

**NELSON MANDELA UNIVERSITY**

**School of Information Technology**

**IT PROJECT 3 (ITPV302)**

**Bachelor of Information Technology (BIT)**

**27 September 2024**

***System Requirements, Specifications and Technical Design***

**Compiled by**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lecturer** | **Student Name** | **Student Number** | **Date Submitted** |
| **Prof Bertram Haskins** | Nathan Rogers  Zanele Mndaweni  Max Naidoo | 221412581  225097524  225227053 |  |

Contents

[2. Introduction 2](#_Toc177134151)

[2.1 Plan to address objectives 3](#_Toc177134152)

[2.2 Project Scope 5](#_Toc177134153)

[2.1 Information Scope 5](#_Toc177134154)

[2.2 Functional Scope 6](#_Toc177134155)

[2.3 Communication Scope 6](#_Toc177134156)

[2.3 Business Requirements 7](#_Toc177134157)

[2.4 Hardware and Software Requirements 8](#_Toc177134158)

[2.4.1 Software Requirements 8](#_Toc177134159)

[2.4.2 Hardware Requirements 8](#_Toc177134160)

[2.5 Design Constraints 9](#_Toc177134161)

[2.5.1 Security Constraints 9](#_Toc177134162)

[2.5.2 Interface Constraints 9](#_Toc177134163)

[2.5.3 Performance Constraints 9](#_Toc177134164)

[2.6 High level use case diagram 10](#_Toc177134165)

[2.7 UML class diagram 11](#_Toc177134166)

[2.8 Relational database model diagram 12](#_Toc177134167)

[2.9 User interface design 13](#_Toc177134168)

# 2. Introduction

# 2.1 Plan to address objectives

* Offer personalized meal plans which cater to various dietary preferences

Finding recipes that satisfy different dietary preferences can be quite difficult. To ensure that various diets are accommodated, the app will provide personalized meal plans based on user preferences. Upon sign up, user will be allowed to select their unique dietary restrictions (vegan, keto, etc.). The app will then use this to filter and recommend recipes and meal plans.

* Provide a means to filter and search for recipes based on available ingredients

To save time searching for recipes and to make use of ingredients that would otherwise be wasted, users will be able to search for recipes based on the ingredients they have in their kitchen.

* Offer step-by-step instructions with integrated timers for efficient cooking

To ensure that users are managing their time effectively while cooking, each recipe will include detailed step-by-step instructions, with integrated timers for different stages of the recipe where necessary. This will guide users through the cooking process and make it less overwhelming for less experienced cooks, especially helpful for dishes that have time-sensitive steps such as baking.

* Enable offline saving of recipes and grocery list

To accommodate users that may use this app to go grocery shopping where there isn’t usually internet or for cases where there isn’t internet connectivity, the app will support offline saving of recipes and grocery lists. The data will then sync automatically with the cloud once connectivity is restored, ensuring every device has the latest information.

* Allow users to adjust recipes (such as changing metric systems and serving sizes)

To ensure users from different regions and with varying cooking metrics and needs are accommodated, the app will provide tools to adjust the recipes to fit the users needs. Users can choose to change the serving size to fit their needs, the app will automatically update the ingredient quantities for them. Users will also be able to switch between metric and imperial measurement systems which makes the app more user-friendly.

* Make planning meals easier by incorporating a meal planner

Planning meals can be a bit of a headache and time-consuming which is why the app will have a built-in meal planner which users can use to plan their meals in advance. Users can select recipes for each day of the week, and the app will generate a meal plan according to that. This feature will ensure users meet their nutritional and dietary goals throughout the week.

* Grocery list generation based on selected recipes

Making grocery lists manually can be a very tedious and time-consuming chore, often leading to users missing ingredients during grocery trips. The app will automatically generate a grocery list with the necessary ingredients based on the user’s chosen recipe. Users can choose to customize the list by adding or removing ingredients they may have already.

* Allow users to add and share their own recipes

Many home cooks have personal recipes that they may want to share or store digitally but current apps don’t have a user-friendly way to upload or store their recipes. The app will allow users to upload and share their own recipes which will foster a community of home cooks. Users will be to add their own recipes by typing them out or by uploading photos of handwritten recipes which will use Google’s Cloud Vison OCR technology to convert it to text. The system will also allow users to add custom tags, ingredients and instructions where they see fit.

* Support multi-platform access so the app is accessible across various devices

To ensure the app is accessible across different devices, it will be developed using Flutter so the app can run natively on both Android and iOS. A web version will also be available so users can seamlessly access it on desktops and tablets. All data will be synced so users can always access their saved data from any device.

# 2.2 Project Scope

## 2.1 Information Scope

* User information

Basic profile information such as usernames, email addresses, users’ dietary preferences, any past meal plan data. Also, user-generated data such as recipes saved, grocery lists generated/made and personalized meal plans will also be stored. This information will be stored locally on the user’s device as well as on the cloud so that data can be synced across all their devices. User login credentials will be encrypted both in transit (HTTPS) and also at rest since they will be stored on Firebase database. Personal information such as the user’s dietary preferences (allergies, diet) and email will also be encrypted to ensure data privacy and protection from threats.

* Recipe information

Recipe details such as the ingredients, step by step instructions, name of the recipe and the associated dietary tags will be stored. Since users will also be able to upload their own recipes, either by typing them or submitting photos of written recipes. The names, photos and the user ratings will also be stored for each recipe. This information will be stored in a cloud-based database, recipes that are accessed regularly will be cached locally on the user’s device so they can access them offline. Recipes that are added by the user will first go through moderation by moderators before they are uploaded to the cloud globally to all users. Recipes that are public to all users and do not contain sensitive information will not be encrypted. Recipes uploaded by the user will be secured so as to prevent data corruption and so that the recipe isn’t edited by unauthorized people.

* Grocery List and Meal Planner information

Grocery lists that a user generated, and meal plans created will be stored so users can reference them when needed. Information such as ingredients, quantities and recipes linked to the grocery list and plans will all be saved. This data will be stored locally on the users’ device as well as on the cloud. Allowing users to then access their grocery lists and meal plans from any device and also view them offline. Basic encryption will be applied to the data syncs with the cloud since it is not highly sensitive, to ensure the integrity of the data.

* Images

Images linked to recipes, this includes user-uploaded photos will be stored. These images will mainly be stored on the cloud (Firestore), with thumbnails. Recently viewed images will be cached locally so user experience is improved a bit.

* Offline Storage and Syncing

Saved recipes, grocery lists and meal plans selected by the user will be stored locally. This information will be stored on the user’s device in a secure format. When internet is regained, the local data will be synced with the cloud data.

In summary the types of information that will be stored in the app will be user-data, recipe information, grocery list and meal planning information. These will be stored securely in the cloud with the use of Firestore. Recipes can be saved locally on the user’s device to allow accessibility offline as well. User credentials will be encrypted and stored securely on the Firebase database. Local data that has been saved offline will be synced with the cloud once the user has gone online. This allows users to edit their meal plans or grocery lists while offline, which can then be synced across different devices. This includes their selected preferences as well as account information to check if they have been registered.

User data to be stored:

* Usernames
* Email address
* Dietary preferences
* Favourites and saved recipes
* User generated content
* Meal plans
* Grocery list items

Recipe data to be stored:

* Titles
* Images
* Ingredients
* Instructions
* Nutrition information
* Tag data
* Ratings

## 2.2 Functional Scope

The following are a list of the features that are planned to be included in the app:

* Filtering recipes based off ingredients selected
* Filtering recipes based on dietary types and other factors
* Searching of recipes
* Meal Planning
* Saving recipes for later use and offline use
* User can login/register or use the application as a guest
* Generate reports for the system admin

Besides the features that we want to include in the app we also hope to include the following:

* Creating Recipes
* Shopping list option
* Information on nutrition (Possible feature)

Recipe reviews

## 2.3 Communication Scope

The application will work over WIFI mainly and have limited functionality in an offline mode.

Limited functionality available in offline mode will be in the form of the shopping lists and saved recipes.

When the user regains an internet connection the recommendations will appear otherwise a message will inform the user that they are currently not connected to the internet.

# 2.3 Business Requirements

For the project we aim to build a recipe app which home-cooks or any beginner will be able to follow. The app will be cross-platform which allows users to share and sync their recipes and preferences whether they make use of a mobile device or access the website from a desktop.

Home-cooks will be able to depend on the app as it will be accessible offline as well. New recipes will be available based off the initial user preferences or dietary type. These personalized suggestions can be synced across multiple devices for the user. The home-cook will also benefit from the option to plan meals, and add ingredients to a grocery list so that they can prepare for future recipes that they may wish to cook.

The admin will have access to the app to enforce maintenance, monitoring of the performance of the app and moderation of the recipes being posted. The most liked and disliked recipes will be logged for the system admin to view and post. They will also be able to manage other user’s level of access, edit and delete their accounts. The system admin will be responsible for

Unregistered users/guests will have minimal amount of access to the app. They will still be able to view some recipes, but they will not have the features of adding to a grocery list, meal planning or the user preference features.

# 2.4 Hardware and Software Requirements

## 2.4.1 Software Requirements

The application will be developed using Visual Studio Code as the IDE

Flutter will be used for the development of the app within VS Code

Firebase will be used to handle the login and registration as well as the security covering them.

For local storage SQLite will be used

The mobile application requires Android 5 (API level 21) or higher

The web app will be able to run on any browser, but optimized for Microsoft Edge.

## 2.4.2 Hardware Requirements

# 2.5 Design Constraints

## 2.5.1 Security Constraints

Login/Registration security will be handled via Firebase.

{we need to add more here- maybe mentioning the password hashing etc}

When a user adds a recipe to the app/website all text must be verified as text and not code that might be used to create a vulnerability.

## 2.5.2 Interface Constraints

The interface of the app needs to be easy to use and not feature many navigation options that cause a user to become lost. But the app must not be too simple so that it sacrifices features.

Users should be given the opportunity to be able select a diet preference and have a section that recommends recipes that conform to the preferences and an area that allows users to explore new recipes.

The interface should be tailored to various sizes of devices such as tablet or phone and remain consistent.

## 2.5.3 Performance Constraints

The performance of the app may differ across platforms. Since the app is cross-platform, it will be harder to maintain a consistent performance across multiple devices. The app should require a low CPU and memory usage so that it can function across these devices. The responsiveness of the interface and UX design will need to be consistent so that the user’s do not have different experiences in terms of the performance of the system.

## 2.5.4 Data Storage Constraints

Firestore which will be used to store the recipe images on the cloud has a free-tier model which is limited. The free model has a limit of 1GB storage, as well as 1GB of outbound data per month. The database will continue to function normally, but additionally charges will be placed when exceeding this limit.

Users that want to upload their own recipes with images will need to be constrained so that they do not take up the amount of cloud storage space. The maximum file sizes for the image that a user wants to upload should also be limited. As the user saves the recipes too it should be considered if they have enough local storage space to save the recipe offline.

# 2.6 High level use case diagram

# 2.7 UML class diagram

# 2.8 Relational database model diagram

# 2.9 User interface design