

2.2 Design Requirements

From our vision and outlined design problem, we are putting an emphasis on the database interactivity and readability of our program, focusing on its core functions and how users will interact with them.

2.2.1 Functions

These core functions are as outlined:

1. Create/store a recipe:
 - a. Ability to create or upload a recipe in-app.
 - b. Ability to save the recipe for later use.
2. Viewing a recipe:
 - a. Ability to retrieve a saved recipe when needed.
 - b. Ability to retrieve said recipe on different devices, such as a desktop computer, phone, or tablet.
 - c. Ability to view recipes while offline.
3. Organizing recipes:
 - a. Ability to apply custom tags/filters to recipes.
 - b. Ability to sort by custom tags.
 - c. Ability to search from saved recipes.
 - d. Ability to favourite recipes.
 - e. Ability to remove recipes that are no longer needed.
4. Sharing recipes:
 - a. Ability to back-up a collection of recipes.
 - b. Ability to share a collection of recipes with another user.

2.2.2 Objectives

To accomplish our goals with this application, we have the following objectives in place:

1. User-friendly:
 - the system should offer a clean and intuitive interface.
2. Efficient:
 - the system should respond quickly and minimize unnecessary steps.
3. Reliable:
 - the system should handle data accurately and predictably.
4. Maintainable:
 - the system should follow clean design principles with modular components to support future development.
5. Secure:
 - the system should validate user input to prevent invalid or malicious entries and ensure data integrity.

2.2.3 Constraints

Offline capability:

- An online database is not necessary to store and retrieve data, but rather a local storage system that mimics the concepts for retrieval and management in large collections.
- Sharing recipes and backing up a collection of recipes must be done in a way that can be done through file sharing, such as use of a JSON file to store and read information between independent instances of the application. This must include an import/export feature to handle these files for users so they may accomplish this seamlessly.

Security:

- No information needs to be passed online, and removes the need for account verifications.
- File and input verification are important to the integrity and function of the application in all aspects, so we must ensure all inputs to the application are appropriate and malicious behaviour is avoided. This should be achieved by preventing the use of restricted characters and file types.

File types and inputs:

- Files may not necessarily be text-based, and may also include images.
- Tags and filters can and should be able to be used based on text criteria, but must also be applied should the media a user intends to use includes images.

Tags and filters:

- They may not be based on text-skimming technology due to the variance in accepted file types.
- User may edit or change them independently.
- Favoriting recipes as a unique tag and sort option.

Removing recipes and data that are no longer wanted:

- Properly disposing of related data to remove unnecessary bloat and preserve privacy.

Economically:

- We do not wish to use any external services or paid APIs to accomplish the functions of our application.
- It is made to be free and open-source, to the benefit of our users.

Ethically:

- No user account or personal data should be collected, as part of the mindfulness of privacy and security, and all user inputs or outputs shall be handled equally without bias.