Оглавление

[1. Постановка задачи 2](#_Toc152789132)

[2. Описание разработки приложения 3](#_Toc152789133)

[3. Описание интерфейса приложения 6](#_Toc152789134)

[4. Навигация 10](#_Toc152789135)

[5. Список рецептов 11](#_Toc152789136)

[6. Текущий элемент 12](#_Toc152789137)

[7. Обновление базы данных 13](#_Toc152789138)

[8. Добавление блюд в избранное 13](#_Toc152789139)

[9. Часы 16](#_Toc152789140)

[В данной лабораторной работе необходимо разработать модуль проект для часов. 16](#_Toc152789141)

[10. Заключение 16](#_Toc152789142)

# **1. Постановка задачи**

**ЛР**

Требуется разработать приложение, включающее в себя два Activity:

1. Первое Activity должно соответствовать предложенному дизайну:

Изображение выглядит как текст, чек

Автоматически созданное описание

Изображение логотипа находится в ресурсах (https://clck.ru/YBvCK). Цвет фона и кнопки выбирается студентом самостоятельно и отличается от стандартного. Расположение элементов – ConstraintLayout (вертикальный).

1. На первом Activity пользователь вводит информацию в 2 поля (E-mail, пароль). При нажатии на кнопку «Войти» и проверке совпадения полей имеющихся значений E-mail и пароль, пользователь переходит на вторую Activity. Если одно из полей не совпало, то текст становится красным.
2. Приложение должно поддерживать вертикальную и горизонтальную ориентацию. При изменении ориентации на горизонтальную все элементы видны, сохраняют свое расположение, 2 текстовых поля и кнопка масштабируются на всю доступную ширину экрана.
3. Список имеющихся значений E-mail, пароль хранятся в массиве строк в ресурсах (5 значений).
4. Вторая Activity создается по шаблону Bottom Navigation Activity и остается без изменений.

# **2. Описание разработки приложения**

В ходе данной лабораторной работы разрабатывалось приложение, основным направлением которой является кулинария.

Использовалось два вида Activity: Empty View Activity, Bottom Navigation Views Activity. Регистрация в соответствии с заданием требовала хранение данных в виде массива строк, что несколько небезопасно и это не предполагает возможность регистрации новых пользователей. Для осуществления задумки о создании базы данных, которая будет более автоматизирована, в отличие от предлагаемого варианта, использовалась SQLite Database. Работа с Базой данных осуществляется через класс DbHelper, который выполняет роль контроллера и распределителя данных.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 1 - класс DbHelper

**SQLite Database** это быстрая и легкая встраиваемая однофайловая СУБД на языке C, которая не имеет сервера и позволяет хранить всю базу локально на одном устройстве. Для работы SQLite не нужны сторонние библиотеки или службы.

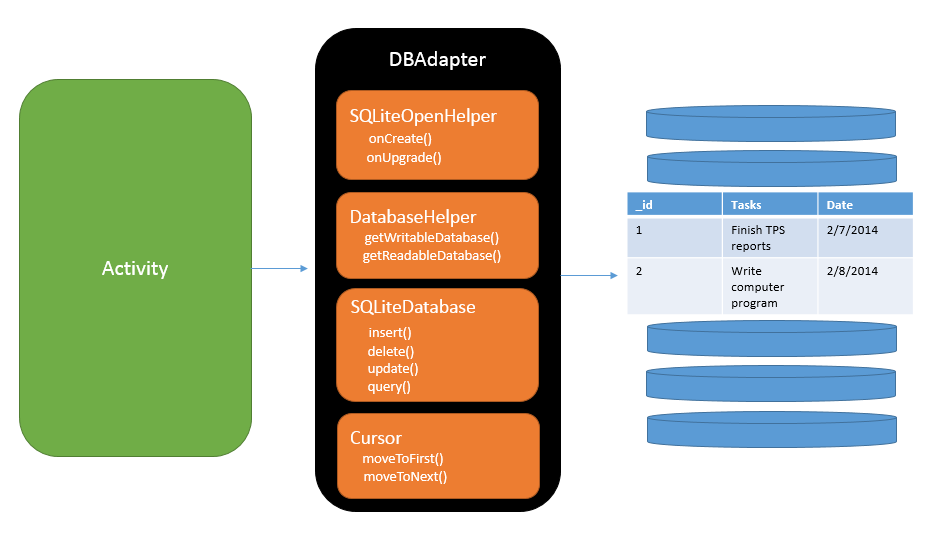


Рисунок 2 - Процесс работы SQLite через DbHelper

Более того кроме добавления функции Регистрации, осуществлены дополнительные проверки при регистрации пользователя и его авторизации. (Изменение с предоставленным заданием: при различного рода ошибках меняется не цвет, а появляется сообщение с типом ошибки, сделано для поддержания минималистичного дизайна)

Предлагаемые сценарии пользователя:   
Новый пользователь -

* Вводит данные в графу Авторизации
* Предупреждаем об отсутствии аккаунта
* Переходит по ссылке на регистрацию
* Регистрируется, и вновь переходит на страницу Авторизации
* Входит в приложение

Существующий пользователь, ошибка в пароле -

* Вводит неправильные данные в графу Авторизации
* Предупреждаем об ошибке
* Переходит по ссылке на регистрацию
* Предупреждаем о существовании аккаунта
* Далее либо регистрируется как новый пользователь, либо вводит правильные данные в Авторизации
* Входит в приложение

Существующий пользователь -

* Вводит данные в графу Авторизации
* Входит в приложение

Соответственно также предусмотрены ошибки пустого поля и отдельно неправильного пароля.

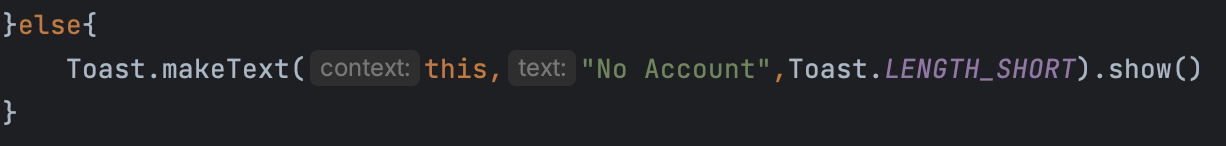


Рисунок 3 - Пример ошибки (Отсутствие аккаунта)

Сам текущий аккаунт сохраняется непосредственно в класс **User** для дальнейшего использования уже во внутреннем функционале приложения.

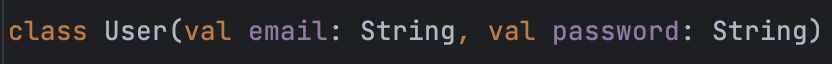


Рисунок 4 - класс User

# **3. Описание интерфейса приложения**

Дизайн интерфейса будет дорабатываться в процессе выполнения дальнейших лабораторных работ, с учетом уже требуемого от приложения функционала.

Примерный сценарий пользователя будет продемонстрирован в скриншотах последовательно, для удобства понимания работы приложения.

Вначале пользователь знакомится с начальным окном **Авторизации**, окно доступно при двух видах ориентации (портретная, альбомная), как другие окна. Фраза **«First time here?»** является ссылкой на окно Регистрации.

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 5 – Авторизация(портретная)

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 6 - Авторизация(альбомная)

При попытке входа с **пустыми ячейками** (или одной из) будет выведено сообщение о пустых полях.

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 7 - Ошибка пустого поля

**Успешная регистрация** также оповещает уведомлением.

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 8 - Создание аккаунта

Далее рассмотрены другие ошибки и предупреждения, которые обрабатывает приложение в процессе регистрации. Это позволяет пользователю подробнее понять на каком этапе пользователь допустил ошибку.

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 9 - попытка повторной регистрации

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 10 - вход в несуществующий аккаунт

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 11 - Вход с неправильным паролем

После успешного входа в аккаунт идет переадресация на основное меню приложения с нижней навигацией. Скрины представлены ниже.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 12 - Меню(портретная)

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 13 - Меню(альбомная)

4. Навигация

После авторизации происходит переход в основное меню приложения.  
Наиболее популярное и удобное перемещение между основной информацией приложения реализовано в качестве Bottom\_navigation.  
Современна реализация предполагает создание не трех отдельных Активностей, а создание одной активности с взаимозаменяемыми фрагментами. В нашем случае это 3 фрагмента.

Изображение выглядит как текст, снимок экрана, овощи, еда

Автоматически созданное описание

Рисунок 14 – Навигация

5. Список рецептов

Далее на самом левом экране Search можно увидеть список рецептов, после которого передастся значение в активность выбранного рецепта.

Каждый элемент списка оснащен названием блюда и его изображением для понимания внешнего вида блюда.

Изображение выглядит как текст, снимок экрана, овощи, еда

Автоматически созданное описание

Рисунок 15 - Список рецептов

6. Текущий элемент

В соответствии с лабораторной работой 5, было добавлена страница выбранного блюда на странице рецептов.

Вся информация поступает из json-файла по ссылке <https://raw.githubusercontent.com/Lpirskaya/JsonLab/master/recipes2022.json>

Таким образом мы выводим актуальную информацию о блюде после того как мы выберем его.

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 16 - Информация о блюде

7. Обновление базы данных

В начале лабораторной работы была добавлена база данных, чтобы поддерживать создание новых аккаунтов и подгрузку информации о пользователе при входе в приложение.

Теперь же добавлена дополнительно информация о том, какие блюда понравились пользователю.

8. Добавление блюд в избранное

Теперь в приложении пользователю доступно добавление блюд в избранное.  
Сделать это можно с помощью долгого нажатия на ячейку понравившегося блюда.  
В данном случае появится диалоговое окно, где вы можете подтвердить конечный выбор.

Изображение выглядит как текст, снимок экрана, мультимедиа, программное обеспечение

Автоматически созданное описание

Рисунок 17 - Диалоговое окно

Для того, чтобы просмотреть ваши любимые блюда необходимо лишь перейти на третью вкладку нижней навигации.

Изображение выглядит как текст, снимок экрана, овощи, Быстрое питание

Автоматически созданное описание

Рисунок 18 - Избранные блюда

Также для проверки мы перешли в аккаунт, где не было добавлено ни одного блюда.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Автоматически созданное описание

Рисунок 19 - Пустая страница

9. Часы

В данной лабораторной работе необходимо разработать модуль проект для часов.

Изображение выглядит как текст, снимок экрана, круг, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, фрукт, еда, оружие

Автоматически созданное описание

Рисунок 20 - Интерфейс проекта для часов

10. Заключение

В ходе данной лабораторной работы было мы познакомились с возможностью создания различного рода окон. Поддерживаемые ориентации приложения, смена дизайна при повороте экрана. Работа с данными и создание полного процесса регистрации. Работа с разнообразным набором инструментов для создания дизайна. Возможность построения полного сценария пользовательской работы с приложением.

Использовалась среда разработки Android Studio на языке Kotlin.