Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж

имени Б.С. Галущака»

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ

«Книга рецептов»

Пояснительная записка к курсовому проекту

ПМ.01 Разработка модулей программного обеспечения  
для компьютерных систем

МДК.01.03 Разработка мобильных приложений

НАТКиГ.200300.010.000ПЗ

Разработал:

студент группы ПР-21.101

Борисов Ф. Е.

**Содержание**

[Введение 3](#_Toc165874993)

[1 Исследовательский раздел 4](#_Toc165874994)

[1.1 Описание предметной области 4](#_Toc165874995)

[1.2 Образ клиента 4](#_Toc165874996)

[1.3 Сценарии использования 5](#_Toc165874997)

[1.4 Сбор и анализ прототипов 6](#_Toc165874998)

[1.5 Основные негативные отзывы от пользователей приложений 7](#_Toc165874999)

[2 Проектирование приложения 9](#_Toc165875000)

[2.1UI/UX дизайн приложения 9](#_Toc165875001)

[2.2 Выбор технологии, языка и среды программирования 14](#_Toc165875002)

[3 Разработка мобильного приложения 15](#_Toc165875003)

[3.1 Разработка базы данных 15](#_Toc165875004)

[3.2 Разработка мультимедийного контента 17](#_Toc165875005)

[3.3 Описание используемых плагинов 18](#_Toc165875006)

[3.4 Описание разработанных процедур и функций 18](#_Toc165875007)

[4 Тестирование 22](#_Toc165875008)

[1.1 Протокол тестирования дизайна приложения 22](#_Toc165875009)

[4.2 Протокол тестирования функционала приложения 23](#_Toc165875010)

[Заключение 25](#_Toc165875011)

[Библиография 26](#_Toc165875012)

[Приложение А (Обязательное) Тех. Задание 27](#_Toc165875013)

[Приложение Б 33](#_Toc165875014)

Введение

В современном мире, где технологии проникают во все сферы нашей жизни, мобильные приложения становятся неотъемлемой частью повседневности. Они упрощают наши задачи, расширяют доступ к информации и помогают нам быть более продуктивными и организованными. В контексте кулинарного искусства мобильные приложения также играют значимую роль, предоставляя пользователям возможность легко находить, сохранять и практиковать рецепты.

В этом контексте представляется важным разработка мобильного приложения "Книга рецептов", которое сможет удовлетворить потребности как опытных кулинаров, так и начинающих любителей готовить.

Целью курсового проекта является создание мобильной приложения для записи и хранения рецептов.

Задачами курсового проекта в связи с указанной целью являются:

* изучение предметной области;
* рассмотрение рецептов с точки зрения пользователя;
* рассмотрение похожих проектов с целью дополнения и улучшение реализуемого продукта;
* выявление предпочтений готовящих людей с целью создать удобный для пользователей продукт;
* разработка и реализация дизайна приложения;
* написание кода приложения;
* тестирование полученного продукта.

Реализуемый продукт – мобильное приложение «Книга рецептов».

# ****Исследовательский раздел****

## ****Описание предметной области****

Приложение "Книга рецептов" охватывает всю сферу кулинарии и приготовления пищи. Пользователи приложения имеют доступ к локальной базе рецептов, которая может включает в себя блюда различных кухонь мира, начиная от классических рецептов до экзотических и авторских блюд. Каждый рецепт представлен с подробными инструкциями, состоящими из этапов приготовления, количества необходимых ингредиентов, а также снабжен фотографиями для наглядности.

Пользователи могут легко находить интересующие их рецепты с помощью удобной системы поиска, основанной на названии блюда или по словам указанных в рецепте, времени приготовления и наличии конкретных ингредиентов. Кроме того, приложение предлагает функцию сохранения любимых рецептов.

Таким образом, предметная область приложения "Книга рецептов" охватывает все аспекты кулинарного искусства, начиная от поиска и выбора рецептов для их приготовления, до записи о способах приготовления новых блюд.

## Образ клиента

При разработке мобильного приложения "Книга рецептов" важно иметь четкое представление о целевой аудитории и их потребностях. Образ клиента включает в себя характеристики и особенности потенциальных пользователей приложения. Целевая аудитория "Книги рецептов" включает в себя:

* любителей готовить, которые увлечены кулинарией и регулярно готовят дома. Они могут быть как опытными шеф-поварами, так и начинающими кулинарами, которые ищут новые рецепты и идеи для приготовления разнообразных блюд;
* занятых людей, ценящих удобство и эффективность в приготовлении пищи. Они могут искать быстрые и простые рецепты для приготовления в ограниченное время, а также инструменты для планирования ежедневного меню и составления списков покупок;
* людей, следящих за здоровьем, которые обращают внимание на питание и стремятся к здоровому образу жизни. Они могут быть заинтересованы в рецептах с низким содержанием калорий, вегетарианских или веганских блюдах, а также в возможности отслеживания пищевого рациона и калорийного потребления;
* социальных пользователей, которые хотят обмениваться рецептами, советами и опытом с другими участниками сообщества.

## Сценарии использования

Пользователь возвращается с работы и не знает, что приготовить на ужин. Он открывает приложение книга рецептов, использует функцию поиска, чтобы найти рецепт, который подходит для его доступных ингредиентов и времени.

Любитель кулинарии решает попробовать что-то новое. Он открывает приложение и ищет интересные или экзотические рецепты, чтобы попробовать что-то необычное и порадовать своих близких.

Пользователь ищет рецепты для особого случая, такого как день рождения, юбилей или праздничный ужин. Он использует приложение, чтобы найти рецепты для особенных блюд или десертов, которые станут украшением праздничного стола.

Лена открывает приложение и выбирает "Добавить рецепт". Она вводит название "Омлет с овощами", Лена добавляет ингредиенты: яйца, помидоры, перец, лук. Затем она описывает процесс: взбивает яйца, нарезает овощи, обжаривает на сковороде. Лена делает фото получившегося омлета. После проверки введённой информации, Лена нажимает "Добавить".

Рецепт доступен в ее списке. Она чувствует себя увереннее и готова попробовать ещё новые блюда не забывая уже проверенные рецепты.

## Сбор и анализ прототипов

В GooglePlay и AppStore существует множество приложений с на эту тему. В качестве примера были выбраны два приложения, выполняющих функции книги рецептов:

* Поваренная книга рецептов;
* Рецепты: Книга рецептов;

Оба этих приложения имеют главный экран, на котором есть список рецептов и меню функционала сверху для сортировки и поиска, как показано на рисунке 1.

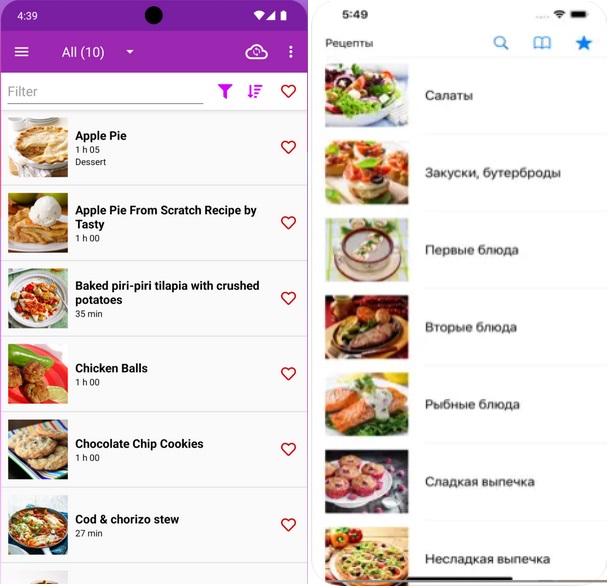


Рисунок 1 – Главный экран приложений-конкурентов

В приложениях на главном экране можно увидеть верхнюю панель, отвечающие за навигацию по приложению. На верхней панели в обоих приложениях почти одинаковые кнопки:

* поиск;
* сортировка;
* избранное.

Основные критерии приложений, вынесенные из использования представлены в таблице 1.

Таблица 1 – Сравнение приложений

|  |  |  |
| --- | --- | --- |
| Параметр | Поваренная книга рецептов (Google Play) | Рецепты: Книга рецептов (App Store) |
| Стоимость | Бесплатно | Бесплатно |
| Онлайн функционал | Требуется подключение к интернету | Требуется подключение к интернету |
| Постоянное подключение | Да | Да |
| Выбор изображения из галереи | Да | Да |
| Просмотр и поиск всех рецептов | Да | Да |
| Создание и редактирование рецептов | Да | Да |
| Наличие рекламы | Да | Нет |
| Платная подписка | Нет | Да |

## Основные негативные отзывы от пользователей приложений

Жалобы пользователей приложений-конкурентов:

* пользователи жалуются на частые сбои приложения или его вылеты во время использования, что мешает нормальной работе и доступу к рецептам;
* неудобство навигации и поиска по приложению;
* назойливая реклама в приложении, которая мешает пользоваться им комфортно и мешает просмотру рецептов;
* неэффективная система поиска рецептов, которая делает поиск трудным или неудобным;
* сложность в использовании интерфейса приложения, возможно, из-за запутанных меню или неясной навигации;
* технические проблемы, такие как медленная загрузка или нестабильная работа приложения.

Рассмотрев два приложения, имеющих похожий функционал, было решено написать приложение, которое имела бы простой и минималистичный интерфейс, с простым и похожим функционалом, с акцентом внимания на лёгкий ввод и надёжность данных, при этом имеющая также удобное для пользователя управление и полностью бесплатной. За основу будут взяты разные элементы из выше описанных приложений, так как каждое из них имеет как свои плюсы, так и минусы.

# Проектирование приложения

## 2.1UI/UX дизайн приложения

Дизайн проекта разработан в «Figma» – это онлайн-сервис, предлагающий пользователям богатый выбор инструментов для создания векторной графики, интерфейсов и разработки прототипов.

Для проекта были определены основные экраны:

* просмотр рецептов;
* добавление рецептов;
* редактирование рецептов;
* поиск рецепта.

Для темы приложения определена цветовая схема. Тема реализуется в тёмных тонах с яркими элементами, поэтому в ней основными цветами являются: серый, черный, белый, оранжевый (Рисунок 2).

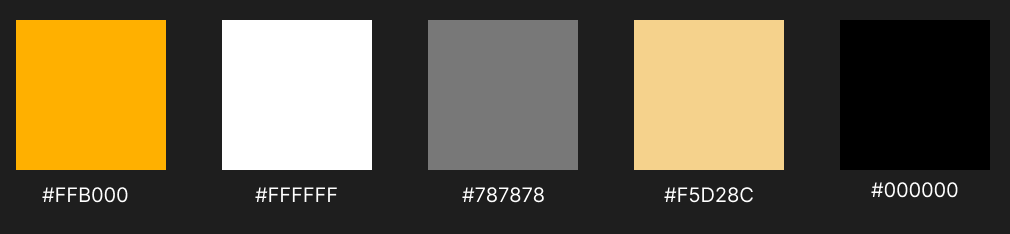


Рисунок 2 – Цветовая схема

Стоит заметить, что данные цвета не будет выделяться из цветовой схемы. Оранжевый цвет сообщает пользователю об основных интерактивных элементах, таких как кнопки поиска, добавления и обновления.

Ниже на рисунке 3 представлен логотип приложения.



Рисунок 3 – Логотип приложения

Цветовая схема логотипа состоит из: основных цветов приложения – тёмно-серого, и контрастного для него белого и бледно оранжевого цвета.

Сочетание именно этих цветов в такой позиции добавляет приложению

атмосферу домашней еды и простого завтрака, что может привлечь внимание пользователей.

Определившись с цветовой схемой приложения и создав его логотип, был разработан дизайн следующих экранов:

* рецепты (ListFragment)
* добавление рецептов (AddFragment)
* изменение рецептов (UpdateFragment)
* поиск рецепта (SearchFragment)

На рисунке 6 представлен дизайн приложения.

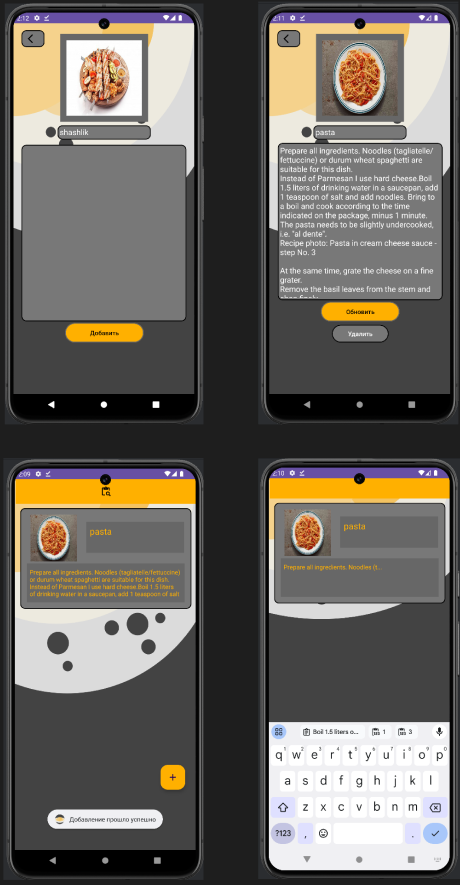


Рисунок 6 – Дизайн приложения

На рисунке 7 показано перемещение пользователя в приложении.

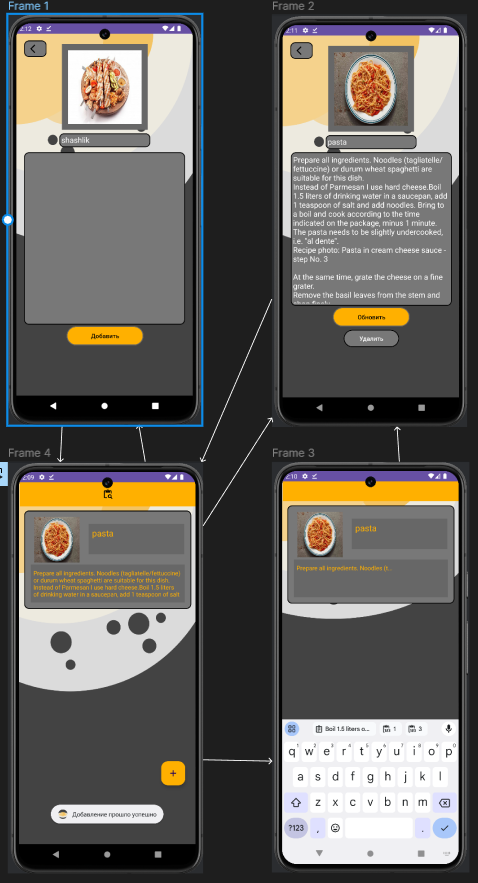


Рисунок 7 – Перемещение пользователя в приложении

При первом запуске приложения пользователь попадает на экран просмотра рецептов. При первом входе пользователь еще не добавил ничего в базу данных, при нажатии на плюс пользователь заходит на экран добавления рецепта. При заполнении полей рецепт добавляется в базу данных это продемонстрированно на рисунке 8.

У пользователя есть несколько путей взаимодействия с приложением на выбор: добавить еще рецептов, изменить уже добавленный рецепт.

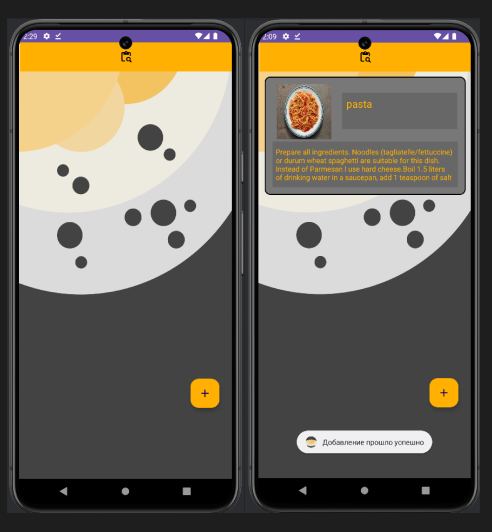


Рисунок 8– Изменение главного экрана от действий пользователя

Для отображения разметки в проект были добавлены «Адаптеры». Они необходимы в приложении для связи фрагментов и разметки для элементов списка recyclerView.

## 2.2 Выбор технологии, языка и среды программирования

Официальная интегрированная среда разработки (IDE) для разработки приложений под Android. Android Studio обладает мощными инструментами для разработки, тестирования, отладки и оптимизации приложений, что делает его предпочтительным выбором для разработчиков Android.

В качестве языка был выбран современный, выразительный и мощный язык программирования, предназначенный для разработки Android-приложений. Он обладает простым синтаксисом, безопасными по умолчанию нулевыми ссылками и позволяет писать более чистый и эффективный код по сравнению с Java. Kotlin также полностью совместим с Java, что позволяет постепенно переходить с Java на Kotlin.

Для реализации базы данных была выбрана библиотека Room это библиотека для работы с базами данных SQLite в приложениях Android. Она предоставляет удобный способ работы с базами данных через абстракции, такие как DAO (объект доступа к данным) и Entity (сущность), а также автоматически генерирует множество рутинного кода, связанного с работой с базой данных.

Для улучшения управления переходами между экранами в приложении была интегрирована библиотека Navigation. Она позволяет работать с фрагментами вместо активностей, что сделало написание кода более удобным. Благодаря этой библиотеке разработчики могут создавать переходы между экранами с помощью визуального интерфейса, что упрощает процесс создания и редактирования навигации в приложении. Кроме того, для улучшения передачи объектов между экранами был установлен плагин kotlin-parcelize, который облегчает передачу объектов классов между фрагментами.

# Разработка мобильного приложения

## 3.1 Разработка базы данных

Для создания базы данных в приложении использовалась библиотека Room. Для реализации базы данных с помощью Room также необходимы:

– интерфейсы DAO, создаются для каждой сущности, в них прописываются все запросы и методы для взаимодействия с таблицей базы данных;

– класс базы данных. К нему подключаются DAO и сущности, так же через него будет разорвана связь между DAO и репозиторием;

– репозиторий. Это необязательный компонент, но он является «правилом хорошего тона». В нем прописывается асинхронность методам из DAO;

– view model. Это тот компонент, к которому будут обращаться фрагменты для получения информации для отображения пользователю.

Схематично взаимодействие всех этих компонентов с пользователем изображено на рисунке 10.

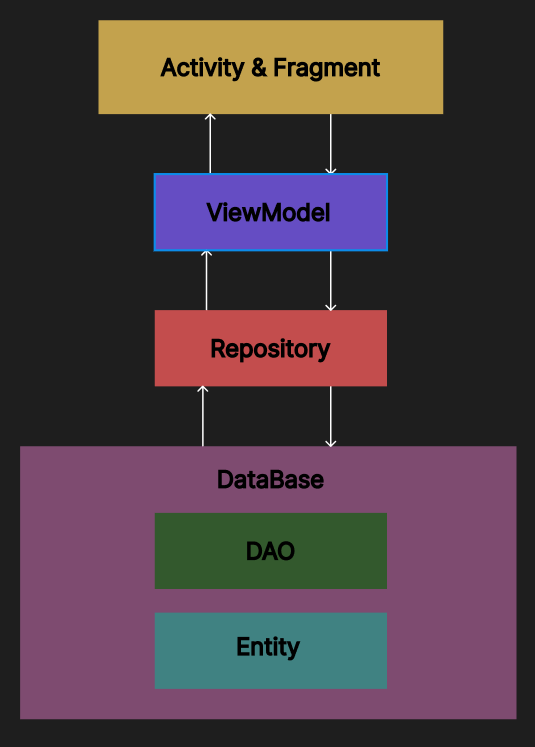


Рисунок 10 – Взаимодействие элементов в Room

В качестве основной сущности был создан класс Recipe, который был аннотирован как @Entity и связан с таблицей "Recipe". Класс Recipe также был аннотирован как @Parcelize для удобной передачи между различными компонентами приложения.

@Parcelize

@Entity(tableName = "Recipe")

data class Recipe(

@PrimaryKey(autoGenerate = true)

var IDRecipe: Int,//? = null,

var RecipeName: String,

var Image: String?,

var Discription: String ):

Parcelable

Этот код определяет класс данных Recipe для использования в базе данных с помощью Room в приложении Android, написанном на Kotlin. Каждый экземпляр Recipe представляет собой запись в таблице базы данных. В классе определены следующие поля:

* IDRecipe: Уникальный идентификатор рецепта, который автоматически генерируется базой данных при добавлении новой записи. Тип данных Int, по умолчанию равен 0.
* RecipeName: Название рецепта. Тип данных String.
* Image: Путь к изображению рецепта. Тип данных String? (может быть null).
* Discription: Описание рецепта. Тип данных String.

Аннотация @Entity указывает, что класс Recipe представляет собой таблицу базы данных с именем "Recipe". @PrimaryKey(autoGenerate = true) указывает, что поле IDRecipe является первичным ключом и автоматически генерируется при добавлении новой записи.

Аннотация @Parcelize указывает, что класс Recipe может быть сериализован для передачи между компонентами приложения (например, между фрагментами или активностями) с помощью механизма Parcelable.

## 3.2 Разработка мультимедийного контента

Весь мультимедийный контент в приложении был разработан с использованием языка разметки XML. Для вёрстки был использован дизайн, предварительно разработанный в приложении Figma. В процессе разработки дизайн подвергался изменениям, которые были внесены по мере необходимости.

Иконки и другие визуальные элементы приложения были импортированы в проект в формате XML-файлов, как показано на рисунке 11. Такой подход к хранению контента позволяет снизить размер приложения и избежать проблем с потерей качества мультимедийного контента. Все иконки расположены в папке "drawable", а визуальные элементы списков хранятся в папке "layout".

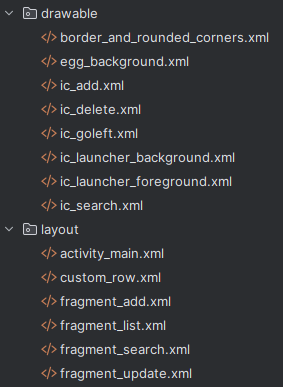


Рисунок 11 – Мультимедийный контент

## 3.3 Описание используемых плагинов

В проекте используются библиотеки с различными компонентами и функциями. Список всех библиотек, а также их описание представлены ниже

Таблица 2 – Библиотеки и их описание

|  |  |
| --- | --- |
| Библиотека | Описание |
|  |  |
| kotlin-kapt | Обеспечивает обработку аннотаций, этот плагин не будет использоваться напрямую, он необходим для работы Room |
| kotlin-parcelize | Позволяет передавать объекты классов между экранами. |
| androidx.navigation:  navigation-fragment-ktx:2.5.3,  androidx.navigation:  navigation-ui-ktx:2.5.3 | Библиотеки, предоставляют компоненты навигации для управления переходами между фрагментами и активностями в приложении. |
| androidx.lifecycle:  lifecycle-livedata-ktx:2.6.1,  androidx.lifecycle:  lifecycle-viewmodel-ktx:2.6.1 | Библиотеки, обеспечивающие жизненный цикл компонентов Android (таких как активности и фрагменты) и предоставляющие LiveData и ViewModel для удобной работы с данными в приложении. |
| org.jetbrains.kotlinx:kotlinx-coroutines-core | Богатая библиотека для корутин, разработанная JetBrains. Она содержит ряд высокоуровневых примитивов с поддержкой корутин, которые рассматриваются в этом руководстве, включая launch , async и другие. |

Данные библиотеки являются неотъемлемой частью приложения, обеспечивая его правильное функционирование и реализацию всех задуманных возможностей. Без этих библиотек приложение не сможет работать в полной мере, так как они предоставляют необходимые инструменты и функции, которые необходимы для его работы.

## 3.4 Описание разработанных процедур и функций

В мобильных приложениях процедуры и функции используются для создания модульного кода, который облегчает поддержку, тестирование и расширение приложения. Код, который выполняет одни и те же действия в разных частях приложения, можно упаковать в процедуры и функции, чтобы избежать его повторения.

В таблице 3 представлены методы мобильного приложения.

Таблица 3 – Методы приложения

|  |  |
| --- | --- |
| Метод | Описание |
| textViewLimit() | Ограничение максимальной длины отображаемого текста, отображаемого пользователю. Если текст больше длины ограничения, то текст отображается не полностью |
| saveImageToDatabase() | Сохраняет выбранное пользователем изображение для дальнейшего хранения |
| setImageToImageButton() | При нажатии на ImageButton пользователь может выбрать изображение из памяти устройства |
| openGalleryForImage() | Выбор изображения из галереи |
| setData() | Передает данные в адаптер для дальнейшего показа данных пользователю |
| replaceCurrentImage() | Замена изображения в записи базы данных |
| updateItem() | Обновление записи базы данных |
| getDatabase() | Получение экземпляра базы данных |
| deleteRecipe() | Удаление записи в базе данных |
| ImageSelection() | Проверяет заполненное поле картинки |
| inputCheck() | Проверяет заполнение полей для записи |
| OnBindViewHolder() | Заполняет список рецептов записями из базы данных |
| updateAdapter() | Фильтрует базу данных по введённой поисковой строке |
| InsertDataToDataBase() | Добавляет запись в базу данных из введённых данных |

Эти методы играют ключевую роль в функциональности мобильного приложения "Книга рецептов". Они обеспечивают удобство и функциональность для пользователей:

* сохранение и обновление изображений в базе данных позволяет пользователям добавлять, обновлять и удалять изображения для рецептов, делая их более информативными и привлекательными;
* выбор и загрузка изображений из галереи. Обеспечивает удобный способ добавления изображений к рецептам путем выбора изображений из галереи устройства;
* управление данными, позволяет эффективно управлять информацией о рецептах, включая добавление, обновление и удаление данных в базе данных;
* проверка заполнения полей необходимые поля заполнены перед сохранением или обновлением рецепта;
* фильтрация данных по поисковому запросу позволяет пользователям быстро находить нужные рецепты, фильтруя базу данных по введенному поисковому запросу;
* обновление интерфейса приложения с учетом внесенных изменений в данные, обеспечивая актуальное и удобное отображение информации.

При нажатии на recyclerView, создаются записи, в сущности, базы данных в «ListAdapter», которая отвечает за связь между сущностями. По завершении создания рецепта пользователя перенаправят на экран просмотра, содержащейся в образе, он показан на рисунке 13.

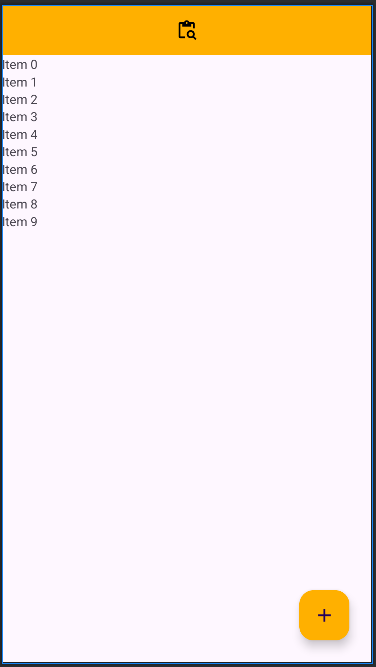


Рисунок 13 – Экран просмотра рецептов

Так же на этот экран можно попасть, через просмотр образа. В нижней части приложения реализован FloatingActionButton, который позволяет переключаться пользователю между просмотром рецептов и страницы с их добавлением. При нажатии на элементы списков для отображения пользователя направит на экран обновлении. Подобный экран почти аналогичен добавлению, они показаны на рисунке 14.

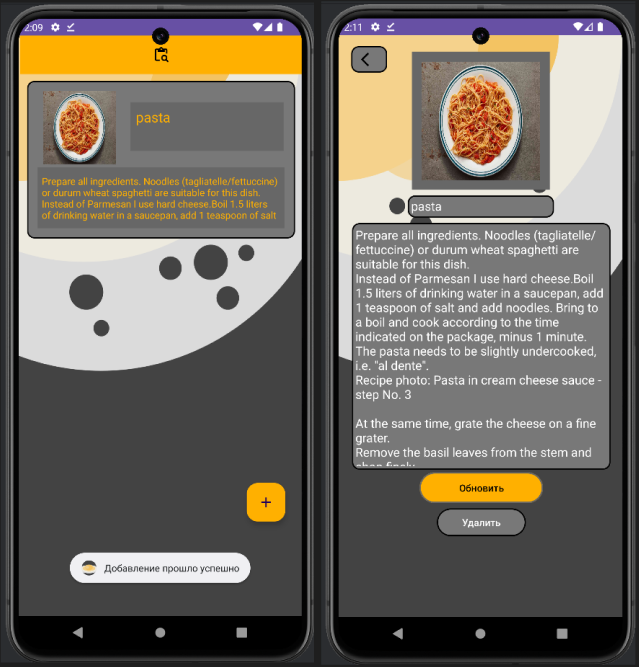


Рисунок 14 – Экран просмотра рецептов (слева) и обновления (справа)

На окне с обновлением рецептов существуют кнопки со следующими опциями:

* при нажатии на кнопку обновления пользователя отправит на экран просмотра рецептов, а данные о рецепте будут обновлены;
* при нажатии на кнопку удалить пользователю будет показано окно для подтверждения удаления. Это окно показано на рисунке 15.

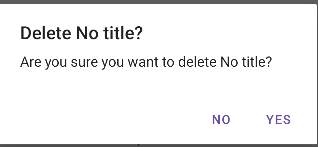


Рисунок 15 – Окно подтверждения удаления рецепта

# Тестирование

## Протокол тестирования дизайна приложения

Тестирование дизайна приложения проводится на Android SDK 31 с различной диагональю экранов для проверки разметки страниц и вёрстки приложения.

Примеры проверок отображения элементов на экране представлены на рисунках 16–19.

Рисунок 16 – Экраны просмотра рецептов

На экране не обнаружено ошибок, все кнопки отображаются корректно

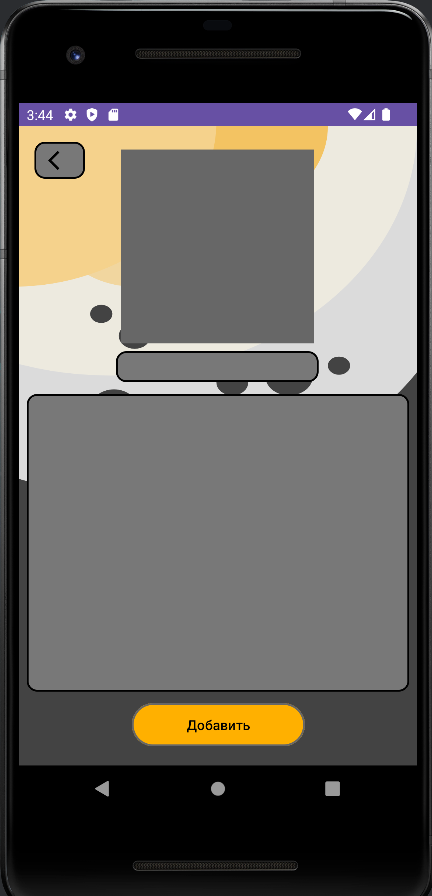
 

Рисунок 17 – Экраны добавления

Экран добавления так же отображается корректно на обоих устройствах, все элементы интерфейса расположены на своих местах.

Рисунок 18 – Экраны обновления рецептов

Экран обновления рецептов тоже отображается корректно. Верхняя и нижние кнопки нормальных размеров, элементы не сдвинуты.

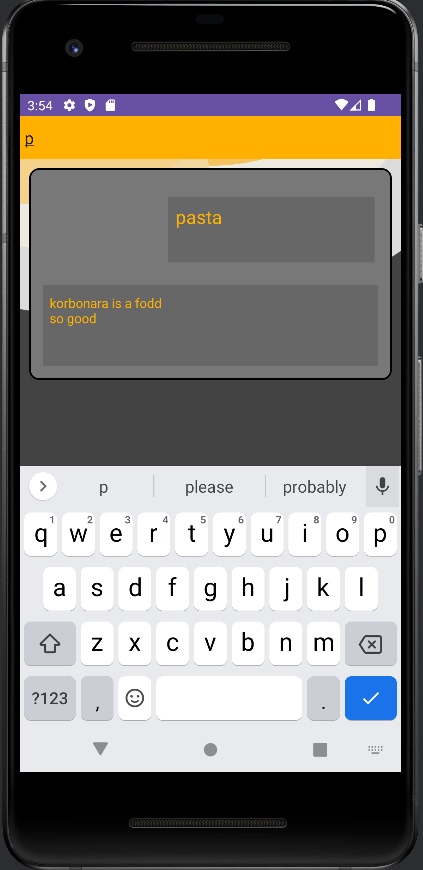
 

Рисунок 19 – Экраны поиска

Элементы экрана отображены корректно. Клавиатура не мешает разметке и не закрывает каких-либо элементов дизайна.

## 4.2 Протокол тестирования функционала приложения

Необходимо проверить функционал приложения. Для этого, для 3 случайных функции были разработаны TestCase.

В таблице 4 представлено тестирование функции добавления рецепта.

Таблица 4 – Тестирование функции добавление рецепта.

|  |  |  |
| --- | --- | --- |
| **Название**: | InsertDataToDataBase() | |
| **Функция**: | Добавление рецепта | |
| **Действие** | **Ожидаемый результат** | **Результат теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение | Приложение запустилось |  |
| Нажать на плюс | Открылся фрагмент для добавления |  |
| **Шаги теста (positive)**: |  | **Шаги теста (positive)**: |
| Заполнить поля названием и описанием | Поля заполнены корректно | Заполнение полей названием и описанием |
| Нажатие на кнопку добавить | Открылся фрагмент просмотра рецептов с добавленным рецептом | Открылся фрагмент просмотра рецептов с добавленным рецептом |

В таблице 5 представлено тестирование функции поиска рецептов.

Таблица 5 – Тестирование функции поиска рецептов.

|  |  |  |
| --- | --- | --- |
| **Название**: | UpdateAdapter() | |
| **Функция**: | Фильтрует базу данных по введённой поисковой строке | |
| **Действие** | **Ожидаемый результат** | **Результат теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение | Приложение запустилось |  |
| Нажать на верхний поиск | Открылся фрагмент поиска |  |
| **Шаги теста (positive)**: |  | **Шаги теста (positive)**: |
| Заполнить строку поиска | Фрагмент выводит рецепт с похожим названием | пройден |

Разработанные TestCase демонстрируют корректную работу функций приложений. Так же в процессе тестирования не выявлено ошибок в дизайне приложения и его логике. Каждый экран прошёл проверку на разных устройствах на корректное отображение элементов соответствующих экранов.

Заключение

В заключении можно отметить, что при разработке мобильного приложения для рецептов были учтены предпочтения и потребности пользователей. Мы успешно внедрили систему CRUD для управления основными данными в базе. Интерфейс приложения разработан с акцентом на удобство использования, минимизируя количество шагов для достижения цели пользователя. Каждый элемент на основных экранах функционален, что обеспечивает эффективное взаимодействие.

Одним из ключевых принципов приложения является отсутствие ограничений, которые часто встречаются в конкурирующих приложениях, таких как необходимость постоянного подключения к сети или ограничения бесплатного функционала.

Тестирование дизайна было важным этапом, подтверждающим его корректность и эффективную работу. Это гарантирует, что разработанное приложение соответствует нашим целям и требованиям. В целом, разработка электронного гардероба привела к созданию функционального и удобного приложения, а проделанная работа успешно достигла поставленных целей курсового проекта, служа основой для его будущего развития и улучшения.

Библиография

Нормативно-правовые акты:

1. ГОСТ Р 2.105-2019. ЕСКД. Общие требования к текстовым документам. – Москва: Стандартинформ, 2019. – 36 с

Электронные ресурсы:

1. Android Developers [Электронный ресурс]. – Документация Android Studio. – URL: https://developer.android.com/develop (дата обращения: 26.04.2024)
2. Figma [Электронный ресурс]. – Прототипирование проектов.– https://www.figma.com/ (дата обращения: 01.04.2024)
3. KotlinLang [Электронный ресурс]. – Документация Kotlin. – URL: https://kotlinlang.org/docs/home.html (дата обращения: 28.04.2024)
4. Developer android [Электронный ресурс]. – Документация Room. Вытягивание данных с базы данных – URL: https://developer.android.com/training/data-storage/room (дата обращения: 28.04.2024)
5. Developer android [Электронный ресурс]. – Документация Navigation – URL: https://developer.android.com/guide/navigation (дата обращения: 28.04.2024)
6. Material Design [Электронный ресурс]. – Bottom Navigation. – URL: https://material.io/components/bottom-navigation/android (дата обращения: 28.04.2024)

Приложение А  
 (Обязательное) Тех. Задание

Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж

имени Б.С. Галущака»

РАЗРАБОТКА Мобильного приложения

«Книга рецептов»

Техническое задание

НАТКиГ.200300.010.000ПЗ

Выполнил:

Студент группы ПР-21.101

Борисов Ф. Е.

2024

**Содержание**

Введение 29

1 Назначения разработки 29

2 Требования к мобильному приложению 29

2.1 Требования к функциональным характеристикам 29

2.2 Требования к надёжности 30

2.3 Условия эксплуатации 30

2.4 Требования к составу и параметрам технических средств 30

2.5 Требования к информационной и программной совместимости 30

2.6 Требования к защите информации 30

2.7 Требования к маркировке и упаковке 30

3 Требования к программной документации 30

4 Технико-экономические показатели 31

5 Стадии и этапы разработки 31

6 Порядок контроля и приёмки 32

**Введение**

Настоящее техническое задание распространяется на разработку мобильного приложения «Книга рецептов», используемого для просмотра рецептов.

Наименование приложения: «Рецептуар».

Краткая характеристика области применения: мобильное приложение предоставляет поиск и создание новых рецептов.

Основанием для проведения разработки является Протокол №6 от 21 февраля 2024 года.

Наименование темы разработки − «Разработка мобильного приложения «Книга рецептов»».

Условное обозначение темы разработки – «Книга рецептов».

**1 Назначение разработки**

Основное назначение приложения заключается в:

* обеспечении удобного и понятного интерфейса приложения для пользователя;
* предоставление большого числа рецептов.

Лица, которые могут работать с данной системой:

пользователь приложения – добавлять, удалять, редактировать записи о рецептах, осуществлять поиск.

администратор – управляет полностью всем приложением, следит за его работоспособностью, обновляет информацию в файлах данных приложения;

**2 Требования к мобильному приложению**

**2.1 Требования к функциональным характеристикам**

Требования к составу выполняемых функций:

* реализация взаимодействия с рецептами;
* правильное расположение элементов управления;
* возможность запускать без соединения с интернетом.

**2.2 Требования к надёжности**

Обеспечение устойчивого функционирования должно выполняться несколькими действиями:

* организация стабильной работы устройства.

Приложение должно контролировать входную информацию:

* соблюдение контроли версии устройства.

**2.3 Условия эксплуатации**

Пользователь должен иметь практические навыки использования мобильного устройства под управлением операционной системы Android.

**2.4 Требования к составу и параметрам технических средств**

Для работы приложения необходимо мобильное устройство с установленной операционной системой Android с API не ниже 28.

**2.5 Требования к информационной и программной совместимости**

Проектирование взаимодействия с файловой системой должно быть выполнено в рамках разработки курсового проекта. При разработке взаимодействия с файловой системой должен быть использован язык программирования Kotlin.

**2.6 Требования к защите информации**

Вся информация, хранимая в базе данных расположена локально. Приложение не требует от пользователя дополнительных разрешений.

**2.7** **Требования к маркировке и упаковке**

Требования к маркировке и упаковке не предъявляются.

**3 Требования к программной документации**

Состав программной документации должен включать в себя:

* техническое задание;
* пояснительная записка.

**4 Технико-экономические показатели**

Экономические преимущества разработки и ориентировочная экономическая эффективность не рассчитывается.

**5 Стадии и этапы разработки**

Таблица 1 – Стадии разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № | **Этапы разработки КП** | **Сроки выполнения** | **Отчётность** |
| 1 | Определение цели и задач, объекта и предмета исследования | 24.04.2023 | Пояснительная записка |
| 2 | Описание предметной области | 24.04.2023 | Пояснительная записка |
| 3 | Выбор технологии, языка и среды программирования | 24.04.2023 | Пояснительная записка |
| 4 | Оформление технического задания | 24.04.2023 | Техническое задание |
| 5 | Проектирование UI/UX дизайна | 25.04.2024 | Спецификации программного обеспечения |
| 6 | Разработка мобильного приложения | 27.04.2024 | Схема структурная системы и спецификации компонентов |
| 7 | Разработка базы данных | 27.04.2024 | Программный  продукт |
| 8 | Отладка и тестирование приложения | 30.04.2024 | Тексты программных компонентов |
| 9 | Оформление документации | 04.05.2024 | Программная  документация |
| 10 | Защита | 06.05.2024 |  |

**6 Порядок контроля и приёмки**

Виды испытаний – защита курсового проекта.

Общее требования к приёмке:

* техническое задание;
* пояснительная записка;
* программный продукт.

Приложение Б

Навигация приложения через Navigation

