МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 1-98 01 03 «Программное обеспечение информационной безопасности мобильных систем»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Программирование мобильных систем»

Тема Мобильное приложение «Библиотека рецептов»

**Исполнитель**

студент 3 курса 8 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_ М. С. Клочко

подпись, дата

**Руководитель**

ассистент \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. И. Уласевич

должность, учен. степень, ученое звание подпись, дата

Допущен(а) к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ Н. И. Уласевич

подпись дата инициалы и фамилия

**Содержание**

[Введение 3](#_Toc166801177)

[1 Анализ требований к программному средству 4](#_Toc166801178)

[1.1 Аналитический обзор аналогов 4](#_Toc166801179)

[1.1.1 Аналог «SuperCook» 5](#_Toc166801180)

[1.1.2 Аналог «Cookmate» 6](#_Toc166801181)

[1.2 Вывод 8](#_Toc166801182)

[2 Проектирование программного средства 9](#_Toc166801183)

[2.1 Проектирование базы данных 9](#_Toc166801184)

[2.2 Описание средств разработки 10](#_Toc166801185)

[2.3 Вывод 11](#_Toc166801186)

[3 Реализация приложения 12](#_Toc166801187)

[3.1 Разработка мобильного приложения 12](#_Toc166801188)

[3.2 Вывод 12](#_Toc166801189)

[4 Тестирование приложения 14](#_Toc166801190)

[4.1 Вывод 15](#_Toc166801191)

[5 Руководство по использованию 16](#_Toc166801192)

[5.1 Руководство гостя 16](#_Toc166801193)

[5.2 Руководство пользователя 17](#_Toc166801194)

[Заключение 24](#_Toc166801195)

[Список использованных источников 25](#_Toc166801196)

[Приложение А 26](#_Toc166801197)

# **Введение**

Актуальность темы разработки мобильного приложения "Библиотека Рецептов" нельзя переоценить в контексте современной кулинарной культуры и повседневной жизни. С развитием информационных технологий и широким распространением мобильных устройств интерес к кулинарии и приготовлению разнообразных блюд становится всё более активным и востребованным. Мобильные приложения в сфере кулинарии приобретают огромную популярность, предоставляя пользователям удобный и доступный способ нахождения, сохранения и использования рецептов.

Целью данного курсового проекта является разработка мобильного приложения "Библиотека Рецептов", которое предоставит пользователям возможность хранить, управлять и находить рецепты блюд, а также получать дополнительную информацию о них. Суть приложения заключается в создании удобного инструмента для всех, кто интересуется кулинарией, независимо от уровня опыта и кулинарных предпочтений.

Приложение предоставит пользователю широкий функционал, включающий в себя вход и регистрацию новых пользователей, управление рецептами (включая добавление, удаление и изменение), поиск рецептов по различным фильтрам, возможность добавления рецепта в избранное для быстрого доступа, а также подбор рецептов на основе имеющихся у пользователя продуктов. Дополнительно, предусмотрено добавление возможности пользовательского внесения рецептов с указанием их характеристик, включая стоимость и подробные сведения о блюде, такие как калорийность, вес и другие.

Такое приложение позволит пользователям экспериментировать с кулинарными идеями, обогатить свой кулинарный опыт, удовлетворить интересы в разнообразии кулинарных рецептов, а также сэкономить время на поиск нужной информации о блюдах и их приготовлении. Разработка такого приложения важна для обеспечения комфортной и удобной среды для любителей кулинарии и активных пользователей мобильных технологий.

# **Анализ требований к программному средству**

1. Вход и регистрация новых пользователей:

* Приложение должно предоставлять форму для ввода логина и пароля для входа.
* Должна быть возможность регистрации новых пользователей с указанием логина, пароля и электронной почты.
* Предусмотреть механизм проверки уникальности логина и электронной почты.
* Обеспечить безопасное хранение учетных данных пользователей.

1. Управление рецептами:

* Добавление нового рецепта:
* Форма для заполнения информации о блюде: название, описание, ингредиенты, шаги приготовления и т.д.
* Возможность добавления изображения блюда.

1. Удаление рецепта:

* Возможность удаления рецепта из библиотеки пользователя.

1. Изменение рецепта:

* Возможность редактирования информации о рецепте.

1. Просмотр детальной информации о рецепте: калорийность, вес, время приготовления и т.д.
2. Поиск рецептов по фильтрам:

* Фильтрация рецептов по категориям (завтрак, обед, ужин, десерты и т.д.).
* Поиск по названию блюда или его ингредиентам.
* Возможность сортировки результатов по различным критериям.
* Добавление рецепта в избранное:
* Возможность добавления понравившегося рецепта в список избранных для быстрого доступа.

1. Подбор рецепта по имеющимся продуктам:

* Пользователь вводит список доступных у него ингредиентов.
* Приложение предлагает рецепты, которые можно приготовить на основе этих продуктов.

1. Пользовательский внесение рецептов:

* Возможность пользователям добавлять свои собственные рецепты с указанием всех необходимых параметров.
* Обязательные поля для заполнения при добавлении рецепта.
* Валидация данных перед добавлением в базу данных.

1. Дополнительная информация о блюде:

* Предоставление подробной информации о блюде, такой как калорийность, вес, время приготовления, количество порций и т.д.
* Возможность оценивать и комментировать рецепты.

## **Аналитический обзор аналогов**

Критерии анализа:

* наличие функционала добавления и редактирования рецептов;
* наличие функционала добавление рецепта в избранное;
* наличие функционала удобная фильтрация;
* наличие функционала поиск рецептов по ингредиентам;

Цели пользователя: удобство интерфейса приложения для поиска, просмотра и добавления рецептов, возможность управления рецептами: добавление, удаление, редактирование.

Бизнес-цели: возможные модели монетизации приложения, таких как платные подписки, реклама, платные функции и т.д.

### **Аналог «SuperCook»**

Первый аналог «SuperCook» [1] преимуществом данного мобильного приложения является фильтрация рецептов в данном приложении достаточно много фильтров для поиска нужных блюд , так же есть голосовой ввод что безусловно удобно в использовании приложением (Рисунок 1.1).

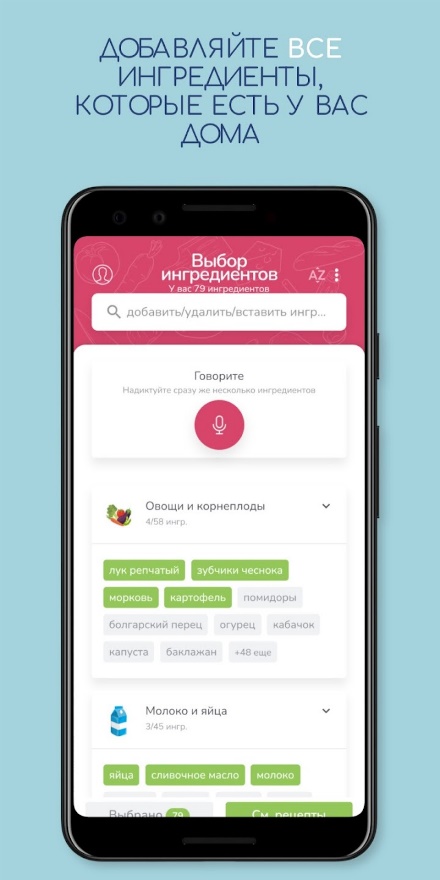


Рисунок 1.1 – Голосовой ввод

Один из плюсов приложения - возможность поиска рецептов по уже имеющимся ингредиентам, что экономит время пользователей и облегчает процесс выбора блюд. Пользователи могут вводить список доступных ингредиентов, и приложение предложит им соответствующие рецепты, используя эти ингредиенты. Такая функция делает приложение более удобным и привлекательным для любителей готовить дома (Рисунок 1.2).



Рисунок 1.2 – Поиск по игридиентам

Из недостатков данного приложения могу выделить его простоту в дизайне, а также у некоторых рецептов в ингредиентах некоторые продукты, которые я не вносил, отмечены как имеющиеся. И почему-то часто это какие-то специфические продукты, которые не всегда есть дома. Неудобно, что нельзя фильтровать сразу по нескольким кухням или типам блюд.

### **Аналог «Cookmate»**

Вторым аналогом является «Cookmate» [2] данное мобильное приложение выделяется тем что имеет в себе календарь в котором можно расписывать рационы а так же смотреть в какой день был добавлен тот или иной рецепт (Рисунок 1.3).

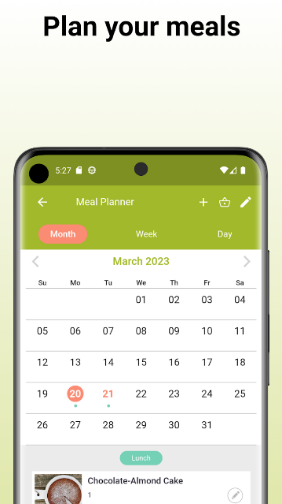


Рисунок 1.3 – Календарь рациона

Ещё одним значимым плюсом этого приложения является возможность сохранения рецептов не только из самого приложения, но и с удалённых ресурсов. Это расширяет доступ пользователей к разнообразным кулинарным идеям и помогает сохранить интересные рецепты для дальнейшего использования. Пользователи могут легко импортировать рецепты из внешних источников, таких как веб-сайты или блоги о кулинарии, и хранить их в едином месте внутри приложения для удобства (Рисунок 1.4).

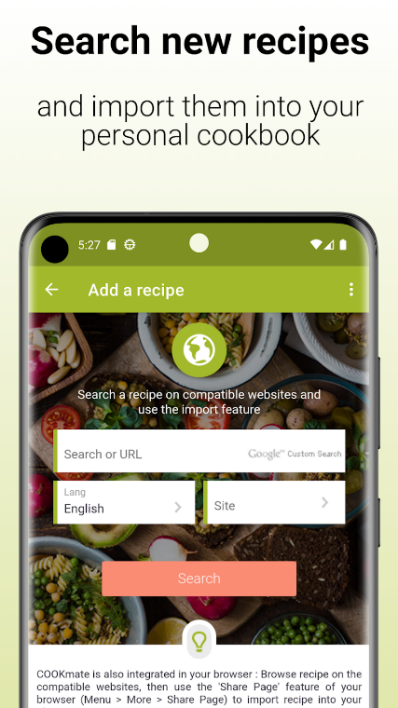


Рисунок 1.4 – Добавление рецепта с удалённого ресурса

Ещё из наиболее приятных функций в приложении есть сканер бумажной версии рецепта что очень помогло бы при переносе старых рецептов, записанных в книги (Рисунок 1.5).



Рисунок 1.5 – Скан рецептов

Из минусов данного приложения хотелось бы отметить, как и в предыдущем аналоге простой дизайн, а также неудобные кнопки, большого размера. Нет подсказок продуктов, если нужна найти какой-то определённый ингредиент, то придётся писать слова целиком.

## **Вывод**

При анализе аналогов мобильных приложений для хранения и поиска рецептов, важно учитывать ряд ключевых критериев, таких как наличие функционала добавления и редактирования рецептов, возможность добавления рецепта в избранное, удобная фильтрация и поиск рецептов по ингредиентам.

Пользователи стремятся к удобству в использовании приложения для поиска, просмотра и добавления рецептов, а также желают иметь возможность управления рецептами, включая добавление, удаление и редактирование.

При анализе приложения "SuperCook" выделяется его преимущество в обширном функционале фильтрации рецептов, что делает процесс поиска нужных блюд более удобным. Дополнительно, наличие голосового ввода упрощает взаимодействие с приложением. Однако, следует отметить недостатки, такие как простой дизайн и некоторые ошибки в отображении ингредиентов, что может затруднить пользовательский опыт.

Приложение "Cookmate" предлагает уникальные функции, такие как календарь рациона, возможность сохранения рецептов с удалённых ресурсов и сканер бумажной версии рецепта. Эти функции придают приложению конкурентное преимущество, однако, также существуют недостатки, такие как простой дизайн и неудобные кнопки управления, которые могут ухудшить пользовательский опыт.

В целом, оба аналога имеют свои сильные и слабые стороны, и успешность приложения "Библиотека Рецептов" будет зависеть от того, насколько эффективно оно сможет удовлетворить требования пользователей и реализовать бизнес-цели.

# **Проектирование программного средства**

## **Проектирование базы данных**

В качестве базы данных был выбран SQLite с помощью DbVizualizer была построена схема самой базы данных.

В соответствии с функциональными требованиями курсового проекта была определена схема базы данных (Рисунок 2.1)

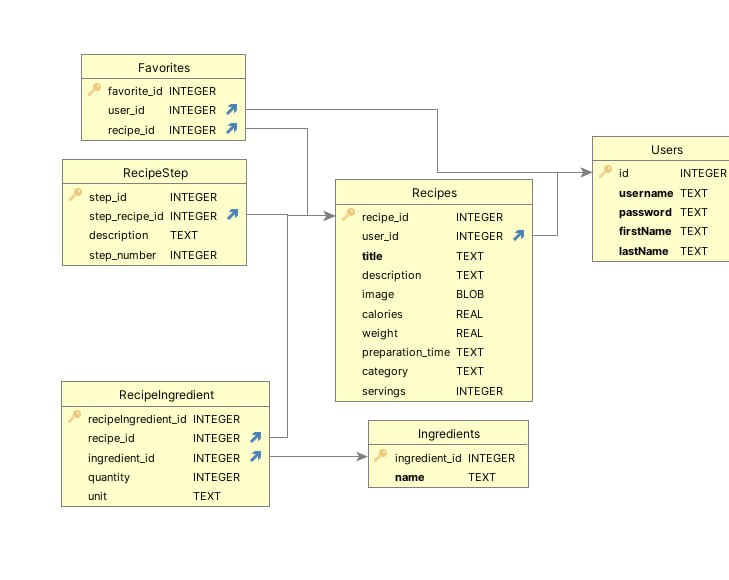


Рисунок 2.1 – Схема базы данных

Описание структуры базы данных:

Таблица "Users":

* user\_id (INTEGER, PRIMARY KEY) - уникальный идентификатор пользователя
* username (TEXT, NOT NULL) - логин пользователя;
* password (TEXT, NOT NULL) - пароль пользователя;
* email (TEXT, NOT NULL) - электронная почта пользователя;
* firstname (TEXT, NOY NULL) – имя пользователя;
* secondname (TEXT, NOY NULL) – фамилия пользователя;

Таблица "Recipes":

* recipe\_id (INTEGER, PRIMARY KEY) - уникальный идентификатор рецепта;
* user\_id (INTEGER, FOREIGN KEY) - идентификатор пользователя, создавшего рецепт;
* title (TEXT, NOT NULL) - название рецепта;
* description (TEXT) - описание рецепта;
* image (BLOB) - изображение блюда (хранится в виде BLOB);
* calories (REAL) - калорийность блюда;
* weight (REAL) - вес блюда;
* preparation\_time (INTEGER) - время приготовления;
* category (TEXT) – категория блюда;
* servings (INTEGER) - количество порций;

Таблица "Ingredients":

* ingredient\_id (INTEGER, PRIMARY KEY) - уникальный идентификатор ингредиента;
* name (TEXT, NOT NULL) - название ингредиента;

Таблица "Favorites":

* favorite\_id (INTEGER, PRIMARY KEY) - уникальный идентификатор записи в избранном;
* user\_id (INTEGER, FOREIGN KEY) - идентификатор пользователя, добавившего рецепт в избранное;
* recipe\_id (INTEGER, FOREIGN KEY) - идентификатор рецепта, добавленного в избранное;

Таблица "RecipeIngedient":

* recipeIngredient\_id (INTEGER, PRIMARY KEY) - уникальный идентификатор ингредиента;
* recipe\_id (INTEGER, FORIEGN KEY) - уникальный идентификатор рецепта;
* ingredient\_id (INTEGER, FORIEGN KEY) - уникальный идентификатор ингредиента;
* quantity (TEXT) - количество ингредиента;
* unit (TEXT) - единица измерения;

Таблица "RecipeStep":

* step\_id (INTEGER, PRIMARY KEY) – уникальный идентификатор шага;
* step\_recepi\_id (INTEGER, FORIEGN KEY) - уникальный идентификатор рецепта;
* description (TEXT) - описание шагов;
* step\_number (INTEGER) – номер шага;

## **Описание средств разработки**

Flutter – это открытая платформа разработки мобильных приложений, созданная компанией Google. Она позволяет разработчикам использовать язык программирования Dart и фреймворк Flutter для создания кроссплатформенных мобильных приложений, поддерживаемых как на устройствах Android, так и на iOS. Одной из ключевых особенностей Flutter является возможность написания одного и того же кода для различных платформ, что способствует ускорению процесса разработки и сокращению затрат. Flutter позволяет разработчикам создавать высокопроизводительные и интуитивно понятные мобильные приложения с помощью привычных инструментов и практик разработки.

SQLite – это легковесная встраиваемая система управления реляционными базами данных, широко используемая в мобильной разработке на Flutter. Она обеспечивает эффективный способ хранения и управления данными приложения, включая создание, чтение, обновление и удаление записей. SQLite идеально подходит для приложений с небольшими и средними объемами данных, требующих локального хранения.

## **Вывод**

В процессе разработки приложения, проектирование базы данных (БД) и мобильного приложения играют решающую роль. Проектирование базы данных определяет структуру данных и их взаимосвязи, в то время как проектирование мобильного приложения определяет пользовательский интерфейс, функциональность и логику приложения. После этапа проектирования, основанного на требованиях проекта, можно выделить роли и обязанности для разработчиков и администраторов. Вот несколько ключевых моментов для каждого раздела:

* Проектирование базы данных (БД):
  + Необходимо определить структуру данных, включая таблицы, поля и их типы данных.
  + Разработать связи между таблицами, определив первичные и внешние ключи для обеспечения целостности данных.
  + Учесть требования к производительности и масштабируемости при проектировании БД.
  + Обеспечить безопасность данных, включая ролевую модель доступа и шифрование при необходимости.
* Проектирование мобильного приложения:
  + Разработать пользовательский интерфейс (UI) с учетом лучших практик дизайна пользовательского опыта (UX).
  + Определить функциональность приложения, включая основные и дополнительные возможности.
  + Разработать архитектуру приложения, учитывая модель представления данных (MVC, MVVM и т.д.).
  + Обеспечить совместимость с различными устройствами и разрешениями экрана, а также поддержку различных операционных систем.

В целом, эффективное проектирование базы данных и мобильного приложения является ключевым этапом разработки, который определяет успех проекта. Четкое определение ролей и обязанностей помогает обеспечить эффективное выполнение каждого этапа разработки.

# **Реализация приложения**

Реализация программного средства — это процесс преобразования детального плана или концепции программного продукта, разработанного в процессе проектирования, в функциональный продукт.

## **Разработка мобильного приложения**

Мобильное приложение содержит в себе несколько папок. Каждая из перечисленных папок в мобильном приложении выполняет определенную функцию:

* Utils: представляет классы-помощники, которые используются во всем приложении и содержат некоторую логику.
* Services: хранит сервисы для аутентификации.
* Welcome: отвечает за splash-экран с приветствием.
* Singup: отвечает за регистрацию пользователя в приложении.
* Login: отвечает за вход пользователя в приложение.
* Screens: отвечает за классы для взаимодействия с пользовательским интерфейсом.

Реализация асинхронной функции для создания таблицы рецептов, показано на листинге 3.1.

Future \_onCreate(Database db, int version) async {

await db.execute('''

CREATE TABLE $recipesTable (

$columnRecipeId INTEGER PRIMARY KEY AUTOINCREMENT,

$columnUserId INTEGER,

$columnTitle TEXT NOT NULL,

$columnDescription TEXT,

$columnImage BLOB,

$columnCalories REAL,

$columnWeight REAL,

$columnPreparationTime TEXT,

$columnCategory TEXT,

$columnServings INTEGER,

FOREIGN KEY ($columnUserId) REFERENCES $table($columnId) ON DELETE CASCADE

)

''');

Листинг 3.1 ­ Создание таблицы рецептов

Пример добавления новых рецептов показан в приложении А.

## **Вывод**

В данном разделе описывается структура мобильного приложения, которая включает в себя различные папки, каждая из которых имеет свою уникальную роль в рамках приложения. Это подчеркивает важность модульности и организации кода в разработке программных продуктов, что является ключевым аспектом в создании масштабируемых и поддерживаемых систем.

1. Модульность: Структурирование приложения на основе различных папок, каждая из которых отвечает за определенный аспект работы приложения (например, аутентификация, пользовательский интерфейс), позволяет упростить процесс разработки и обслуживания. Это обеспечивает легкость внесения изменений в отдельные части приложения без влияния на другие его компоненты.

2. Упрощение поддержки: Разделение кода на модули помогает в более эффективной поддержке приложения. Если требуется исправить ошибку или добавить новую функциональность, разработчикам не нужно искать изменения в одном большом файле; вместо этого они могут сосредоточиться на конкретной области приложения.

3. Гибкость: Такая структура позволяет легко добавлять новые функции или модули в приложение, не затрагивая существующий код. Это особенно важно для мобильных приложений, где требования к функционалу могут меняться со временем.

4. Обучение и масштабирование команды: Для новых членов команды разработки такая структура кода облегчает ознакомление с проектом, так как каждый модуль или папка может быть изучен независимо от остальных. Это также способствует более эффективному распределению задач между членами команды.

5. Адаптивность: Подход, основанный на модульности, делает приложение более адаптивным к изменениям требований и технологий. Например, если потребуется интеграция с новыми API или использование новых библиотек, эти изменения можно внедрить в соответствующие модули без значительных переработок всего приложения.

В целом, организация кода в виде модулей и папок, каждая из которых отвечает за определенный аспект функционирования приложения, является лучшей практикой в разработке программных продуктов. Это обеспечивает не только чистоту и понятность кода, но и повышает его гибкость и масштабируемость.

# **Тестирование приложения**

Тестирование мобильных приложений важно и необходимо для обеспечения их качества и производительности. Оно помогает уменьшить риск сбоев приложения, гарантирует отсутствие ошибок и улучшает восприятие пользователей, что в свою очередь способствует увеличению количества загрузок и успешному запуску приложения на рынке.

Для начала протестируем страницу авторизации. Если пользователь не зарегистрирован, ввел неверные данные или не ввел вообще ничего, то его не запустят в приложение 4.1.

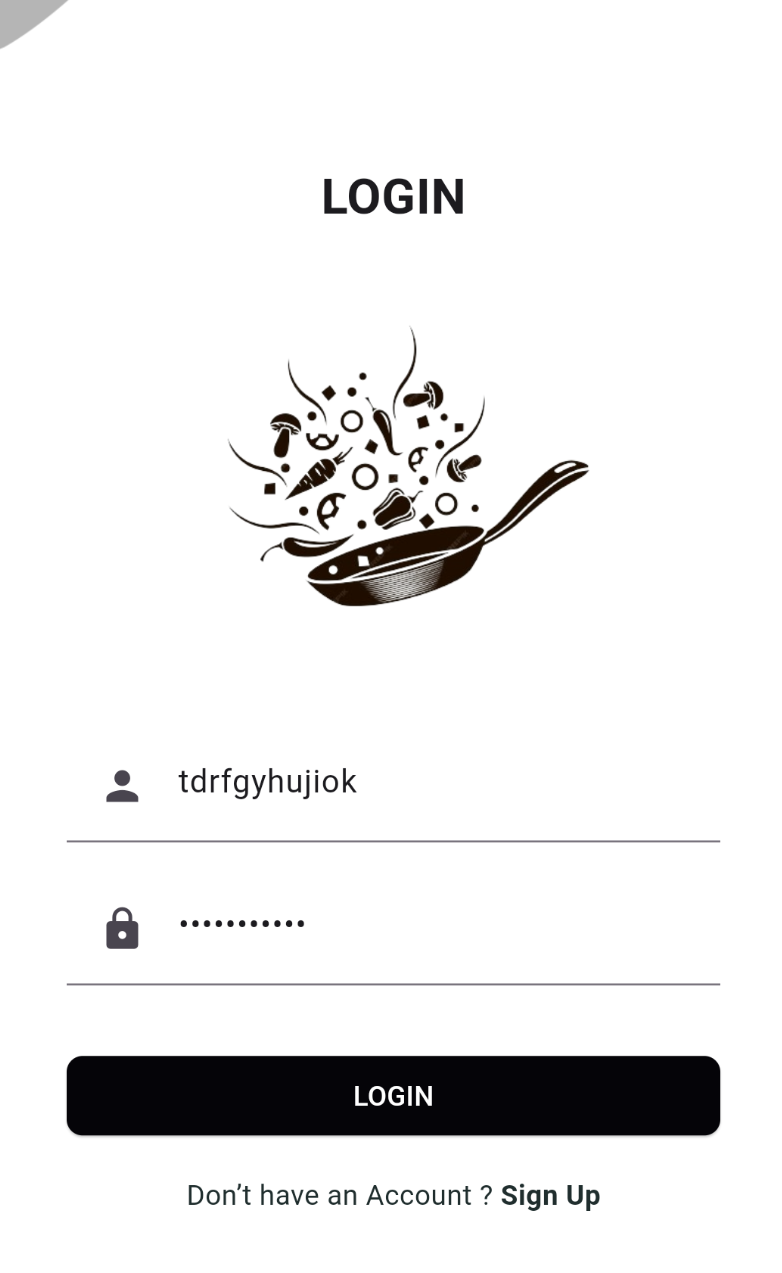


Рисунок 4.1 ­ Тестирование авторизации

Добавление нового рецепта, пользователь может подробно описать свой рецепт: ингредиенты, подачу, шаги приготовления, показано на рисунке 4.2.

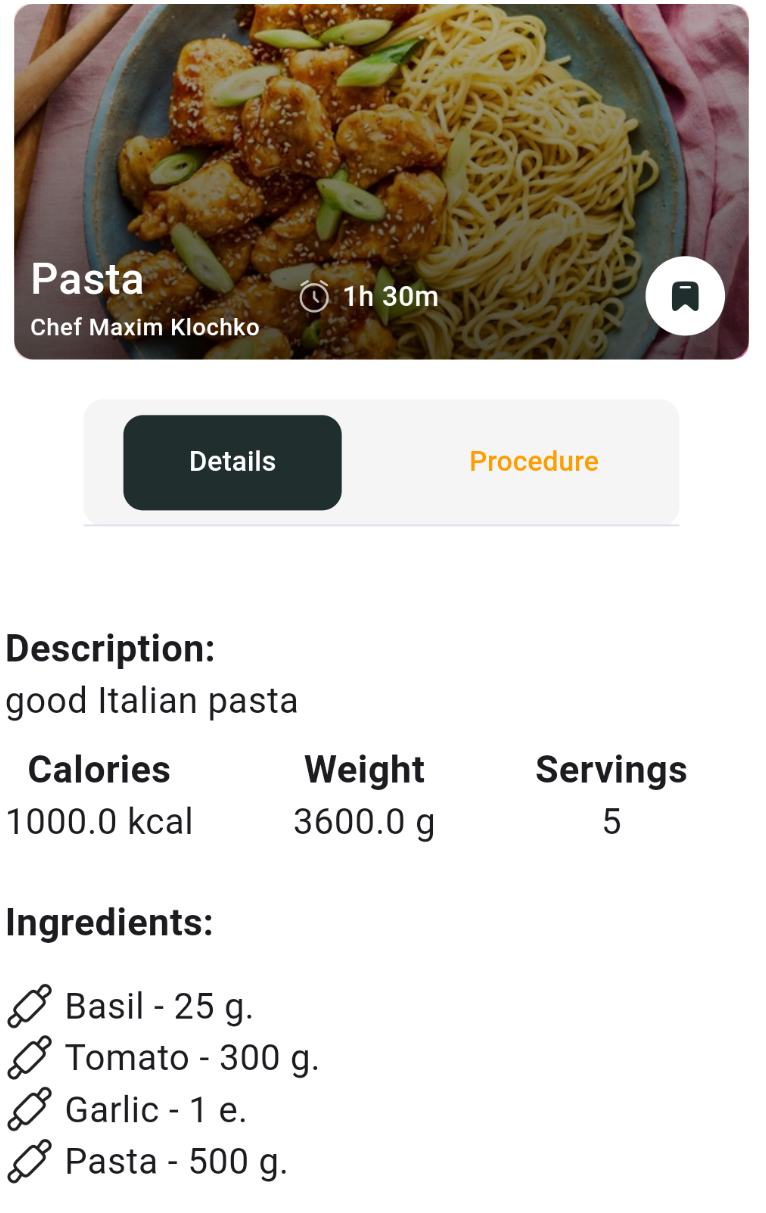


Рисунок 4.2 – Тестирование добавления рецепта

Так же пользователь может добавлять рецепты себе в закладки, созданные пользователем рецепты сохраняются у него в профиле.

## **Вывод**

Тестирование является критически важным этапом в процессе разработки программных продуктов, включая мобильные приложения. Оно позволяет обнаруживать и исправлять ошибки до того, как продукт будет выпущен, обеспечивая высокое качество и надежность конечного продукта. В контексте мобильных приложений, тестирование может охватывать широкий спектр аспектов, начиная от проверки функциональности и производительности до безопасности и совместимости.

1. Ключевой фактор качества: Тестирование является фундаментом для обеспечения качества продукта. Без тщательного тестирования невозможно гарантировать, что все функции работают корректно и что приложение безопасно для использования пользователями.

2. Предотвращение ошибок: Тестирование помогает выявить и исправить ошибки на ранних стадиях разработки, что значительно снижает стоимость их устранения по сравнению с исправлением ошибок после выпуска продукта.

3. Улучшение пользовательского опыта: Тестирование пользовательского интерфейса и взаимодействия с пользователем позволяет оптимизировать дизайн и функциональность приложения, делая его более удобным и интуитивно понятным для конечных пользователей.

4. Поддержание стандартов безопасности: С тестированием безопасности можно убедиться, что данные пользователей защищены, а конфиденциальная информация не попадет в руки злоумышленников. Это особенно важно для мобильных приложений, которые часто сталкиваются с угрозами безопасности.

5. Совместимость и производительность: Тестирование позволяет проверить, как приложение работает на различных устройствах и версиях операционных систем, а также оценить его производительность. Это гарантирует, что приложение будет работать плавно и стабильно для всех пользователей.

6. Эффективность команды разработки: Тестирование помогает определить, какие части приложения требуют дополнительной работы или доработки, позволяя команде разработки сосредоточиться на наиболее важных задачах и улучшениях.

7. Документация и обучение: Результаты тестирования служат ценным источником информации для документации продукта и обучения персонала. Они помогают понять сильные и слабые стороны продукта, а также выявить потенциальные проблемы, которые могут возникнуть при использовании.

# **Руководство по использованию**

В данном разделе будет описано руководство по использованию для гостя и пользователя.

## **Руководство гостя**

Для гостя реализовано окно входа в приложения, где он либо может зарегистрировать новый аккаунт нажав на «Login» либо войти в уже существующий аккаунт нажав на кнопку «Sign up» (Рисунок 5.1).



Рисунок 5.1 – страница входа в приложение

После нажатия кнопки «Sign up» гость переходит на страницу с регистрацией где он может ввести свои данные для регистрации (Рисунок 5.2).

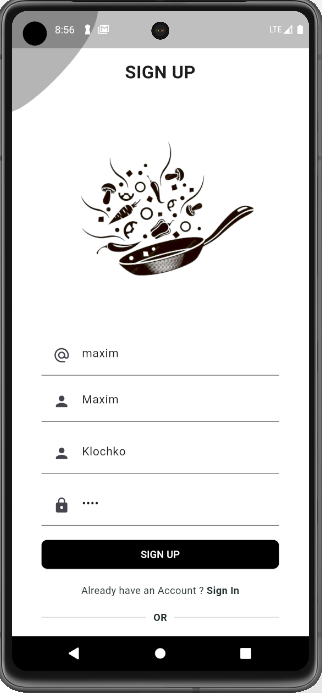


Рисунок 5.2 – страница регистрации

После успешной регистрации гостя перенаправляет на страницу входа где он может ввести свои данные для входа (Рисунок 5.3).

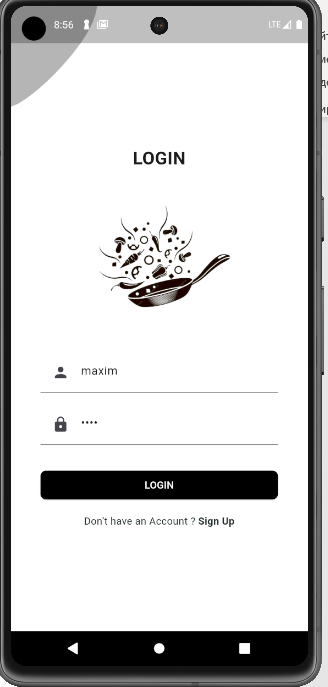


Рисунок 5.3 – страница входа в приложение

## **Руководство пользователя**

После успешного входа пользователь попадает на главную страницу где может увидеть нижнее навигационное меню благодаря которому он может перемещаться по страницам (Рисунок 5.4)

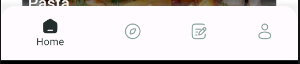


Рисунок 5.4 – навигационное меню

На первой странице пользователь может увидеть рецепты созданные другими пользователя, а также отсортировать их по категориям блюд в зависимости от времени приёма пищи (Рисунок 5.5).

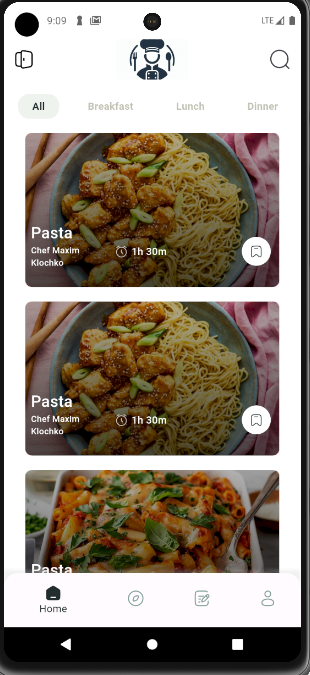


Рисунок 5.5 – главная страница

При нажатии на рецепт пользователю открывается окно с более подробной информацией о рецепте где он с помощью иконки (Рисунок 5.6) может добавить данный рецет к себе в избранное (Рисунок 5.7)



Рисунок 5.6 – иконка добавления рецепта в избранное

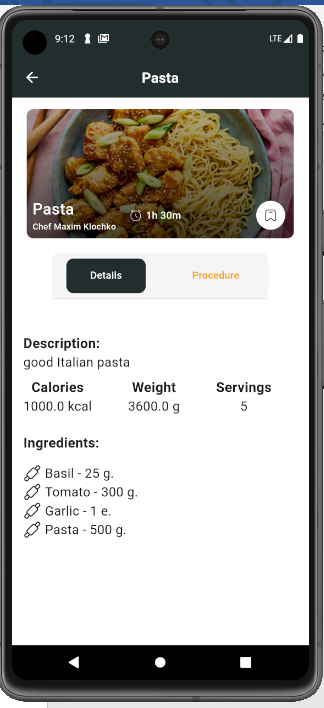


Рисунок 5.7 – подробная информация о рецепте

При нажатии на «Procedure» открывается информация о этапах приготовления рецепта (Рисунок 5.8).

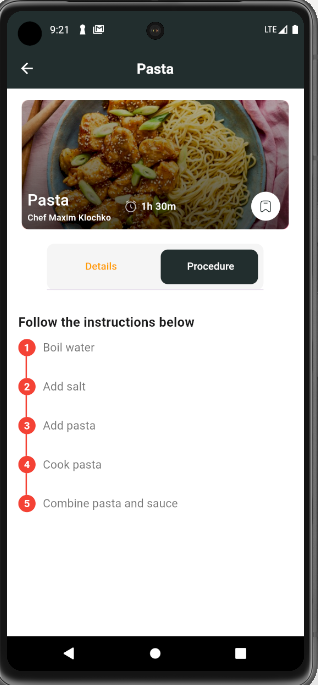


Рисунок 5.8 – описание этапов приготовления рецепта

При нажатии на иконку поиска пользователь может перейти на страницу с поиском где может выбрать определённые ингридиенты по которым он сможет найти нужный рецепт (Рисунок 5.9).

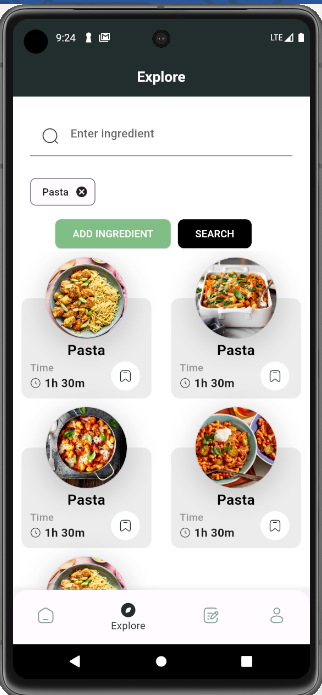


Рисунок 5.9 – поиск по ингридиентам

При нажатии на 3ю иконку пользователь переходит на страницу с добавлением нового рецепта где он может добавить свой собственный рецепт который будет виден другим пользователям (Рисунок 5.10).

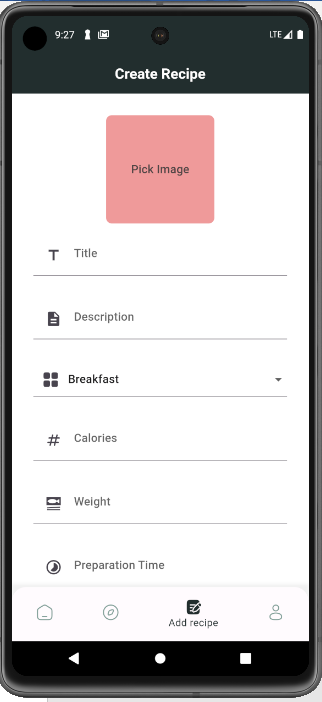


Рисунок 5.10 – добавление нового рецепта

При нажатии на иконку профился пользователь переходит на страницу с профилем где он может просматривать свои рецепты, выйти из аккаунта, просматривать рецепты добавленные в избранное , а так же редактировать профиль (Рисунок 5.11).

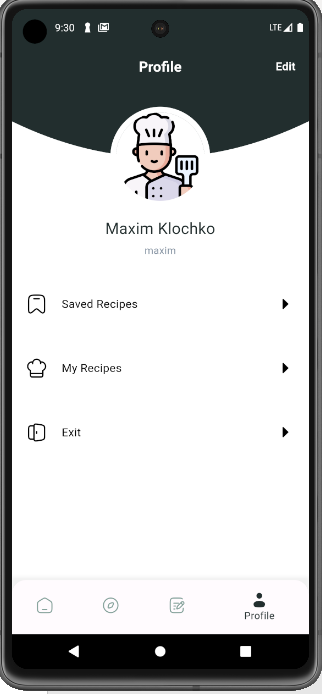


Рисунок 5.11 – профиль пользователя

При нажатии на «Saved Recipes» пользователь переходит на страницу с избранными цептами где он может убрать рецепт из избранного или просматривать более подробную информацю о рецепте (Рисунок 5.12).

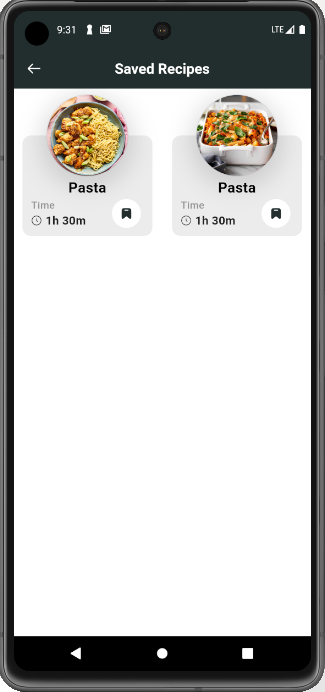


Рисунок 5.12 – страница с избранными рецептами пользователя

При нажатии на «Edit» пользователь переходи на страницу с изменением его личной информации (Рисунок 5.13).

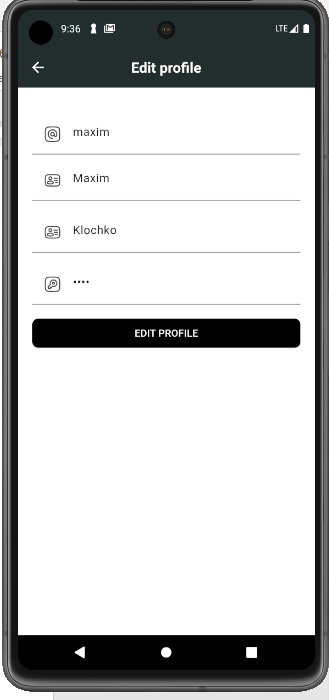


Рисунок 5.13 – страница редактирования пользователя

Если в профиле пользователь нажимает на «My Recipes» он переходит на страницу со списком своих рецептов (Рисунок 5.14).



Рисунок 5.14 – список рецептов добавленных пользователем

При нажатии на рецепт из списка пользователь переходит на страницу с редактированием рецепта где он может либо удалить рецепт либо его отредактировать (Рисунок 5.15).

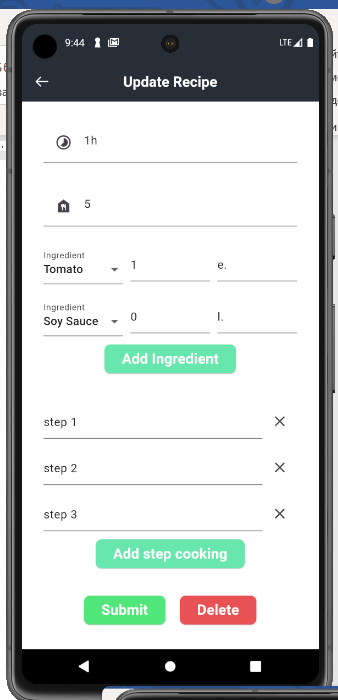


Рисунок 5.15 – редактирование рецепта созданного пользователем

# **Заключение**

В рамках курсового проекта было разработано приложение, в котором был реализован такой функционал как:

* Вход и регистрация новых пользователей:
* Управление рецептами:
* Удаление рецепта:
* Изменение рецепта:
* Добавление нового рецепта:
* Просмотр детальной информации о рецепте
* Подбор рецепта по имеющимся продуктам

В заключении хочется подчеркнуть, что разработка мобильного приложения "Библиотека Рецептов" является значимым шагом в сфере кулинарных технологий. Оно предоставляет уникальную возможность пользователям получать доступ к разнообразным рецептам блюд, управлять ими и находить новые идеи для кулинарных экспериментов. С учетом активного интереса общества к кулинарии и широкого использования мобильных устройств, данное приложение встречает потребности современного пользователя.

Целью проекта было создание интуитивно понятного и удобного в использовании приложения, которое сможет удовлетворить запросы как начинающих кулинаров, так и опытных гурманов. Благодаря функционалу "Библиотеки Рецептов", пользователи могут не только сохранять и управлять рецептами, но и получать дополнительную информацию о блюдах, а также использовать персонализированные рекомендации и подборки.

В итоге, разработка приложения не только способствует расширению кулинарных горизонтов пользователей, но и создает комфортную среду для творчества и экспериментов на кухне. Оно становится незаменимым помощником в повседневных кулинарных занятиях и содействует развитию кулинарной культуры в целом.

# **Список использованных источников**

1. Sqflite [Электронный ресурс]. – Режим доступа: <https://pub.dev/packages/sqflite> – Дата доступа: 05.04.2024
2. Flutter [Электронный ресурс]. – Режим доступа: <https://docs.flutter.dev/tools/devtools/overview> – Дата доступа: 16.04.2024
3. Solutions on Stackoverflow [Электронный ресурс]. – Электронные данные. – Режим доступа: [https://stackoverflow.com.](https://stackoverflow.com./) – Дата доступа: 20.04.2024;
4. Чернышев, С.А. Основы Dart. 2-е издание / С.А. Чернышев. – Москва: Издательство "Наука", 2023. – 350 с.
5. Flutterfor [Электронный ресурс]. – Режим доступа: https://flutterfor.dev/ – Дата доступа: 20.03.2024
6. Dartflutter [Электронный ресурс]. – Режим доступа: https://dartflutter.ru/: 25.03.2024

# **Приложение А**

Future<int> insertRecipe(Recipe recipe, List<RecipeStep> steps, List<RecipeIngredient> recipeIngredients) async {

Database db = await database;

int recipeId = await db.insert(recipesTable, recipe.toMap());

for (RecipeIngredient ingredient in recipeIngredients) {

ingredient.recipeId = recipeId;

await db.insert(recipeIngredientTable, ingredient.toMap());

}

for (RecipeStep step in steps) {

step.recipeId = recipeId;

await db.insert(recipeStepTable, step.toMap(recipeId));

}

return recipeId;

}

void onSubmit(BuildContext context , Recipe recipe, List<RecipeIngredient> ingredients, List<RecipeStep> steps) async {

final DatabaseHelper databaseHelper = DatabaseHelper();

try { await databaseHelper.insertRecipe(recipe, steps, ingredients);

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(

content: Text('Recipe created successfully!'),

backgroundColor: Colors.green,

),);} catch (e) {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(

content: Text('Failed to create recipe: $e'),

backgroundColor: Colors.red,

),

);

}}

ElevatedButton(

onPressed: () async { if (\_formKey.currentState!.validate()) {

SharedPreferences prefs = await SharedPreferences.getInstance();

int? userId = prefs.getInt(\_userIdKey);

final recipe = Recipe( userId: userId!,

title: \_titleController.text,

description: \_descriptionController.text,

image: imageBytes,

calories: double.parse(\_caloriesController.text),

weight: double.parse(\_weightController.text),

preparationTime: \_preparationTimeController.text,

servings: int.parse(\_servingsController.text),

category: selectedCategory,

);

// ignore: use\_build\_context\_synchronously

widget.onSubmit(context ,recipe , ingredients, steps);}