



# Northeastern University

## PROJECT REPORT FOR CS 5200

### Online Recipe Management System

--- Anushka Tak

### Introduction

In 2020, due to COVID, many countries went into lockdown shutting down prominent businesses like food industry. Restaurants were shut down leading to people cooking more often for themselves at home. Now that people were working from home, they would cook more. Not all of us are born chefs and require guidance and recipes to cook meals. In today's world where we would rather ask Alexa about anything and everything than our mothers, I was motivated to build a Management system to support a website where users can search and share recipes. This can be extremely useful as it is not limited to certain cuisines. Being a programmer, if we lack resources to do anything, we can always create opportunities like this project.

The recipe system aims to help users try cooking new dishes. A user can look up recipes by names or by cuisines. In the recipe, one can expect to find ingredients, measurements and order in which to use. A user can also contribute to the system by sharing one's own recipe. By this, we can end up with a database that increases multifold on its own with less maintenance.

### Database

For the project, I created my own database with entering data using SQL Stored procedures. The database contains following tables, description mentioned (The finalized schema will be documented in Project Database):

- Users – captures user information like user name, email id, password, food preferences etc. for each user
- Recipe – consists of the recipe of food items entered by users, along with ingredients, cooking time, ready time, preparation time nutrition value associated with each recipe
- Feedback – any feedback given to our system by a user
- Comment – comments captures information like what is the comment on which recipe
- Cuisine
- Category

## Database Architecture

Since the data involved is highly relational, the chosen database language to support back end operations is SQL. MySQL Database can support complex queries to be performed on the data stored in the database. It has quicker response time and can be hosted on multiple servers handling the large throttle that may be the case in real time scenario.

## Technical Description

The database consists of

- Table Users - 12 users with login information
- Table Recipes - 11 Recipes of different cuisines, categories and contributors
- Table Feedback - 12 feedback to the system by various users
- Table Comments - 16 Comments to various recipes by various users
- Table Cuisine - with 10 options (e.g. Italian, Indian, Mexican)
- Table Category - with 8 categories (e.g. Breakfast, Comfort food, Baking )

Initially, I had proposed building a Flask application and launching a website to support this Management System. Due to some constraints, I finally deployed the Proposed Management System on Command Line Interface.

The Project runs its front end on Python, uses various Python libraries, e.g. pymysql, mysql, sys, for multiple functionalities. I have also modularized the project in a package consisting of different files for each functionality.

For any connection-based function, the main file calls the functions defined in the 'connection.py' file. Similarly, for any recipe based functions (enter, search, delete), the main file calls the functions defined in the 'recipe\_functions.py' file. Same is for comments (view, enter, delete) – comments\_functions.py and feedback (delete, enter) – feedback\_functions.py. Dividing the package in different modules enhances the code readability and makes debugging easy. The code has been provided with comments for future use.

For back end operations, we use python libraries to connect to the database. Once, connection is successful, objects like connection and cursors execute various SQL queries and produce results. The results are then displayed with either a success or user-friendly error message.

Python 3.7.3 version is being used.

Programming Objects used are Stored procedures – 1. To create tables in the database 2. To populate the database.

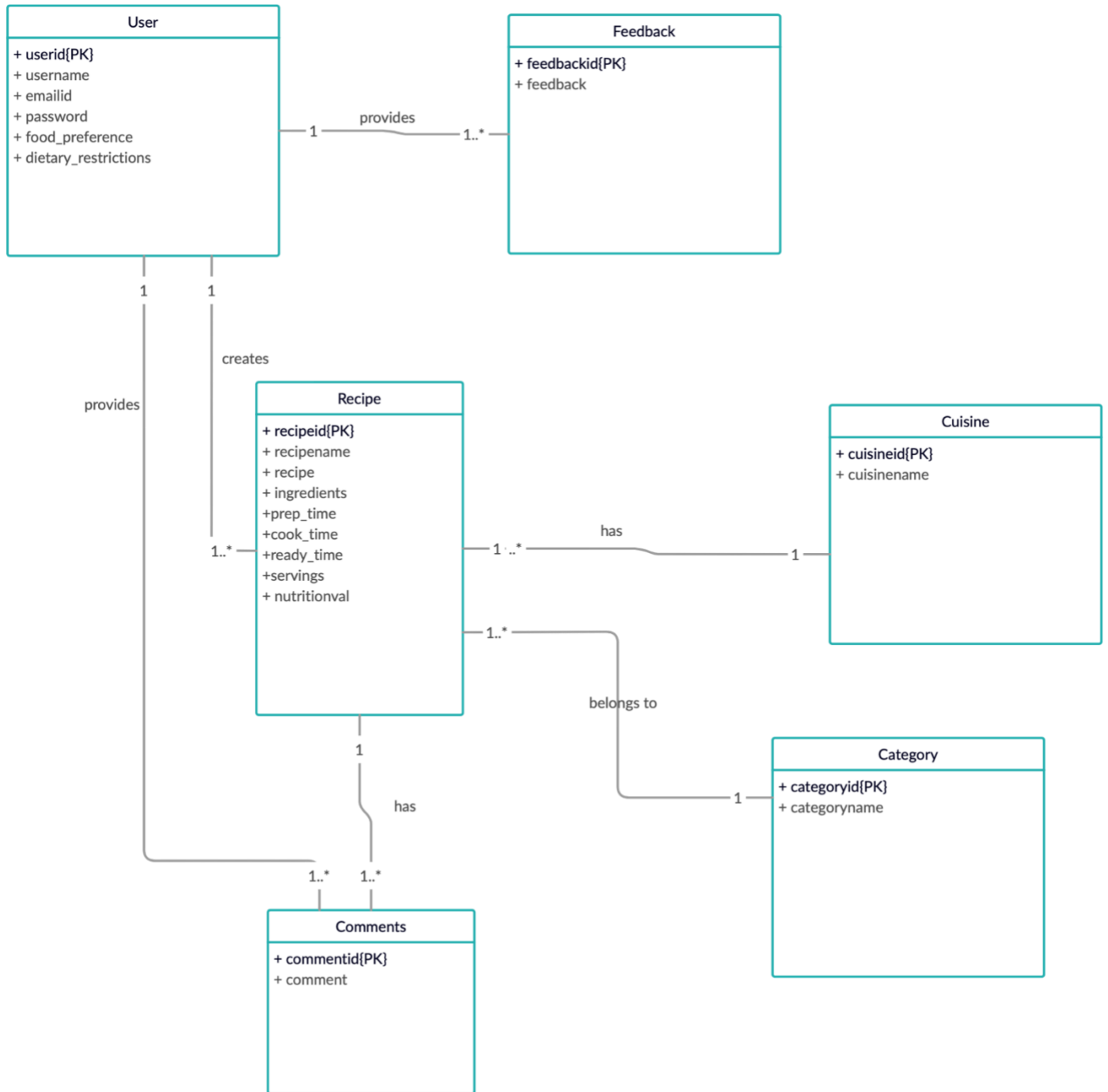
## CRUD Operations

The project is capable of performing these operations (some instances mentioned):

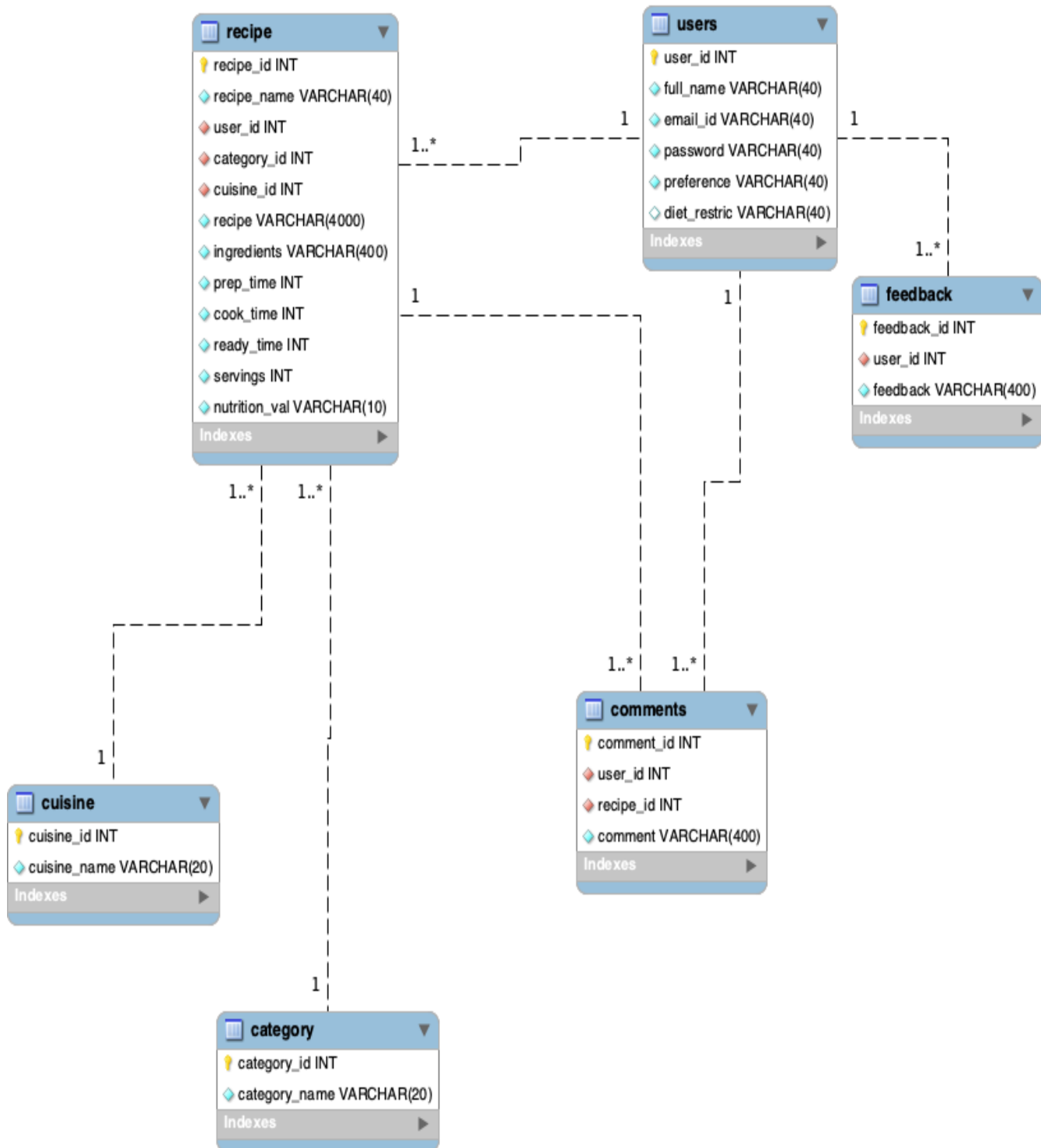
- Create - create a new user profile, creates a new recipe, creates new feedback, creates new tuples for new comments
- Read – look up existing recipes, looks up existing comments, looks up existing feedback, look up interesting facts about the data for further analysis, read log in information of existing users
- Update - update his own comments that he/she may have contributed in the past
- Delete - delete one's own recipe that he/she may have contributed in the past, deletes one's own comments, deletes one's own feedbacks, delete one's own account

As of this point based on project research, there seems to be no machine restrictions.

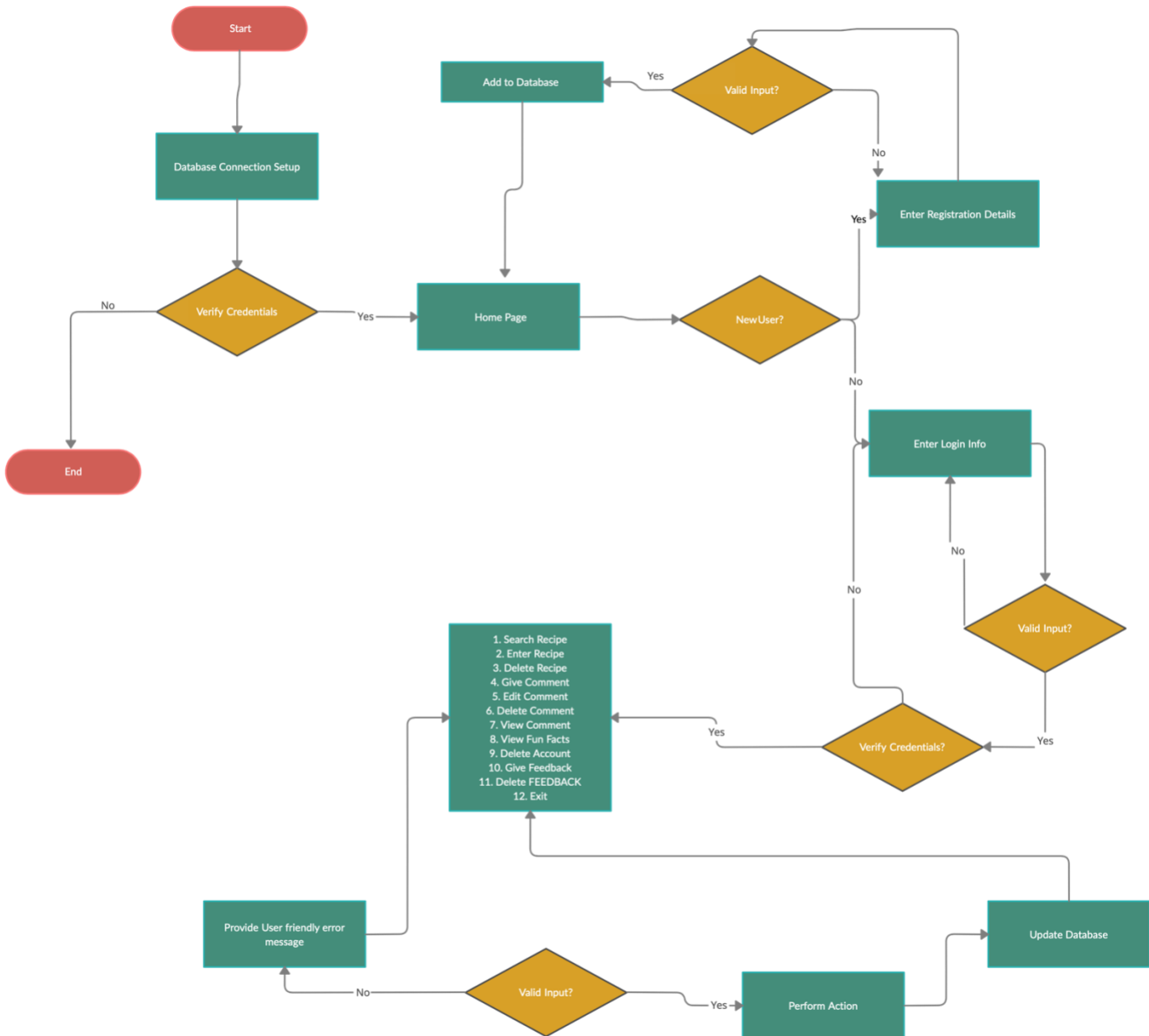
## Current Conceptual Design



## Logical Design (EER)

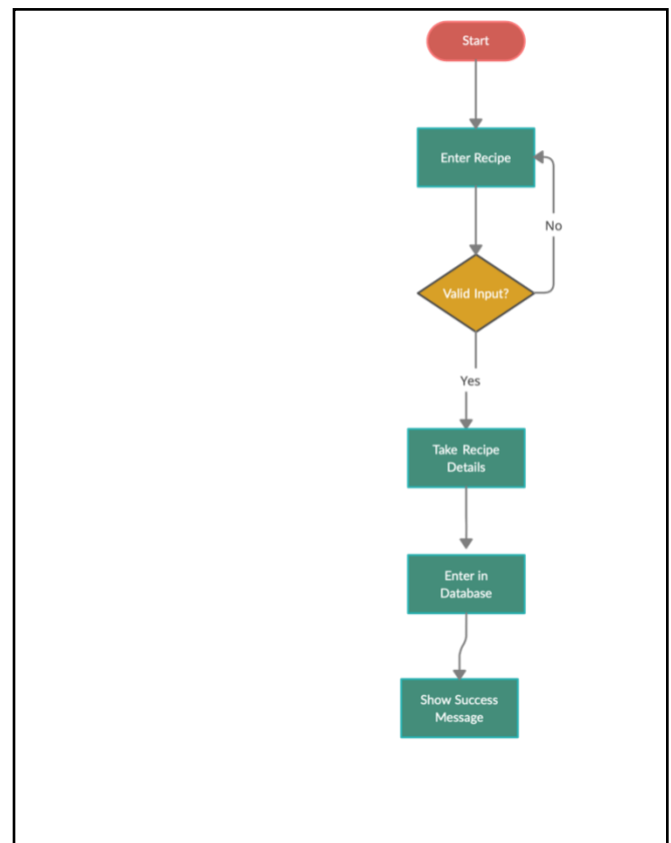
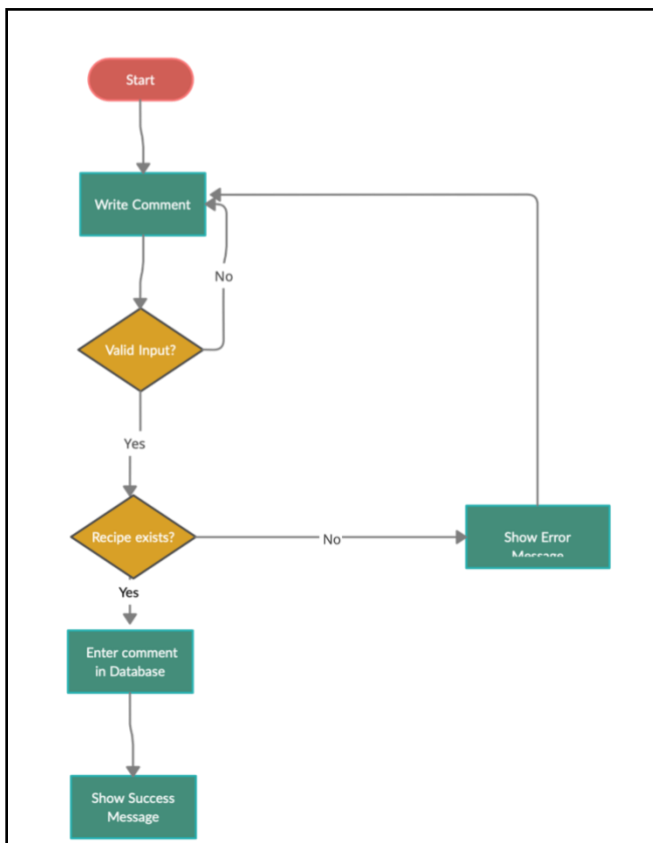
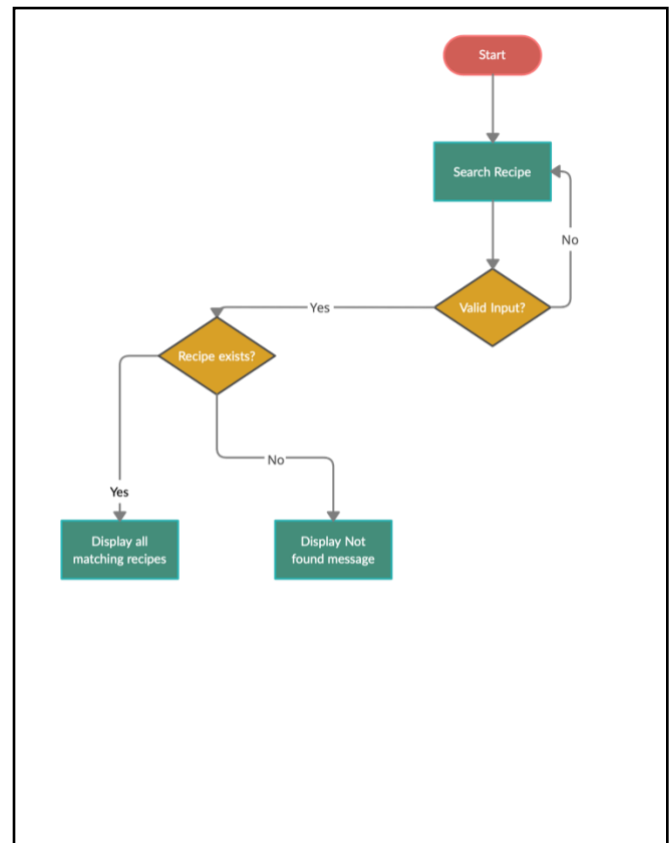
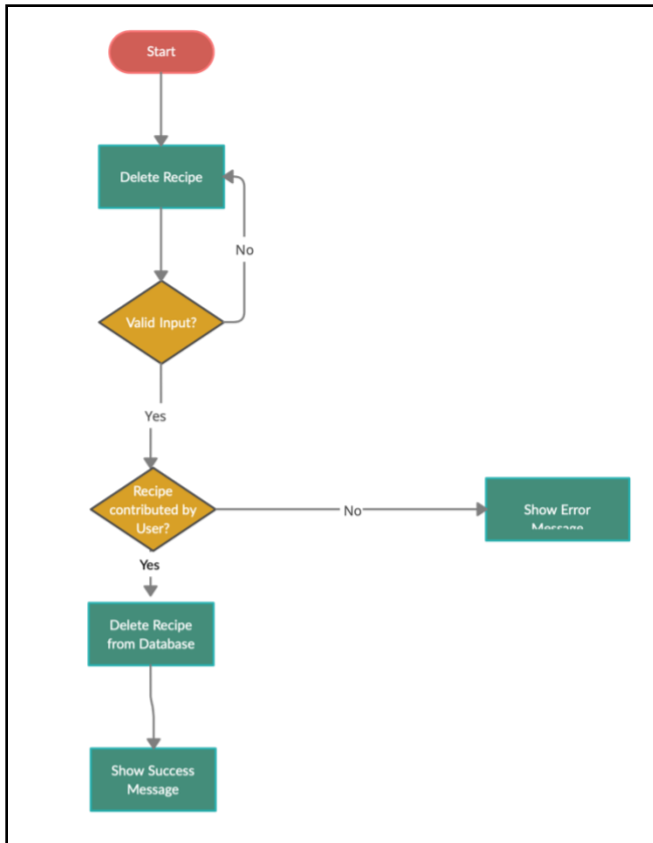


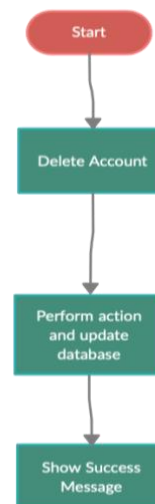
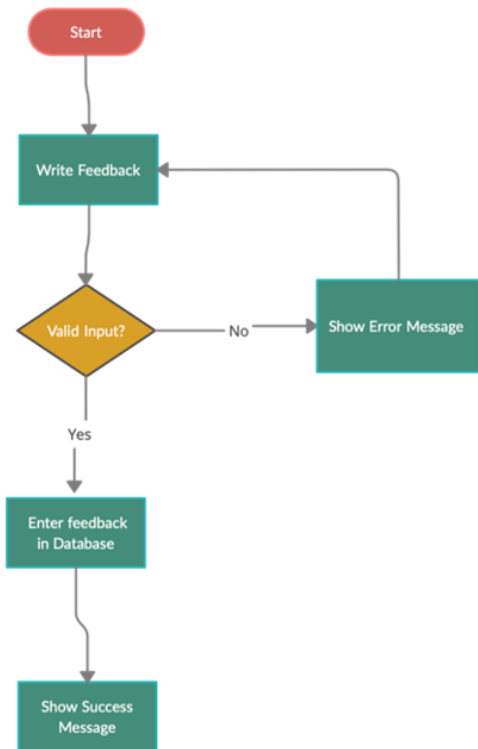
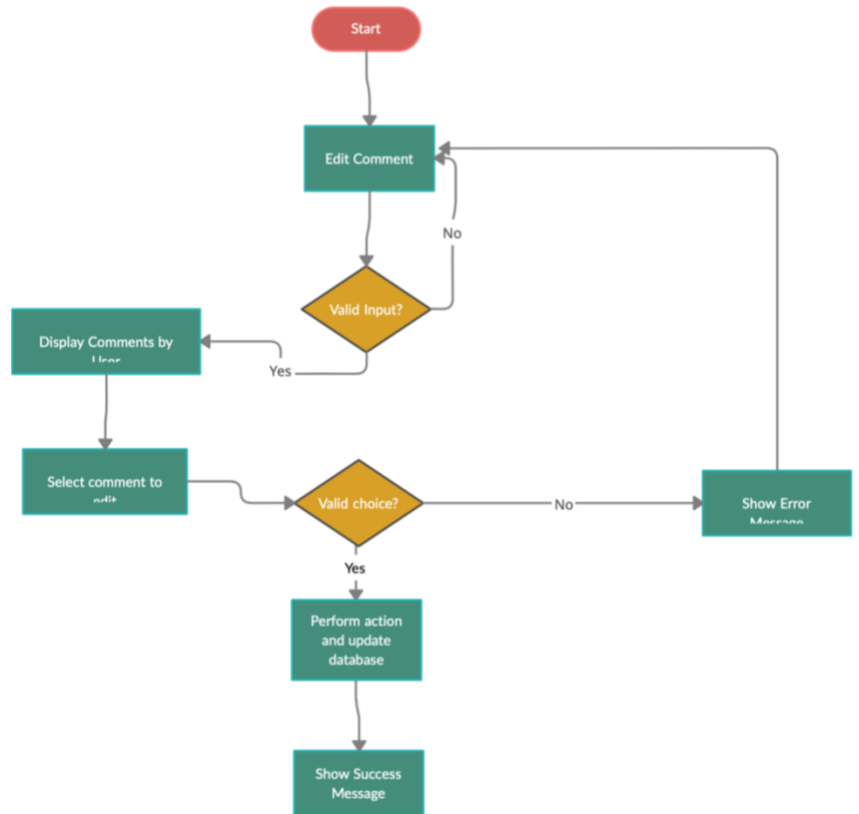
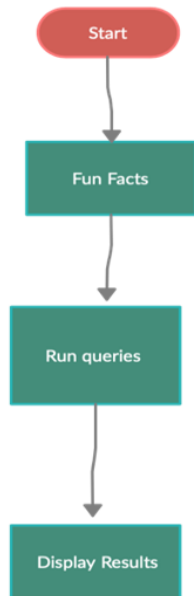
## Flow Diagram

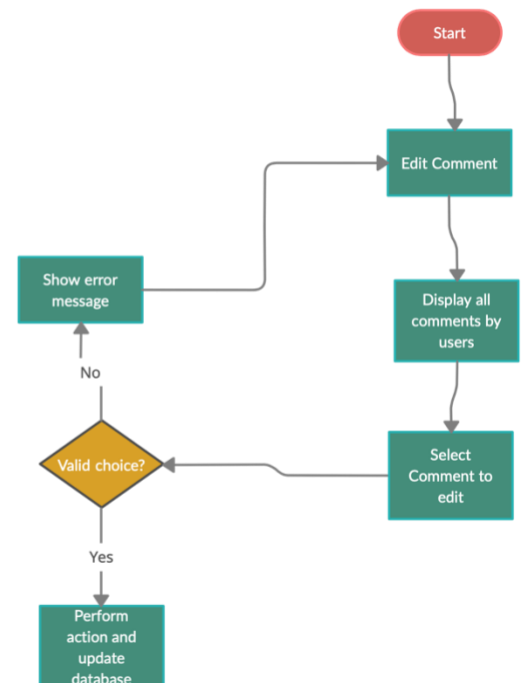
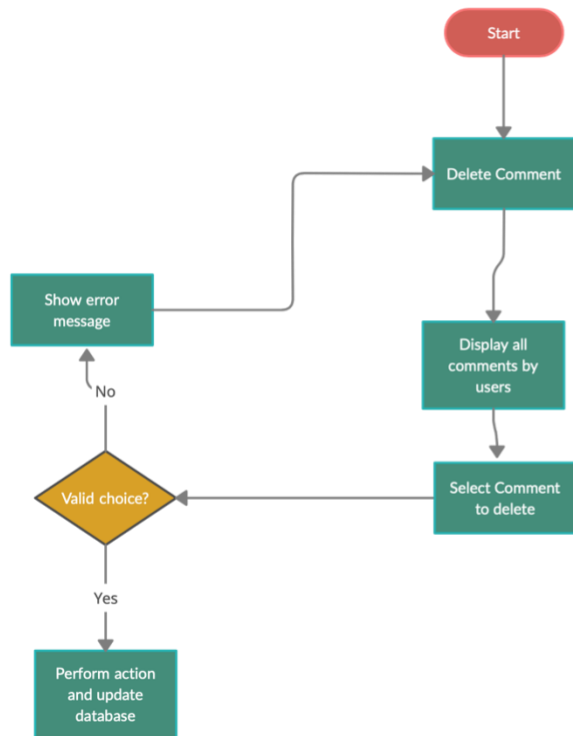
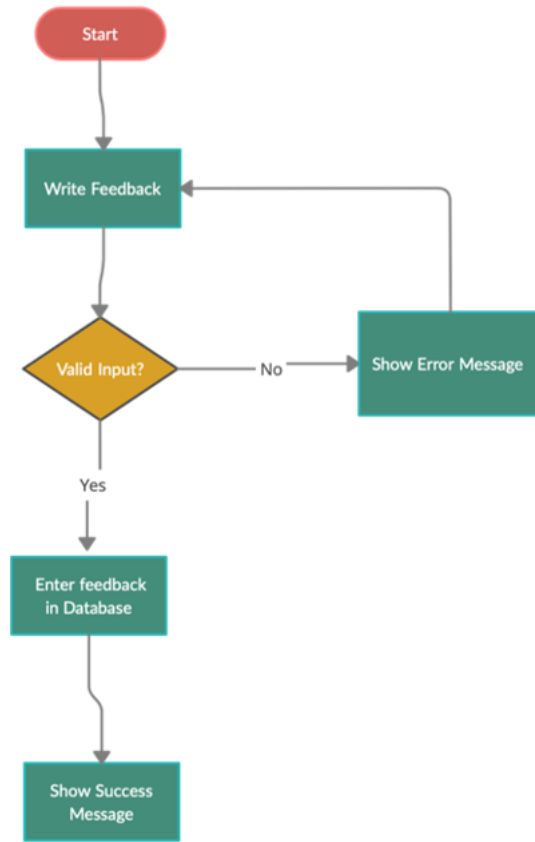
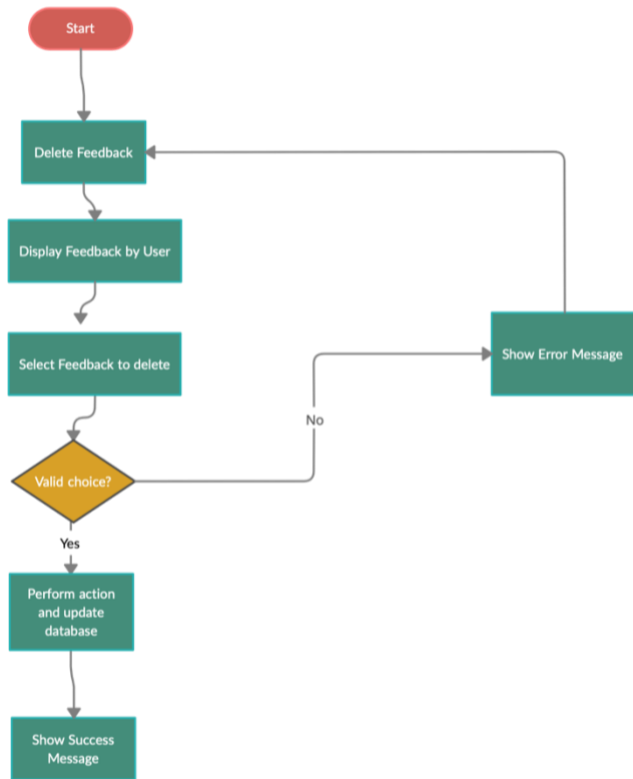


## Extended Flow diagram of each functionality

Here are the flow diagrams for each functionality. These functions come into execution from the menu at home page.







Note: All functions after successful action go back to the main menu.



## Read me:

1. Run the database export dump provided. This will ensure creating the database 'recipe\_management' and using it for the rest of the execution. Next, the dump script creates all the procedures in the script in your local system (MYSQL Workbench) and runs them, the script contains creating and running database programming objects (stored procedures) to create required tables for the management system and populate them.
2. Now, download the folder 'application code' (as submitted under Assignment), make sure all files are kept in the same package (recipe\_management.py, fun\_facts.py, recipe\_functions.py, feedback\_functions.py, comments\_funcitons.py, connection.py). These files define required functions to perform various operations based on user's choice. The main file is – **recipe\_management.py**. We need to execute and run this file in a terminal, use command 'python recipe\_management.py'. Make sure you run the terminal in the folder these files exist. Use 'cd' command to change directory. Once this is done, the management system is in place

3. Now,

- a. Provide connection details (username and password) to connect to your database(MYSQL Workbench localhost credentials)

```
(base) anushkatak@Anushkas-MacBook-Pro ~ % cd Desktop/CS5200/Project/application_code
(base) anushkatak@Anushkas-MacBook-Pro application_code % python recipe_management.py
Enter username for connecting to Database
root
Enter password for connecting to Database
█
```

- b. Once connection is established, the program is well built to instruct you at each point. The next step is whether to sign up as a new user to this system, or log in as an existing user. Similar to any website where you need to be a member first to access its contents, the management systems provided you the option.
- c. If you have an account, you can login using the credentials and the program after validation brings you to the home page where you can select from multiple operations. Or, you can create your account by feeding in relevant information.
- d. While answering any question (for e.g. Enter full name), press enter when you are done entering your response. Once sign up is done, the program brings you back to the log in page for you to login.
- e. Once log in is successful, you are shown the variety of operations to choose from. If it fails, the program lets you re-attempt.

```

Sign up? Enter YES if you're a new user or NO, if you're a registered user
no
Enter your email_id
takanushka@gmail.com
Enter your password
atak123
*****
*
Welcome!

*****
*
Press
1 to search a recipe
2 to enter a new recipe
3 to delete an existing recipe
4 to give a comment
5 to edit a comment
6 to delete a comment
7 to view comments
8 to view fun facts
9 to delete your account
10 to give feedback to our System
11 to delete your feedback
12 to exit

```

- f. At each operation, the program provides user with information about either successful operation or a failure message providing a reason and next step to rectify the error. After successful operation, you are brought back to the home page, unless you select explicitly to exit the program.
1. Search a recipe – enter name of the recipe you’re looking for, and the program displays all recipes with that name (You can use, White pasta, Pizza, Basil fried rice to view recipes existing in the database)
  2. Enter a new recipe – you can enter your recipe, fill in few recipe related responses. Some examples of questions are recipe name, recipe, ingredients, cook time, servings. If all valid selections are made, the recipe is entered in the database for all users to view.
  3. Delete a recipe – you enter name of recipe you want to delete, it displays all recipes associated with that name. You select from that list which recipe you want to delete, if that recipe was contributed by you, the program deletes it and prints success message, else displays ‘You can’t delete someone else’s recipe’ message.
  4. Give comment – You enter which recipe you’d like to comment on. On valid selection, you enter your comment on prompt and the comment is entered successfully for that recipe.
  5. Edit comment - displays all your comments, asks you to select which comment to edit. Now you enter new comment and the database is updated with edited comment. If you haven’t provided any comment so far, it notifies you of the same

6. Delete comment - displays all your comments, asks you to select which comment to delete. If your selection is valid, the program deletes your comment. If you haven't provided any comment so far, it notifies you of the same
7. Views comment – displays all your comments on different recipes.
8. View fun facts – displays fun facts about the recipes database.
9. Delete your account – since you're logged in, no verification needed at this point. Deletes your account from database. Now your credentials no longer work to log in.
10. Give feedback – takes feedback from you and enters in the system for admin's view.
11. Delete Feedback -- displays all your feedbacks to the system, asks you to select which feedback to delete. It removes the feedback from the database if it is a valid selection. If you haven't provided any feedback so far, it notifies you of the same.
12. exits the program.

## **Lessons Learnt:**

1. Establishing a connection and enabling operations with front end and back end.
2. Thorough and excessive error handling to make users experience better.
3. Designing the whole system end to end, understanding users requirements, building a coding module to support them and linking it with front end helped me understand the process in terms of design, management and functionality.
4. I started with Website designing in HTML, CSS, JS but due to time constraints, could not finish deploying it. I'd like to build on that and learn Flask and finally enhance the project deployment by linking all these components together in the future.
5. Technical coding in Python and SQL, using programming objects in Database and Python to make the project modular greatly improved my technical skills.

## **Future Work:**

### Planned Uses of database

1. Easy access - Online searching of various global recipes
2. Sharing one's recipes globally – Food blogging
3. Potentially, including top class recipes by Chefs and easy quick recipes by a commoner.
4. Can be used to analyze most searched recipes and used to understand demand (by Restaurateurs)
5. Exposure to various cuisines, food items.

### Potential Areas for added functionality

## Error Handling

At each input, this project enforces validation by several methods (try catch mechanism and providing User-friendly error messages) based on functionality. Instead, SQLAlchemy could be used to facilitate variable handling and making input validation much smoother. It is a library that facilitates the communication between Python programs and databases. It is used as an Object Relational Mapper (ORM) tool that translates Python classes to tables on relational databases and automatically converts function calls to SQL statements.

## Primary Keys

Selection of primary keys was chosen as auto increment. However, it results in wastage of keys once some entries are deleted from the system. E.g., if user with user\_id =1 deletes his account, the user\_id=1 corresponds to no one in the database now.

## Added Functionality

1. Making use of the food preference of users (e.g. Vegetarian, vegan, pescatarian) to filter the recipes for a specific user.
2. Website – Deployment of the Management system can be extended to a Flask application. And further to a website, which uses post-get methods to take in input from the users and performs operations on the database in the backend using SQL.

Attaching the screenshots of the website, I built but due to time constraints, could not deploy it on the website.

