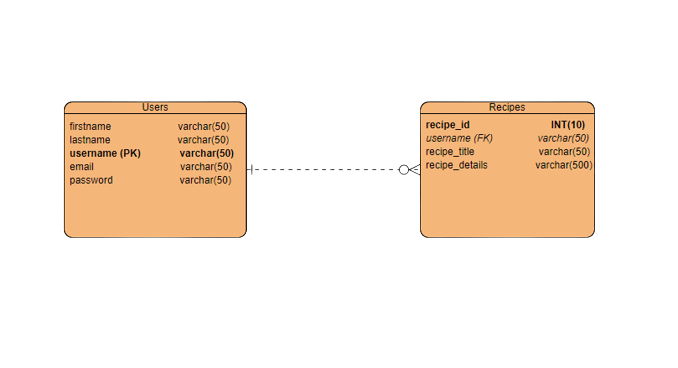
A combination of multiple productivity applications for personal computers designed by Microsoft specifically for business and documentation purposes [(Techopedia, 2020).](#microsoft)

## Tkinter

A python module for creating graphical user interface (GUI) applications. It provides a simple and easy to use interface with its toolkit making it easy to use for creating windows, dialog boxes and creating desktop applications [(ActiveState, 2022).](#tkinter)

# Conceptual Design

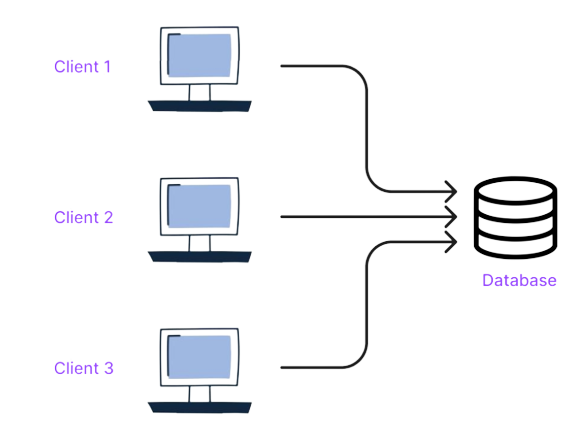
The recipe application follows a one-to-many relationship between the two entities users and recipes and indicates a strong relation between each other [(*How to Identify a Strong Vs Weak Relationship on ERD?*, 2013b).](#erdiagram) The primary key of both entities is shown in a bold letter as well as with a label of (PK) and the foreign key is shown in italics with a label of (FK) as given in the figure below.



### Figure 6: entity relation diagram of recipe application

# System Architecture

The recipe application is created based on two tier system architecture, which splits the application into two logical layers: client tier(presentation) and a server tier (application and data management) [(Techopedia, 2013).](#twotierarchitecture) The presentation side displays ingredients, user information, instructions, user inputs, searches and viewing recipes. The server tier handles the processing and storage of recipe information, such as retrieving recipes from databases updating, creating a new user, deleting and creating recipes.

****

### Figure 7: two tier system architecture

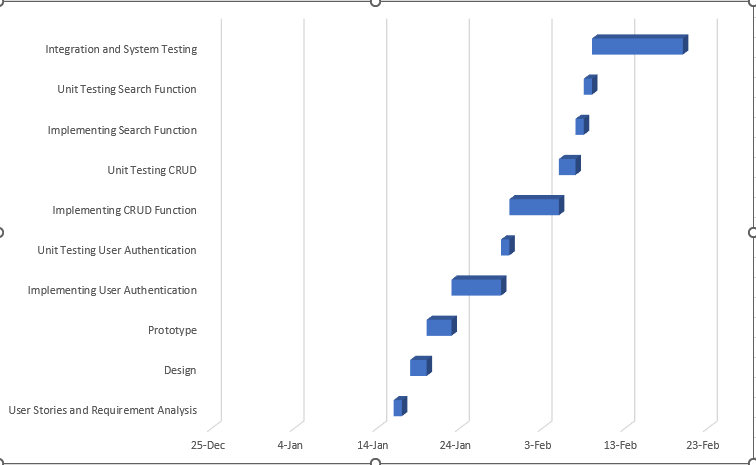
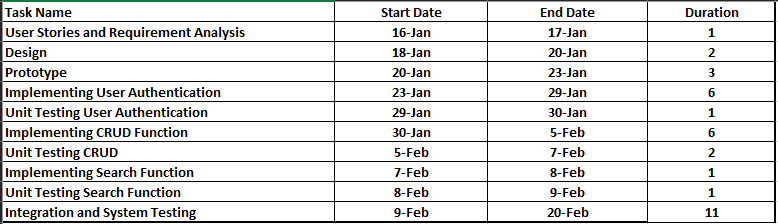
In this architecture, the client and server communicate directly with each other, allowing for simple and direct access to recipe information.

# Project Plan

The project plan for this simple recipe application with SQLite Database is a detailed roadmap that outlines the tasks and timelines for developing the application. The plan includes the following phases:

* User Stories and Requirement Analysis
* Design
* Prototyping
* Implementing User Authentication
* Unit Testing User Authentication
* Implementing CRUD Functionality
* Unit Testing CRUD Functionality
* Implementing Search
* Unit Testing Search Functionality
* Integration and System Testing

The plan was further broken down into series of smaller tasks that align with each other, and includes the spans from Jan 16th to Feb 19th. In addition, the plan includes testing phase for each feature that was implemented to guarantee that each feature is functional, before the integrating it to the application, and is tested again to ensure the application is fully functional and meets the requirements. A progress chart was implemented so that the development team can stay organized, on track and ensure the completion of the project.



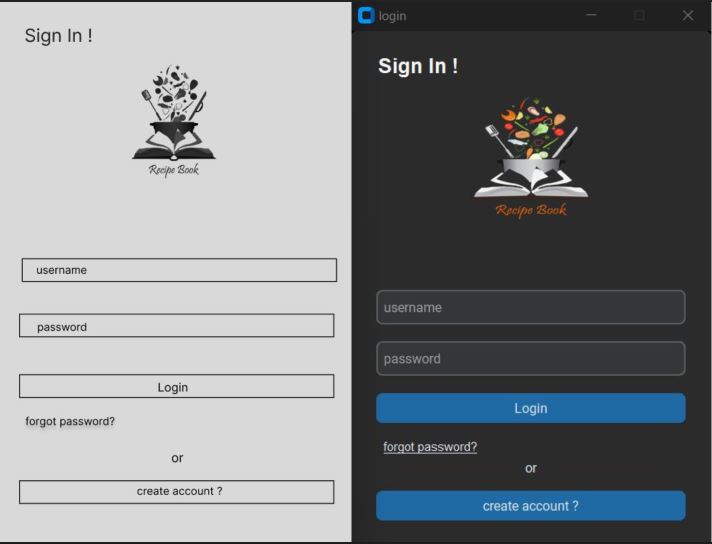
### Figure 8: Gaant Chart work of progress

# Prototypes

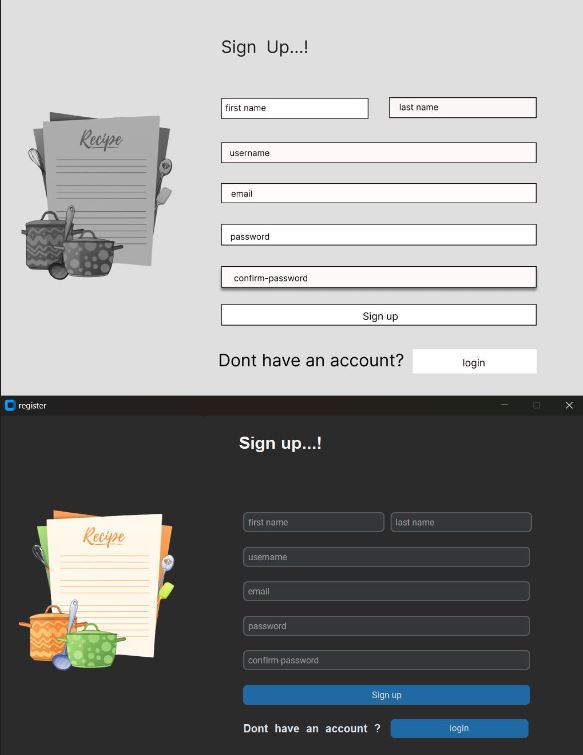
Prototype is an early model or sample of a product that is used to test and refine its design and functionality before proceeding with the full development of the application.

To create a prototype, the development team used Figma to create a visual representation of the application interface and was used to gather feedback from the client, to identify areas for improvement and used the feedbacks to refine the design and functionality of the application interface to ensure it meets the needs and requirements of the users.

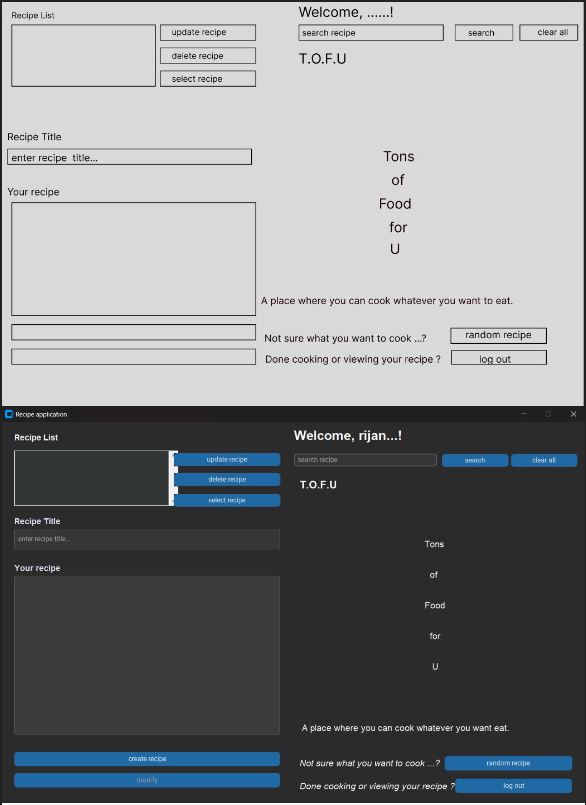
Once the prototype was finalized, it was used as a reference, for the development team to build an actual application according to the designs below:



### Figure 9: low-fidelity and high-fidelity prototype of Login UI



### Figure 10: low-fidelity and high-fidelity prototype of Registration UI

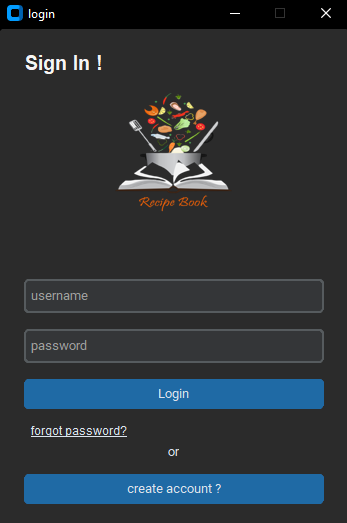


### Figure 11: low-fidelity and high-fidelity prototype of main application

# Developed System

The deployed recipe application leverages the developer’s expertise in SQLite, Python and tkinter to provide user authentication, recipe database CRUD, search, and password reset capabilities.

## Login Page



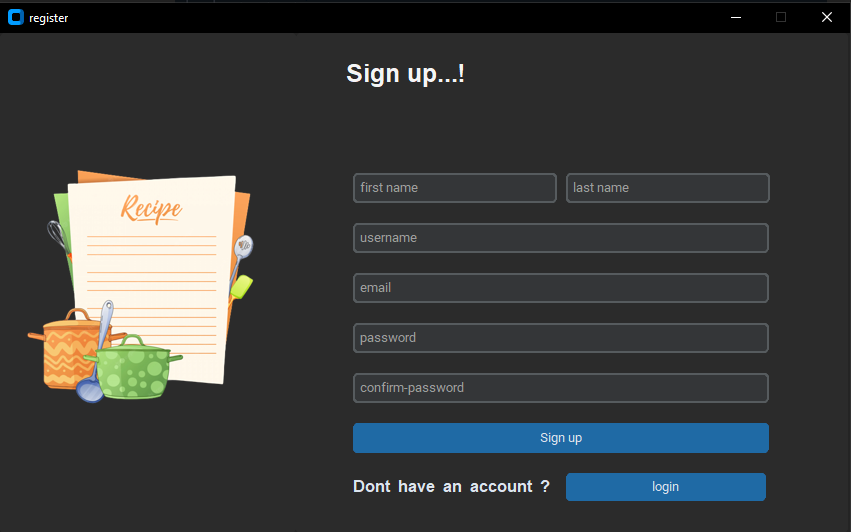
### Figure 12: Login page

A login page verifies user ownership and grants access to sensitive information/resources by comparing username and password to the database information. It also handles exception as shown below.



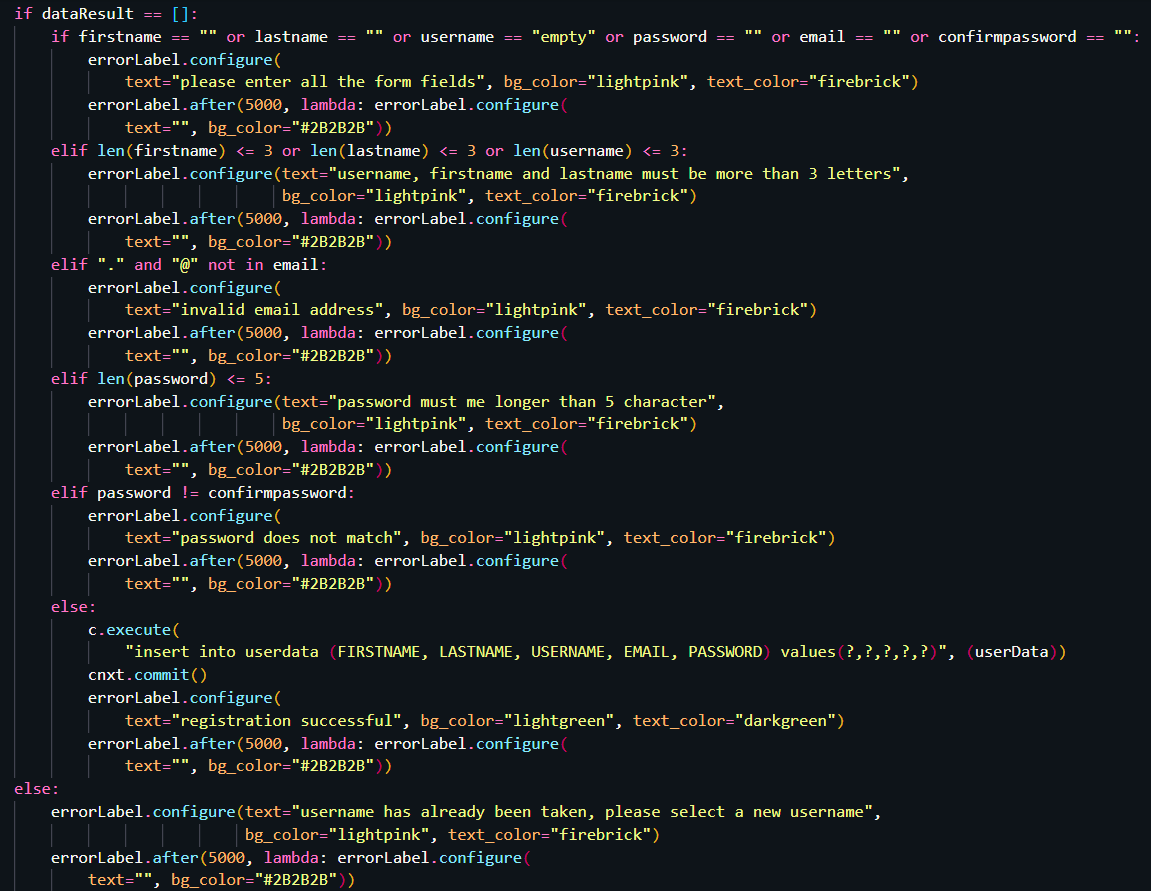
### Figure 13: login-page exception handling

## Registration Page



A registration page is used to collect information from users and create new accounts for them on the application. Its purpose is to gather necessary information to create a unique profile for a user. This information includes first name, last name, username, email address and password.

A user account in an application allows access with unique credentials and stores user information and recipes for personalized experience. It requires validation of user inputs



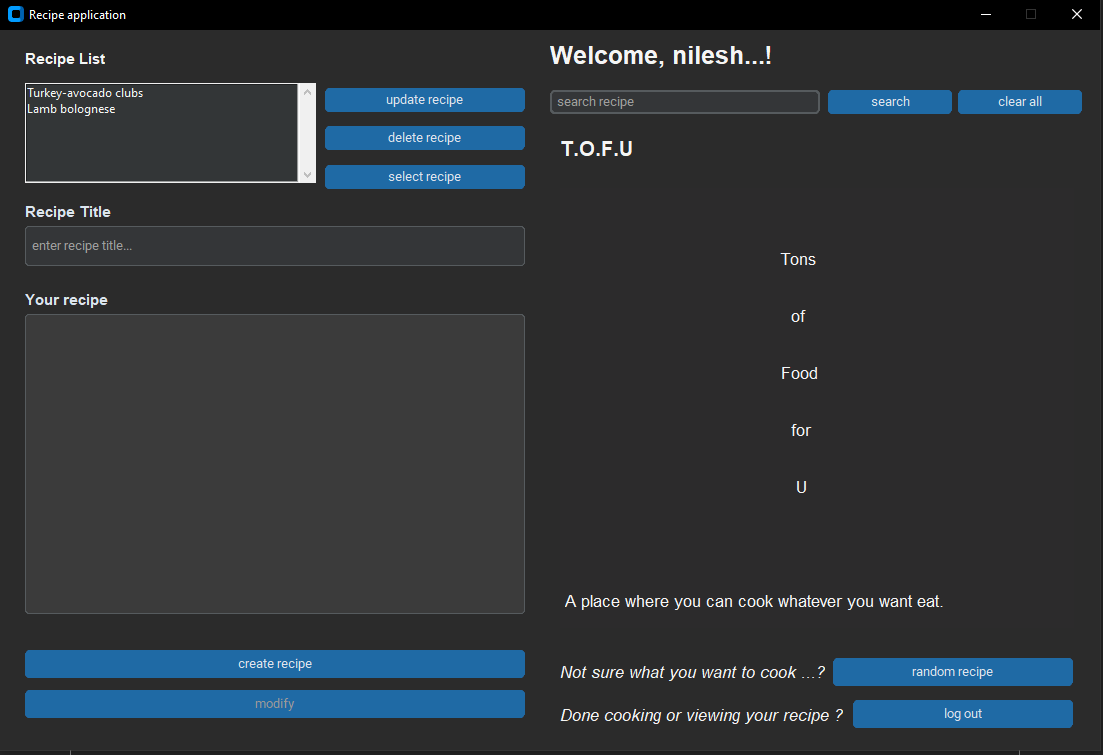
### Figure 14: registration validation

## Recipe Application Page

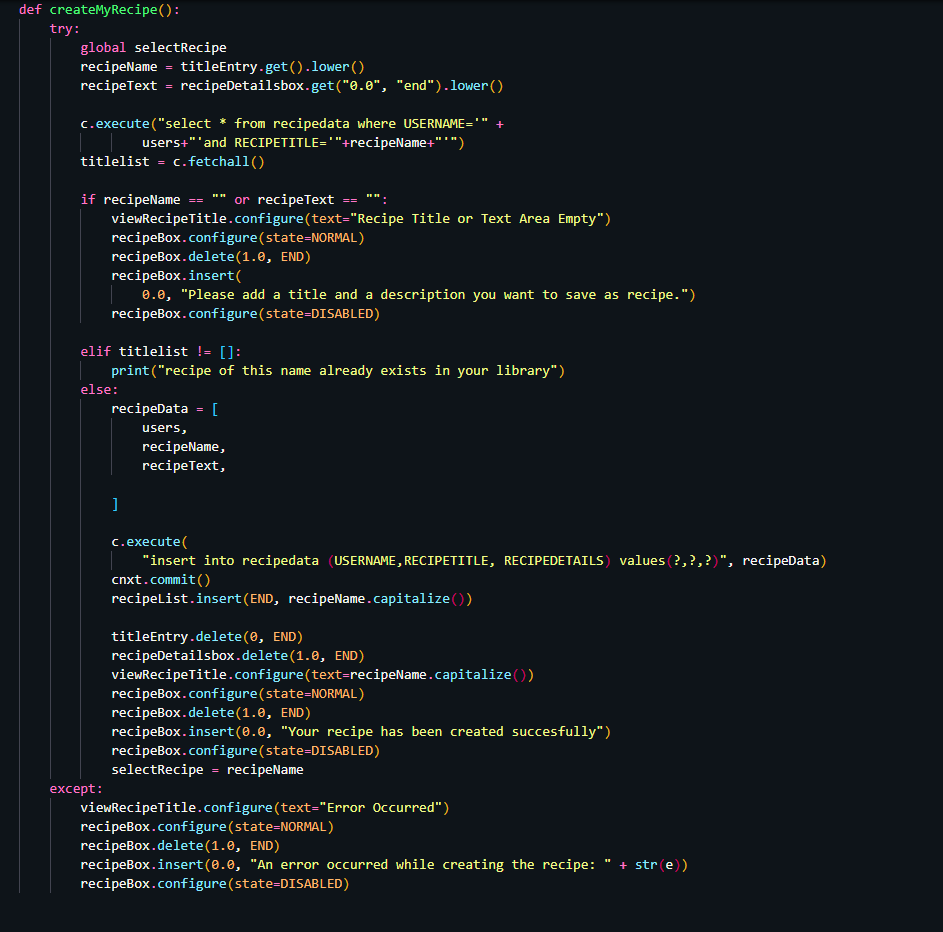
The main recipe application made with python, tkinter and SQLite is only accessible after a user is registered and logged in.

The recipe application’s GUI offers users recipe management with mandatory fields in its CRUD functionality, displaying appropriate error or success messages based on input validity to ensure data accuracy as shown in figures below. It also has a search functionality implemented to aid users in searching recipes by name and a functionality to generate a random recipe for users if they do not know what to cook from an already existing database.

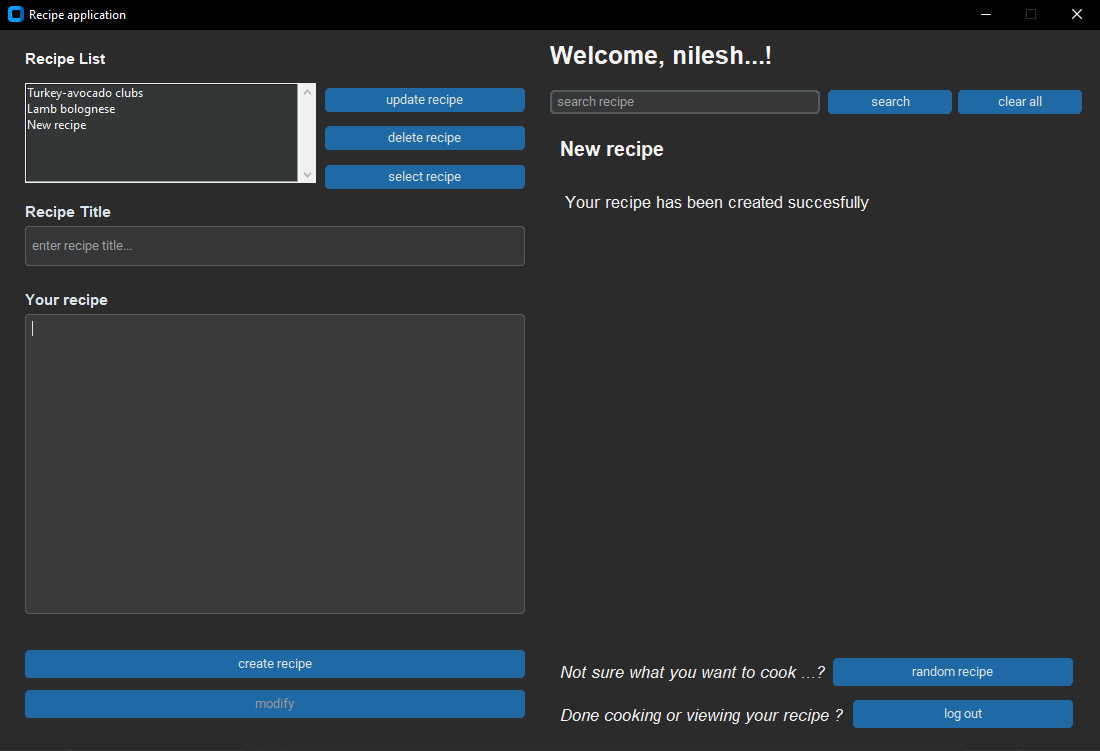
This ensures that the application meets the basic needs and requirements of home cooks and professional chefs



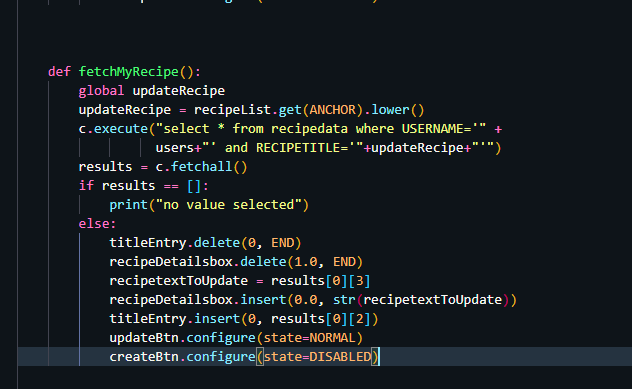
### Figure 15: Recipe Application Page



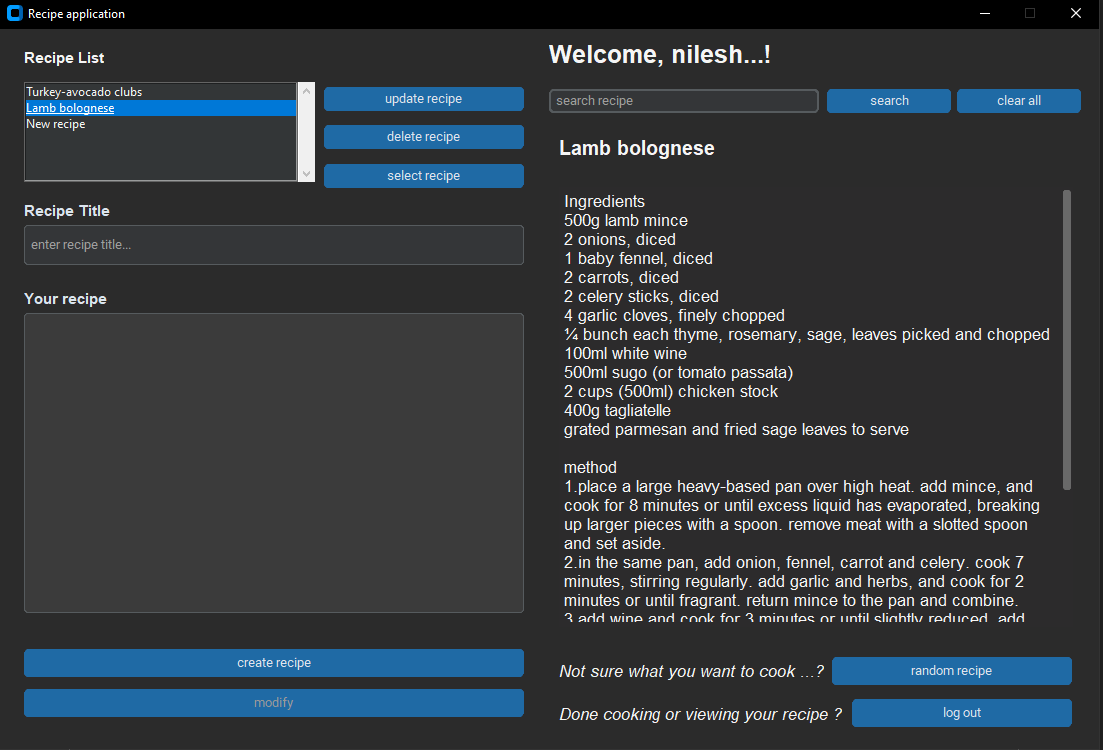
### Figure 16: Code Snippet for Recipe Creation



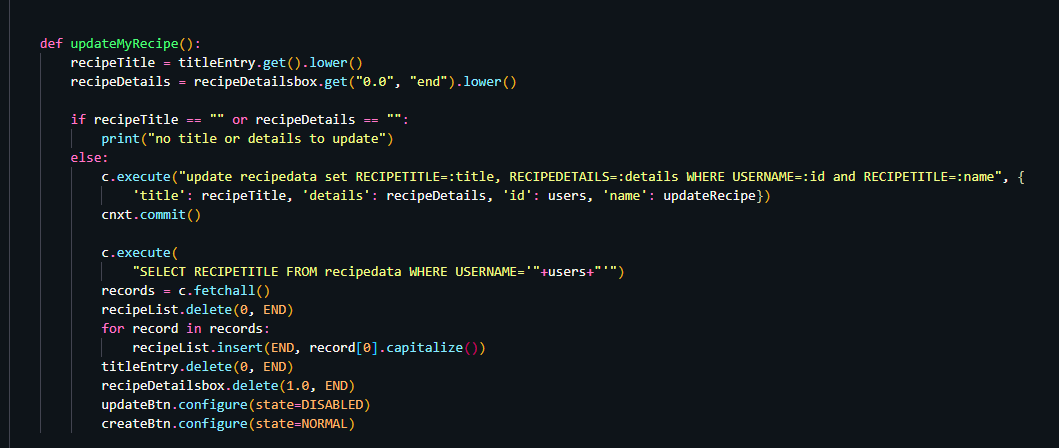
### Figure 17: Ui Message when Recipe is successfully created



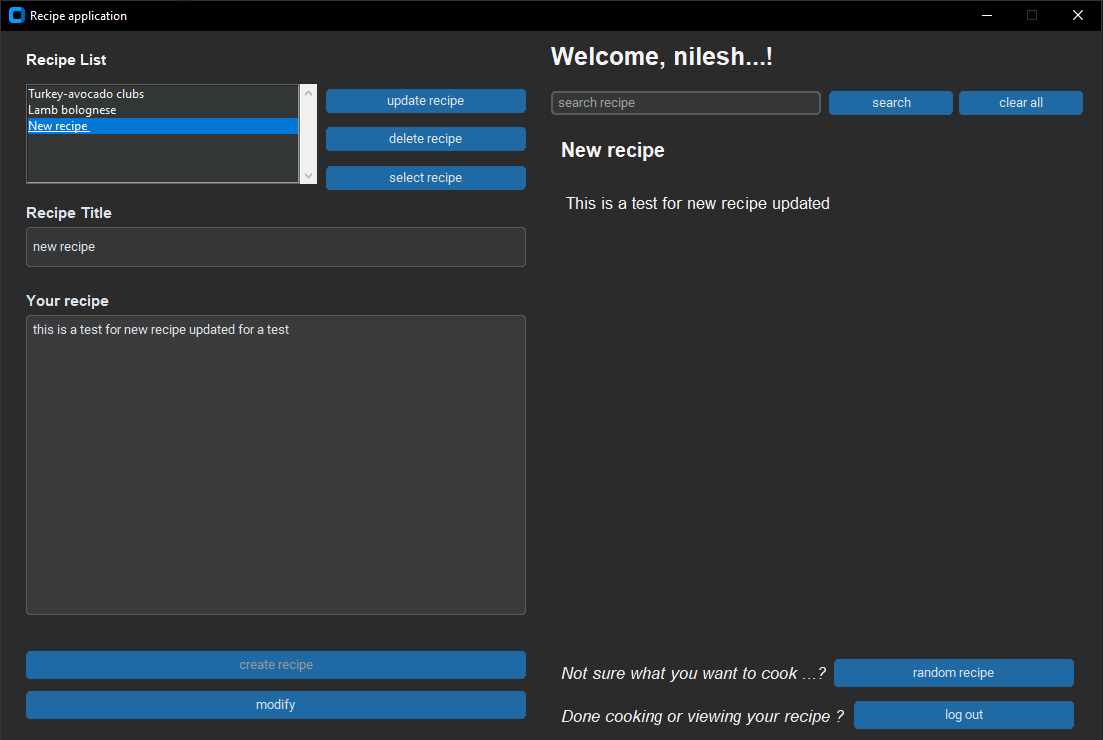
### Figure 14: Code Snippet for Fetching Recipe



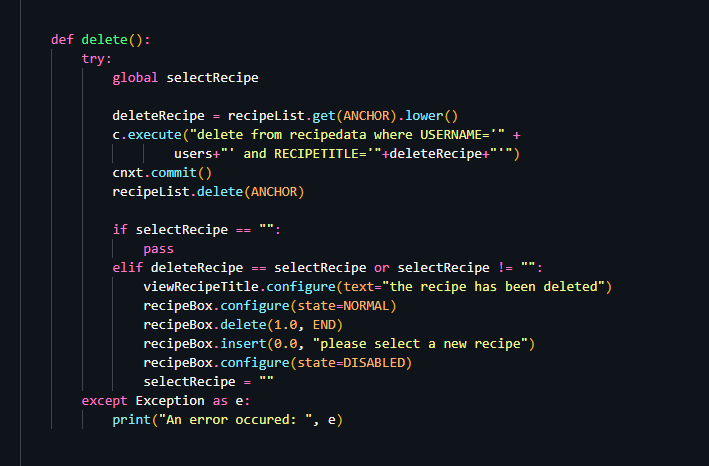
### Figure 18: How the recipe is displayed after selecting a recipe and clicking on select recipe button

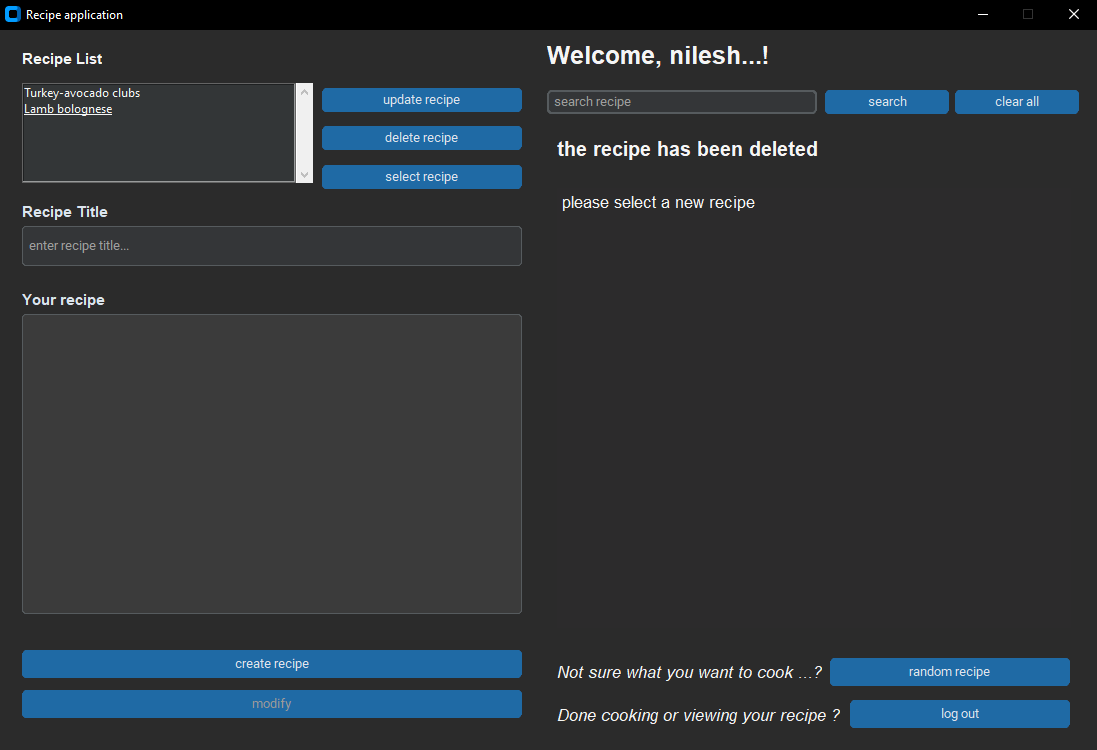


### Figure 19: Code Snippet for Updating recipe



### Figure 20: UI changes for Recipe after being updated

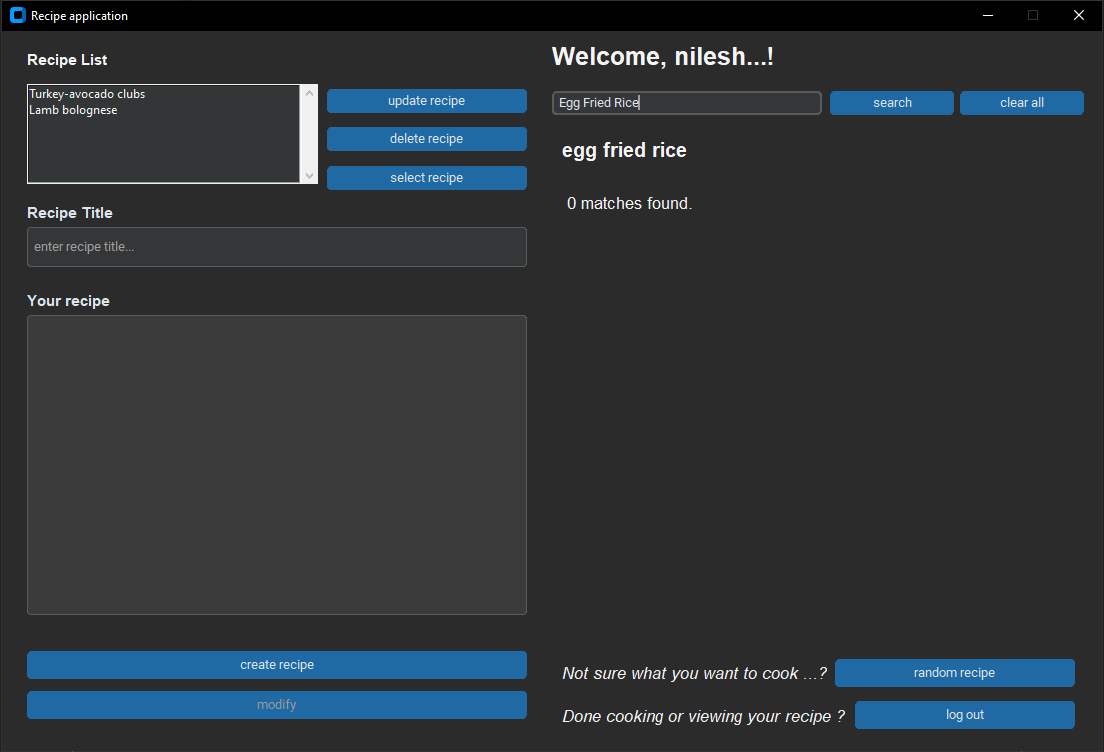
Figure 21: Code Snippet for Recipe Deletion



### Figure 22: UI changes after a recipe has been deleted



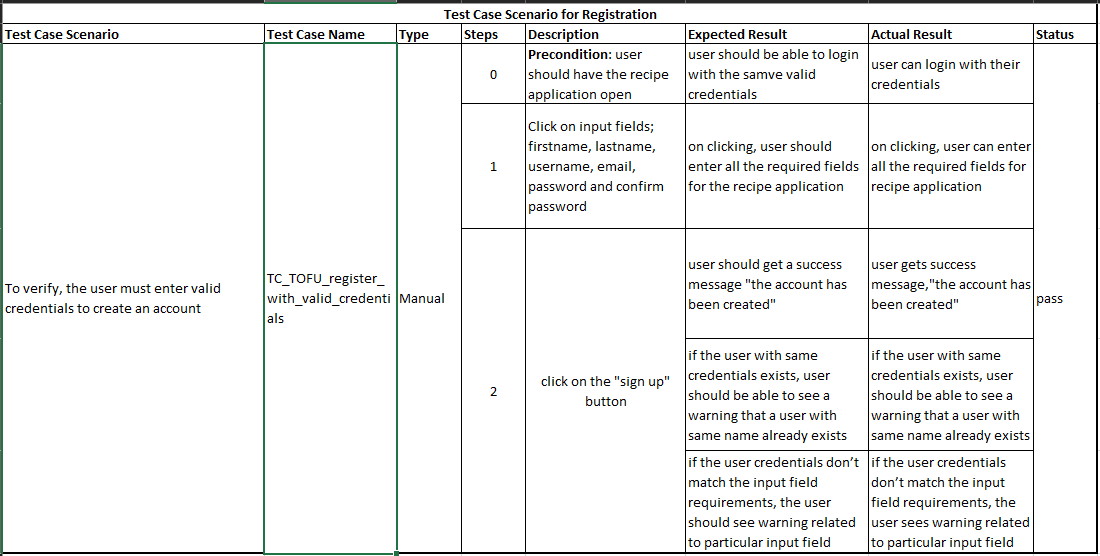
### Figure 23: Code Snippet for searching a recipe



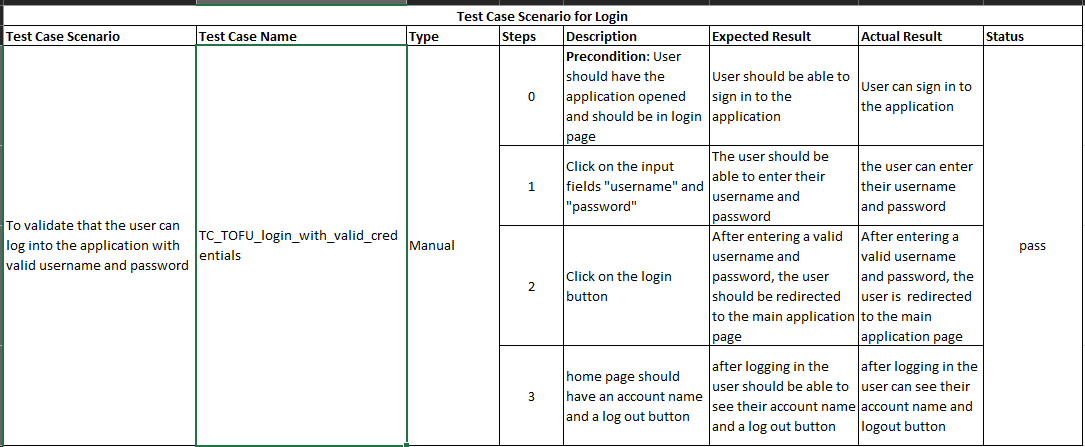
### Figure 24: UI result when 0 matches for recipe is found

# System Testing

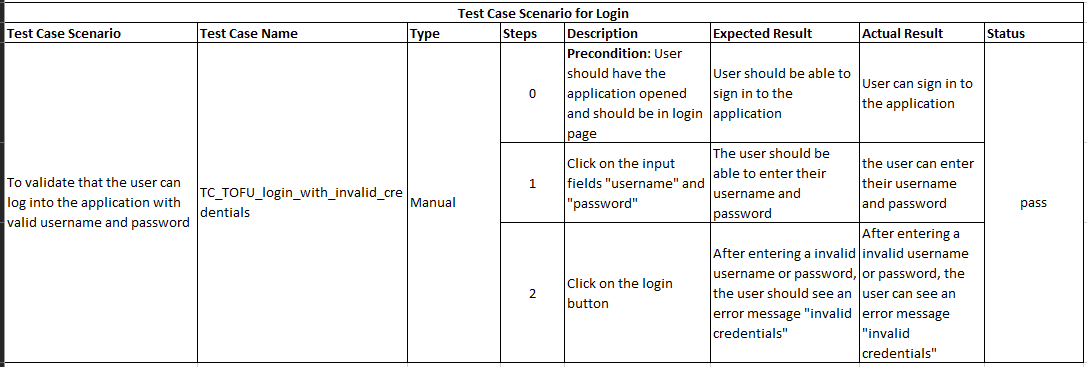
Unit Testing, a form of white box testing, was used to perform a system testing for each and every functional component of the application before integrating them into a fully functional application [(Richardson et al., 2020).](#whiteboxtesting) After the integration, it was tested again to ensure the components were working together correctly and as expected as shown in the figures below.



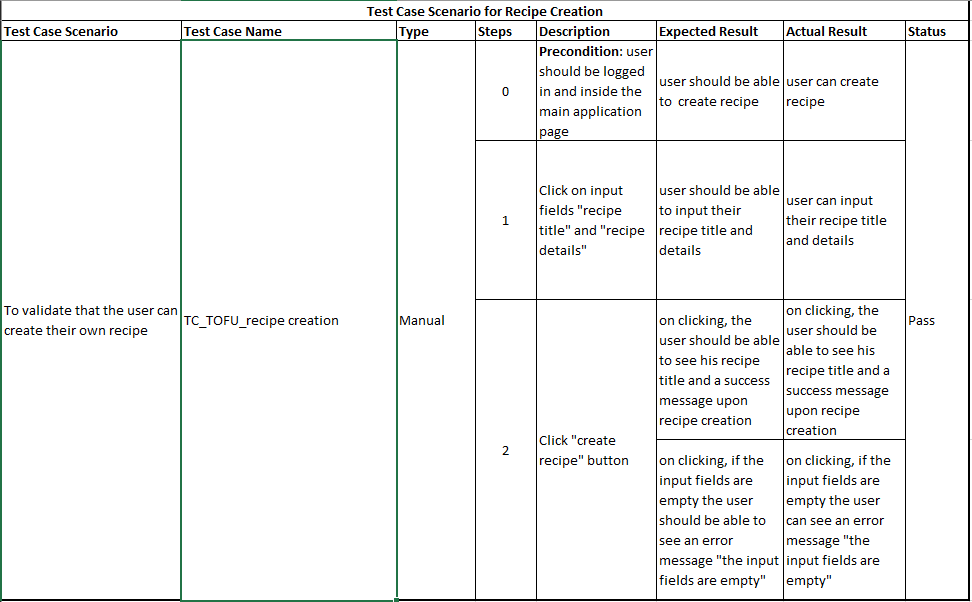
### Figure 25: test case scenario for registration



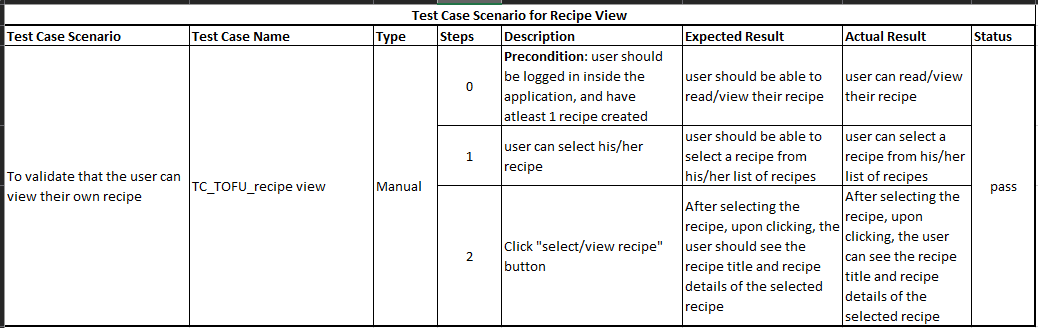
### Figure 26: test case scenario for login with valid credentials

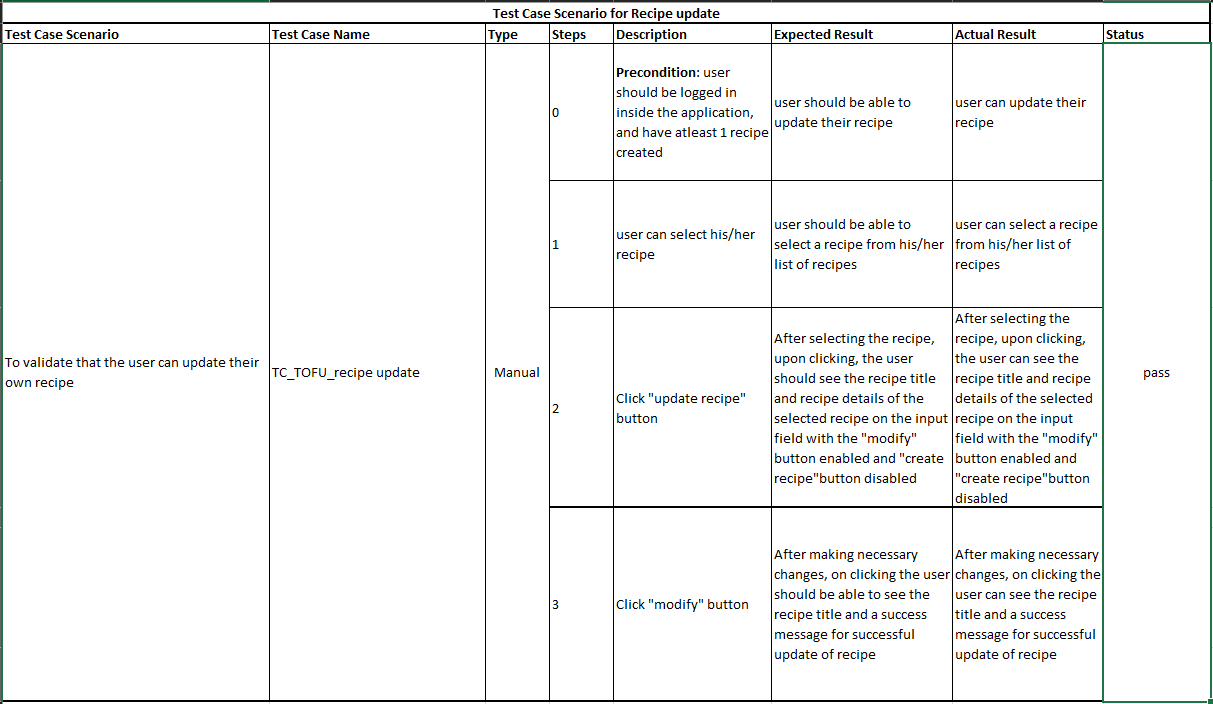


### Figure 27: Test case scenario for invalid login

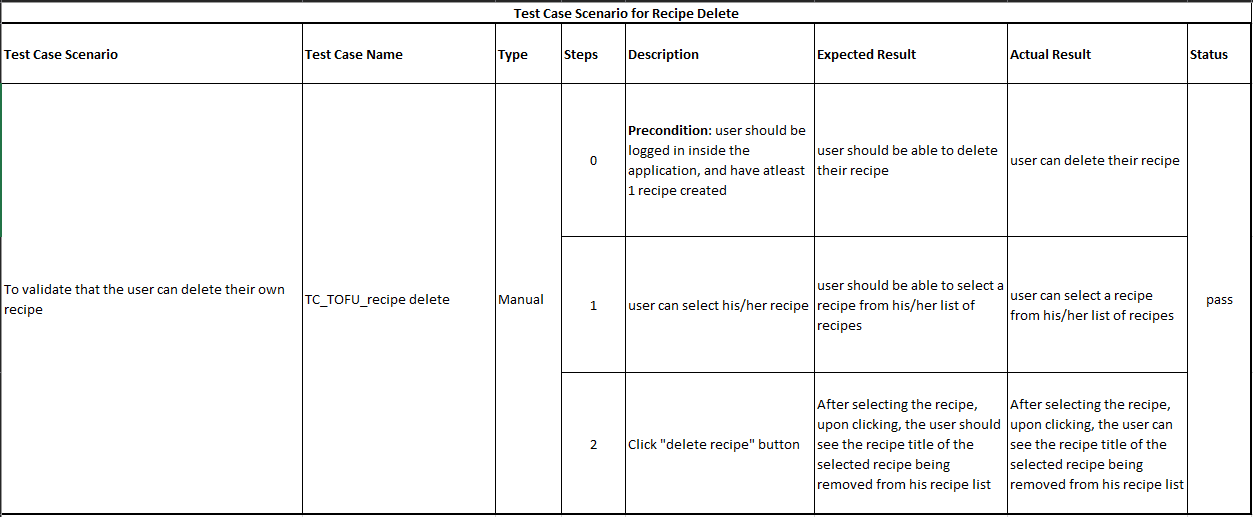


### Figure 28: Test case scenario for recipe creation

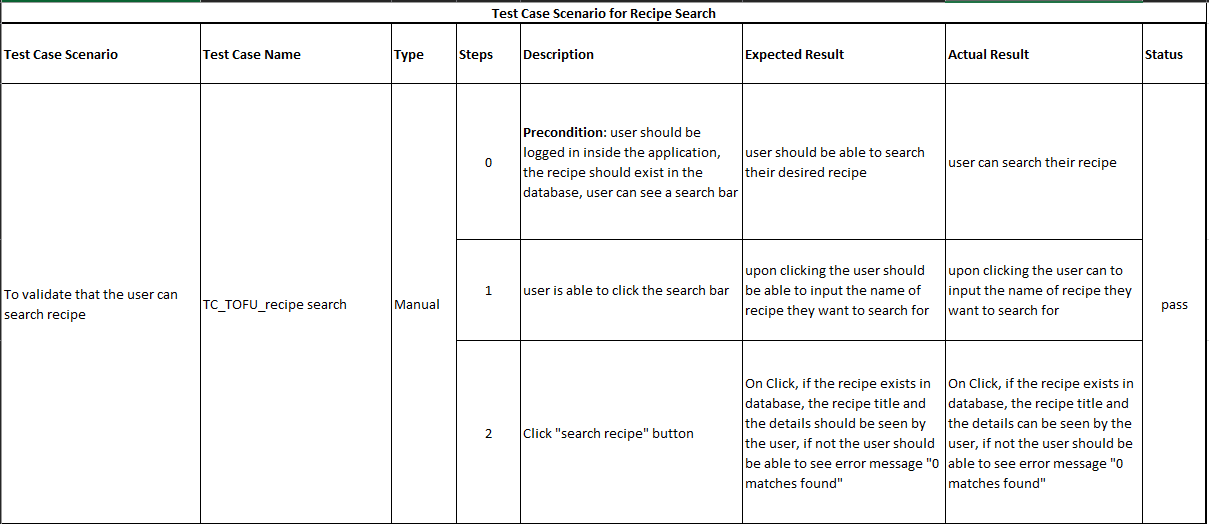
Figure 29: Test case scenario for recipe view



### Figure 30: Test case scenario for recipe update



### Figure 31: Test case scenario for recipe Delete

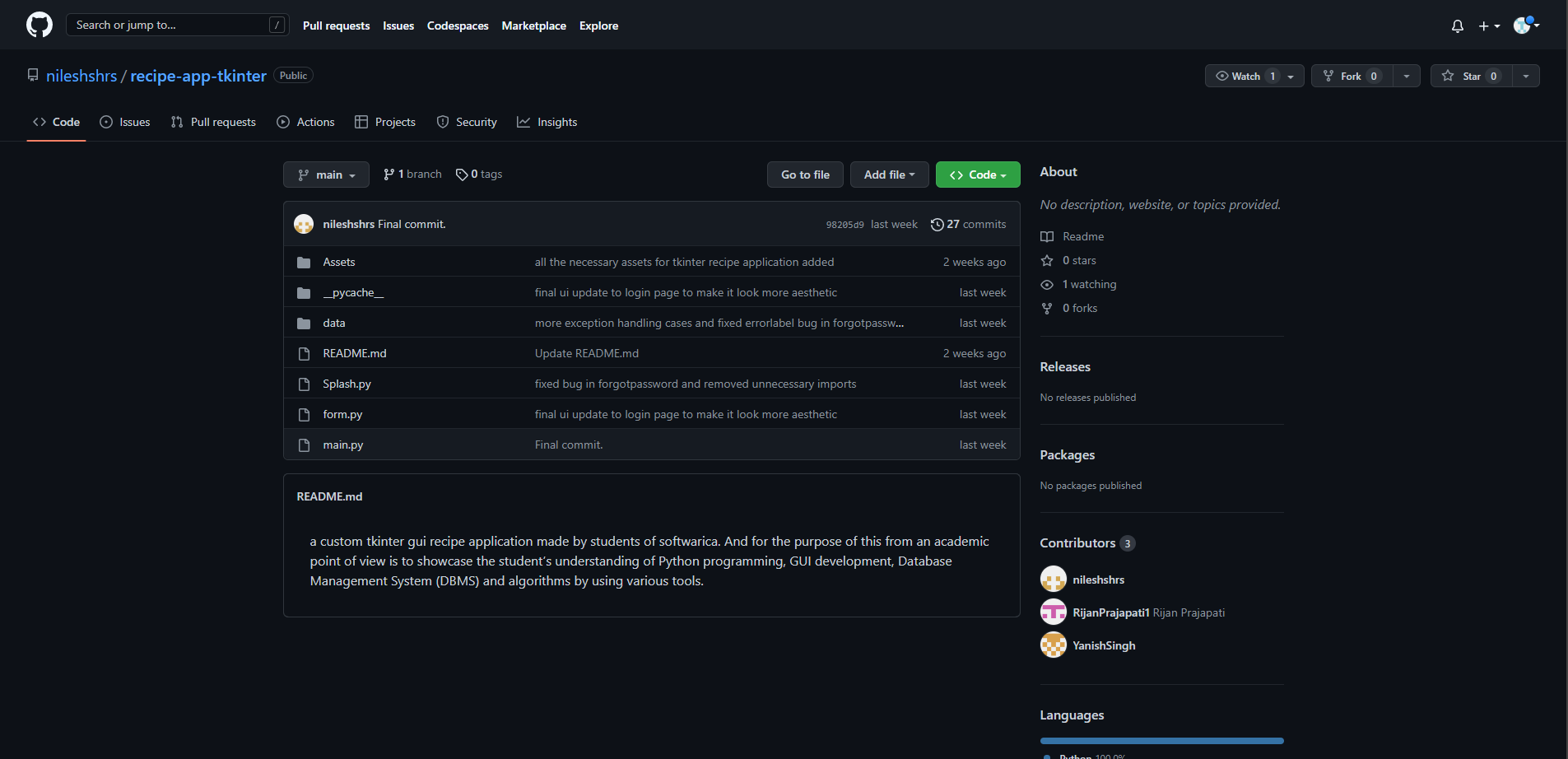


### Figure 32: Test case scenario for recipe search

# Version Control

* **GitHub**

<https://github.com/nileshshrs/recipe-app-tkinter.git>



### Figure 33: GitHub Repository

* **YouTube**

# Conclusion

In conclusion, the development of a tkinter application was a very exciting project. The development process of the application was optimized by using Agile methodology, which focuses on an iterative and incremental approach to development. This ensured that the development team stayed on track with the project requirements and deliver high-quality application.

The application’s system architecture, which includes the frontend and backend components, was designed to ensure scalability and maintainability. The frontend, which includes GUI, was designed using a conceptual approach and promotes a very friendly and positive user experience. The backend was designed to process all the user inputs, validate them and perform CRUD operations on the SQLite database.

By utilizing the user stories, the application was designed to meet the specific needs of its audience. User stories were used as a guideline in the development process to prioritize features and ensure that the application meets its intended purpose, it also served as the basis for the system requirements, which helped guarantee that the development team focused on meeting the end-user needs throughout the development process

Overall, the development of this application offered an excellent opportunity for us as students to gain hands-on experience in software development, as well as enhance our knowledge and skills in database management, graphical user interface design, and other related areas.

# References

*Main Points of Food Recipe App Development*. (n.d.). agilie.com. Retrieved February 6, 2023, from <https://agilie.com/blog/main-points-of-food-recipe-app-development>

Pfeil, M. (2010, October 22). *What is Data Persistence and Why Does it Matter? | Datastax*. DataStax. Retrieved February 6, 2023, from <https://www.datastax.com/blog/what-persistence-and-why-does-it-matter>

Agile Alliance. (2023, February 3). *What is Agile? | Agile 101 | Agile Alliance*. Agile Alliance |. Retrieved February 8, 2023, from <https://www.agilealliance.org/agile101/>

*Iteration*. (2022, May 6). Retrieved February 8, 2023, from <https://www.productplan.com/glossary/iteration/>

*What is Python? Executive Summary*. (n.d.). Python.org. Retrieved February 12, 2023, from <https://www.python.org/doc/essays/blurb/>

*About SQLite*. (n.d.). Retrieved February 12, 2023, from <https://www.sqlite.org/about.html>

Husbands, J. (2022, January 4). *The complete guide to Figma and why it’s the future of design*. Medium. Retrieved February 13, 2023, from <https://bootcamp.uxdesign.cc/the-complete-guide-to-figma-and-why-its-the-future-of-design-a82839d3b5b9>

User, D. (2022, July 1). *Git vs. GitHub: What’s the Difference?* Devmountain. Retrieved February 13, 2023, from <https://devmountain.com/blog/git-vs-github-whats-the-difference/>

*What is Visual Studio Code?* (n.d.). Educative: Interactive Courses for Software Developers. Retrieved February 13, 2023, from <https://www.educative.io/answers/what-is-visual-studio-code>

Techopedia. (2020, August 20). *Microsoft Office*. Techopedia.com. Retrieved February 13, 2023, from <https://www.techopedia.com/definition/20737/microsoft-office>

ActiveState. (2022, August 6). *What is Tkinter used for and how to install this Python Framework?* Retrieved February 13, 2023, from <https://www.activestate.com/resources/quick-reads/what-is-tkinter-used-for-and-how-to-install-it/>

*How to identify a strong vs weak relationship on ERD?* (2013b, May 21). Stack Overflow. Retrieved February 16, 2023, from <https://stackoverflow.com/questions/16675127/how-to-identify-a-strong-vs-weak-relationship-on-erd>

Techopedia. (2013, July 19). *Two-Tier Architecture*. Techopedia.com. Retrieved February 16, 2023, from <https://www.techopedia.com/definition/467/two-tier-architecture>

Richardson, L., Sillam, Y., Hasson, E., Hewitt, N., McKeever, G., McKeever, G., Rossi, E., & Hewitt, N. (2020, September 24). *What Is White Box Testing | Types & Techniques for Code Coverage | Imperva*. Learning Center. Retrieved February 17, 2023, from <https://www.imperva.com/learn/application-security/white-box-testing/>