



Пловдивски университет „Паисий Хилендарски”
Факултет по математика и информатика

Курсова работа

По дисциплина „Мобилни приложения“

На тема: „Мениджър на задачи“

Изготвил: Онур Солак
Специалност: СИ Редовно
Факултетен номер: 2101321005

Проверил:
доц. д-р С. Хаджиколева

Пловдив, 2023 г.

Съдържание

1.	Увод.....	3
2.	Основни функционалности	3
3.	Структура на проекта.....	3
4.	Използвани технологии и библиотеки	5
5.	Инструкции за употреба	6
5.1	Преглед на задачите в отделни списъци.....	6
5.2	Създаване на нова задача	7
5.3	Преглед и редактиране на детайлите за дадена задача	10
5.4	Маркиране на задача като приключена	12
5.5	Маркиране на задача като незавършена	13
5.6	Изтриване на задача.....	14
5.7	Извличане на мотивиращи цитати.....	15
6.	Имплементация	17
6.1	Извличане на задачи от базата данни	17
6.2	Десериализация на JSON обект до обект по избор.....	18
6.3	Извличане на цитати с Retrofit и тяхното визуализиране	20
6.4	Допълнителна информация	21
7.	Заключение	21
8.	Библиография	22
9.	Списък на фигурите.....	24

1. Увод

Има различни хора на този свят и аз се причислявам към тези, които имат лоша памет. За щастие, има различни софтуери, които помагат на хората да са по-организирани. Пример за такъв софтуер е едно мобилно приложение на моя телефон, което ми позволява да си водя записки, в които мога да опиша различни задачи от моето ежедневие. Това приложение ме вдъхнови да разработя моя версия на този тип приложения, но с допълнителна функционалност. Целта на моята курсова работа е не само да позволи на потребителя да създава и управлява задачи, но също така да го мотивира да си приключи задачите.

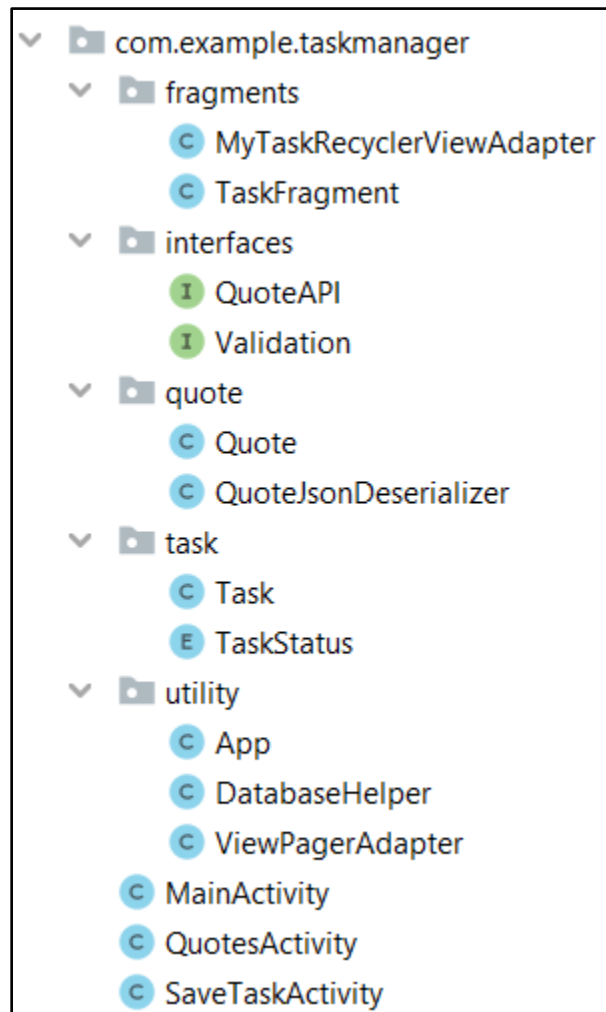
2. Основни функционалности

Приложението ми включва следните ключови функционалности:

- Създаване на нова задача;
- Преглед на задачите в отделни списъци, в зависимост от тяхното състояние;
- Преглед и редактиране на детайлите за дадена задача;
- Маркиране на задача като приключена;
- Маркиране на задача като незавършена;
- Изтриване на задача;
- Извличане на мотивиращи цитати.

3. Структура на проекта

Тук ще бъде представена кратка информация за структурата на проекта.



Фигура 1 Дървовидна структура на проекта

- Fragments – пакет, който съдържа класове за работа с фрагменти:
 - MyTaskRecyclerViewAdapter – помага с манипулирането на данните в списъците със задачи;
 - TaskFragment – фрагмент, показващ задачите в списък;
- Interfaces – пакет, който съдържа интерфейси:
 - QuoteAPI – интерфейс, който извлича мотивиращи цитати от Zen Quotes API [1];
 - Validation – интерфейс за валидация;
- Quote – пакет, съдържащ класове за работа с цитати:

- Quote – клас, описващ данни за цитат;
- QuoteJsonDeserializer – клас за десериализация на информация за цитати от формат JSON [2];
- Task – пакет, съдържащ класове за работа с задачи:
 - Task – клас, който описва данни за задачи и предоставя методи за работа с тях;
 - TaskStatus – енумерация за състоянието на задачите [3];
- Utility – пакет, съдържащ помощни класове, необходими за правилното функциониране на приложението:
 - App – клас, който осигурява достъп отвсякъде до контекста на приложението;
 - DatabaseHelper – клас, осигуряващ възможност за работа с базата данни;
 - ViewPagerAdapter – клас, помагаш при изобразяването на правилния фрагмент, съдържащ списък със задачи [4];
- MainActivity – главното Activity [5], което представлява началния екран на приложението. В него са показани списъците със задачи;
- QuotesActivity - Activity, което позволява на потребителя да извлича и чете вдъхновяващи цитати;
- SaveTaskActivity – Activity, което дава възможност на потребителя да преглежда, създава и редактира задачи.

4. Използвани технологии и библиотеки

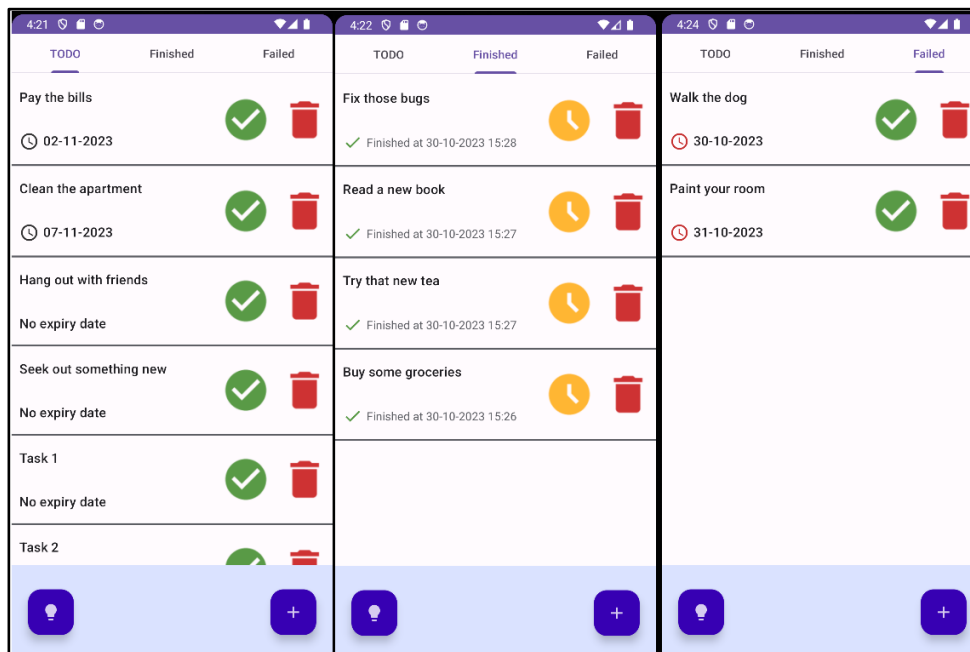
- Android Studio [6];
- Java [7];
- Android SDK [8];
- Gson [9] [10];
- Retrofit [11] [12];
- SQLite [13];
- SQLiteOpenHelper [14];
- ViewPager2 [15];

- FragmentStateAdapter [4];
- Fragments [16];
- RecyclerView [17];
- DatePickerDialog [18] [19];
- GitHub [20] [21].

5. Инструкции за употреба

5.1 Преглед на задачите в отделни списъци

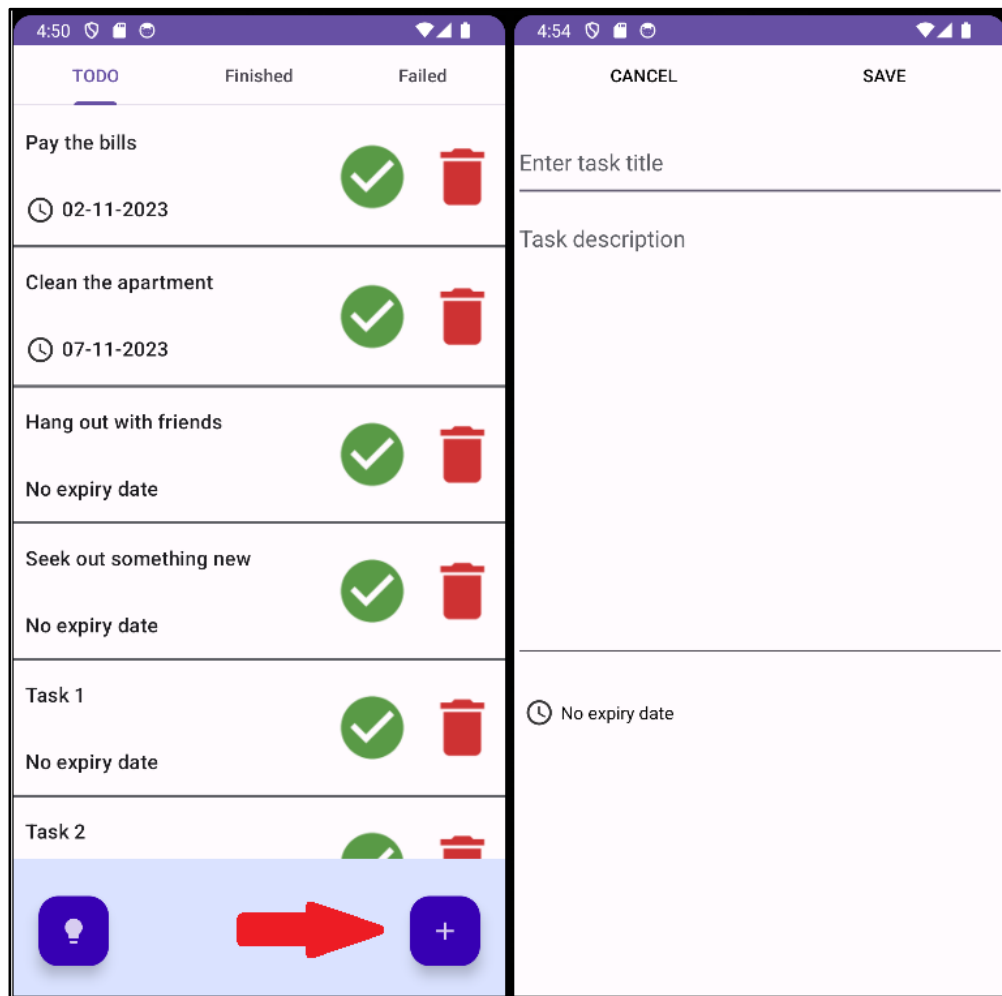
При стартиране на приложението, потребителят се озовава в началния изглед, откъдето може да разгледа всички видове задачи. Задачите могат да са „незавършени“ (TODO), „приключени“ (Finished) и „неуспешни“ (Failed). Незавършените задачи са тези, които нямат краен срок или крайният им срок не е изтекъл. Една задача става „приключена“, когато потребителят я маркира като такава. Неуспешните задачи са тези, чийто краен срок е изтекъл. Потребителят може да разгледа различните списъци със задачи като натисне върху съответния раздел, който се намира горе на екрана.



Фигура 2 Главният изглед с различните списъци със задачи

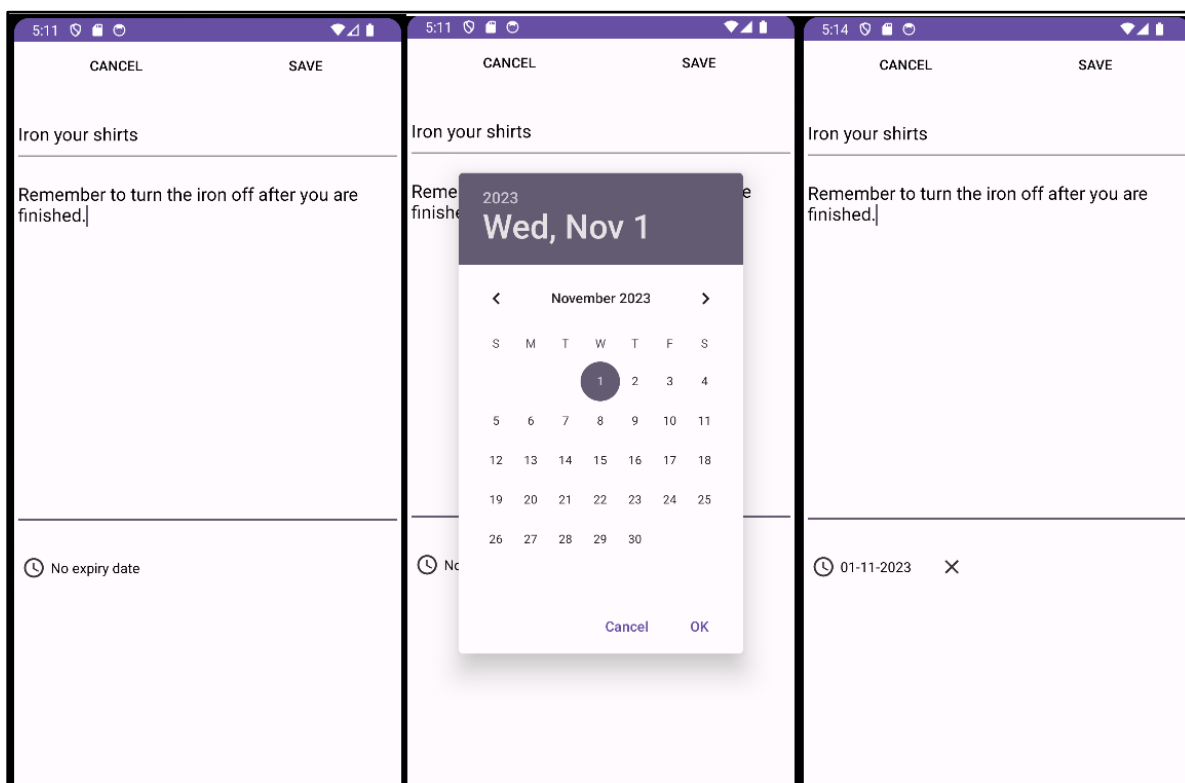
5.2 Създаване на нова задача

Потребителят може да отвори изгледа за създаване на нова задача като натисне бутона за добавяне, който се намира в долния десен ъгъл на екрана. Това ще доведе до отваряне на формуляра за попълване на данни относно задачата. Най-отгоре са бутоните „Cancel“ и „Save“, които са съответно за отменяне на действието и запазване на промените. Под тях се намират полетата за въвеждане на име и описание на задачата. Името на задачата задължително трябва да бъде въведено. Описанието на задачата може да не се въвежда. Най-отдолу се намира мястото за избор на крайна дата за задачата. Не е задължително да се посочва крайна дата.



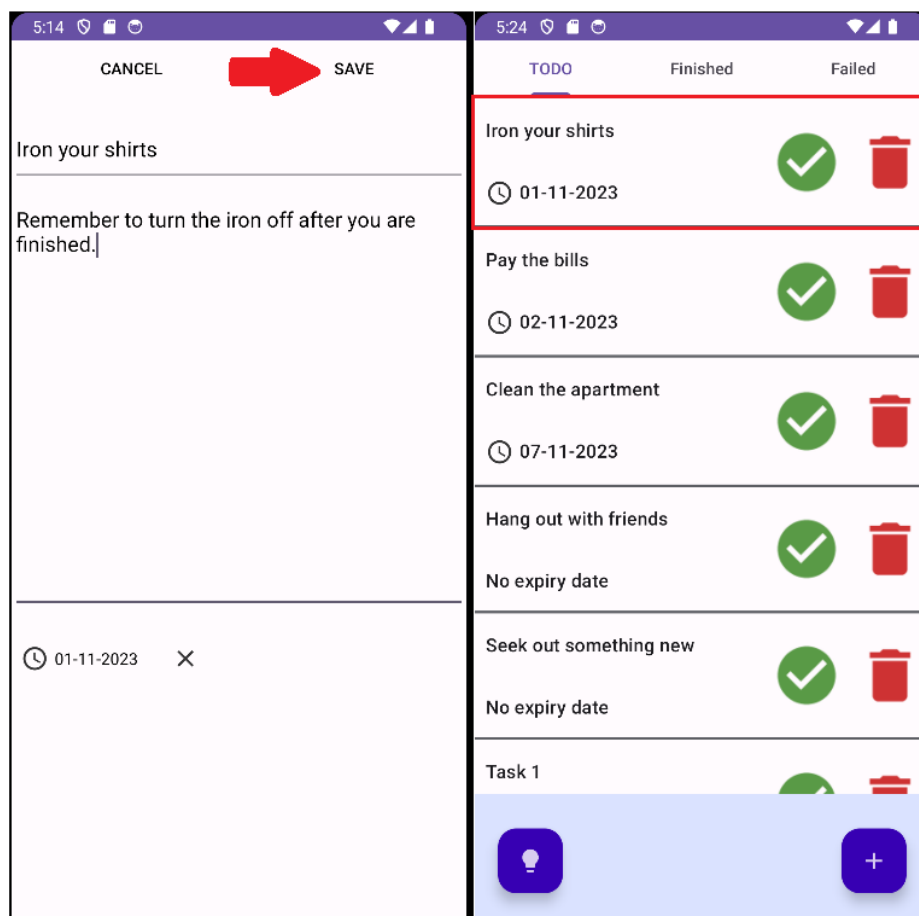
Фигура 3 Отваряне на изгледа за създаване на задача

След посочване на крайна дата, потребителят може да я махне чрез натискане на хикса ОТДЯСНО.



Фигура 4 Попълване на данни за задачата

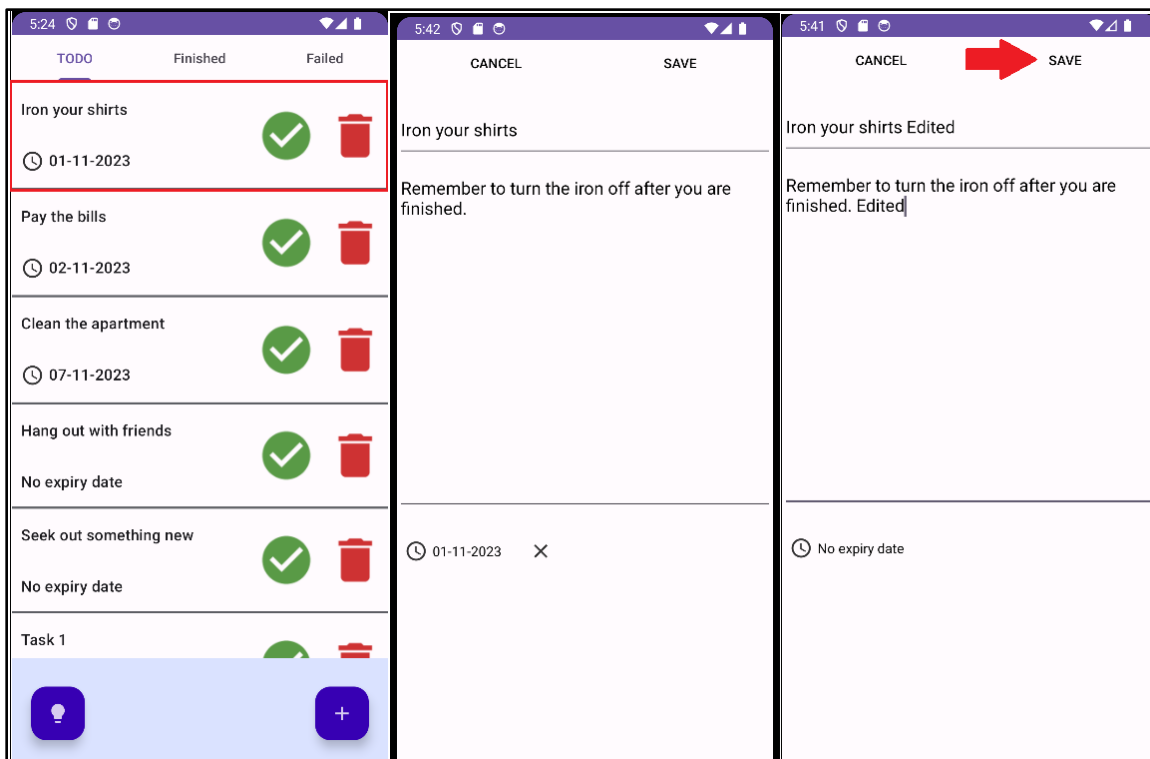
След като е готов с попълването, потребителят натиска „Save“. Това води до създаване на задачата и потребителят е върнат обратно към списъка с незавършени задачи.



Фигура 5 Създаване на задача

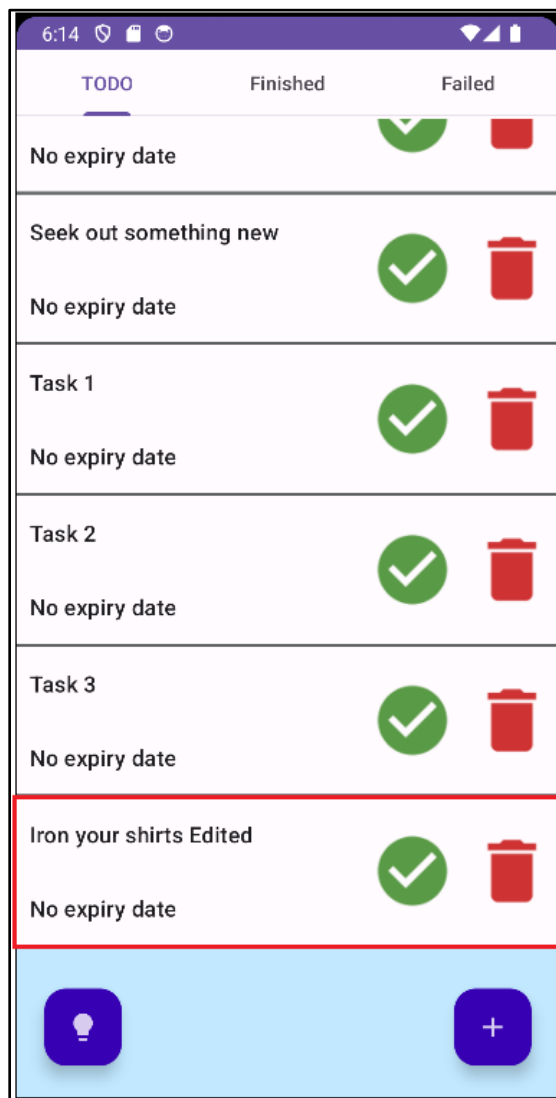
5.3 Преглед и редактиране на детайлите за дадена задача

Потребителят може да отвори съществуваща задача като натисне върху нея. След това той редактира данните за задачата и избира „Save“. Действието е аналогично на това за създаване на задача.



Фигура 6 Редактиране на задача

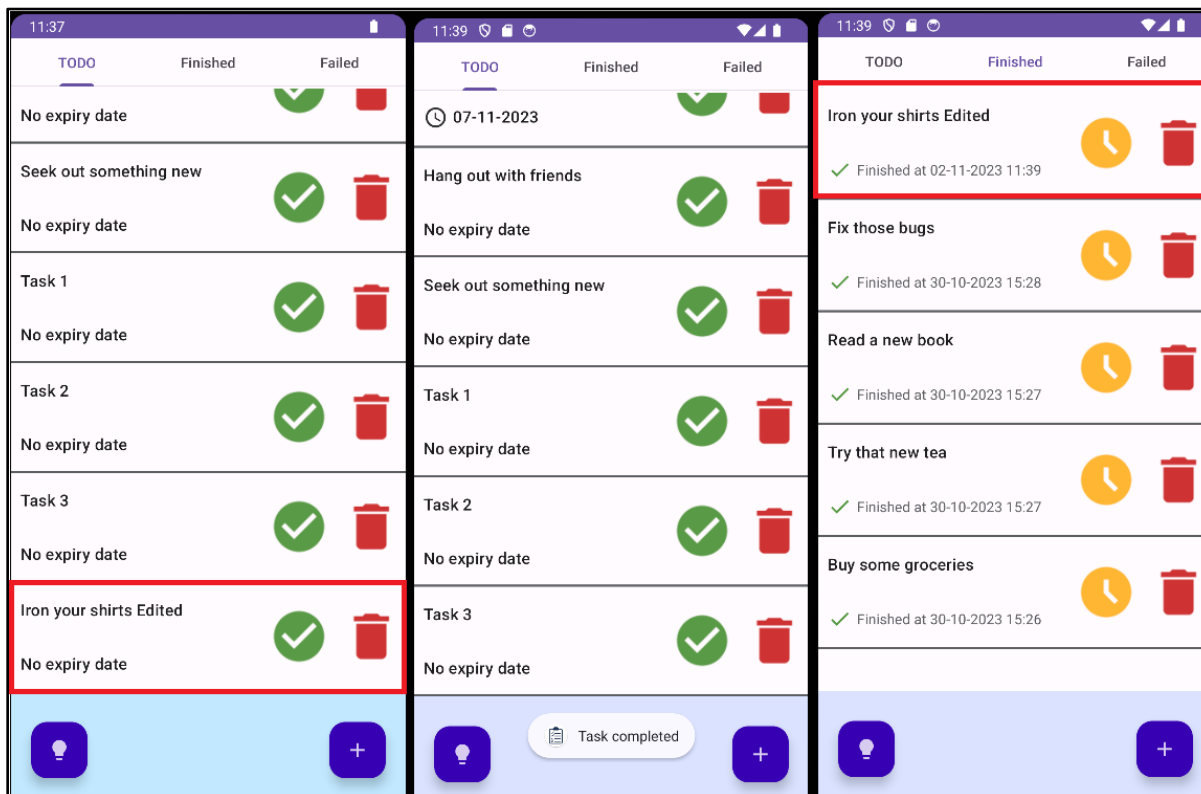
След успешна редакция, потребителят ще бъде върнат към списъка, откъдето е редактирал задачата.



Фигура 7 Резултат от редакция на задача

5.4 Маркиране на задача като приключена

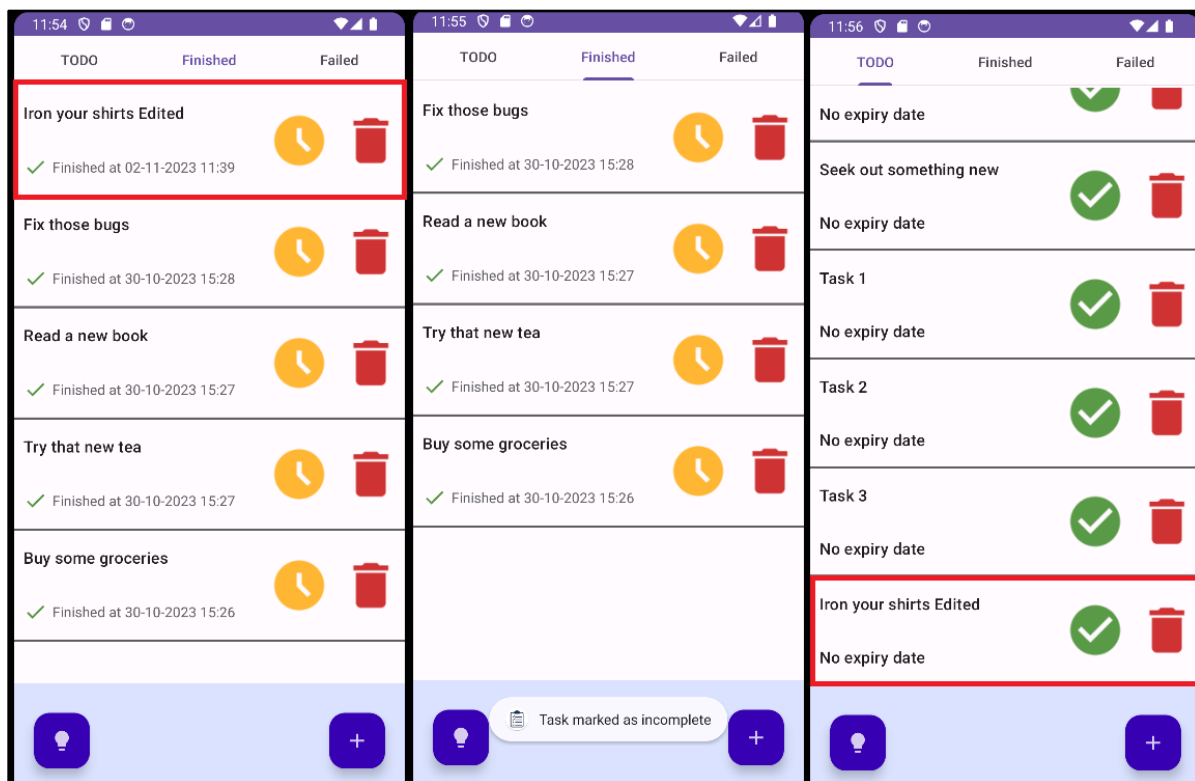
Незавършените (TODO) и неуспешните (Failed) задачи могат да бъдат маркирани като приключени чрез натискане на зелената отметка срещу тях.



Фигура 8 Маркиране на задача като приключена

5.5 Маркиране на задача като незавършена

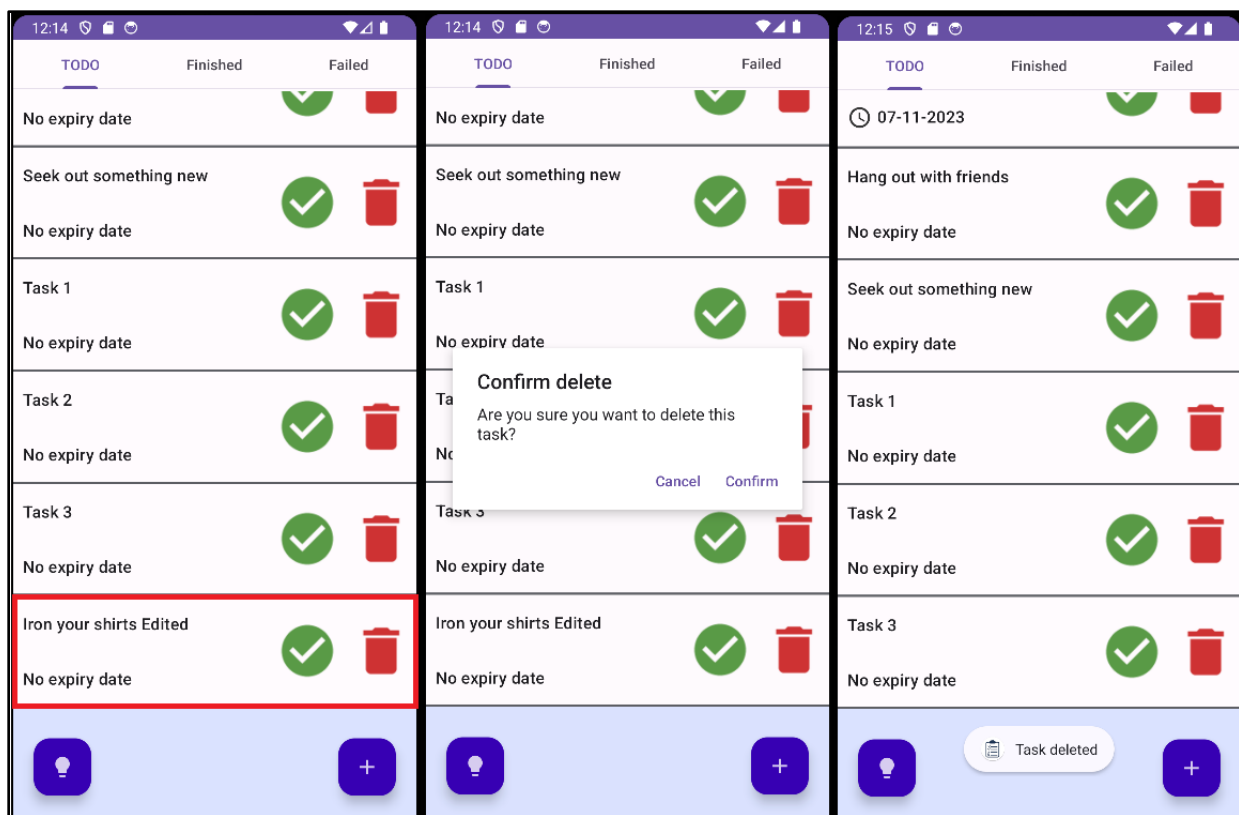
Приключените (Finished) задачи могат да бъдат маркирани като незавършени чрез натискане на жълтия часовник срещу тях.



Фигура 9 Маркиране на задача като незавършена

5.6 Изтриване на задача

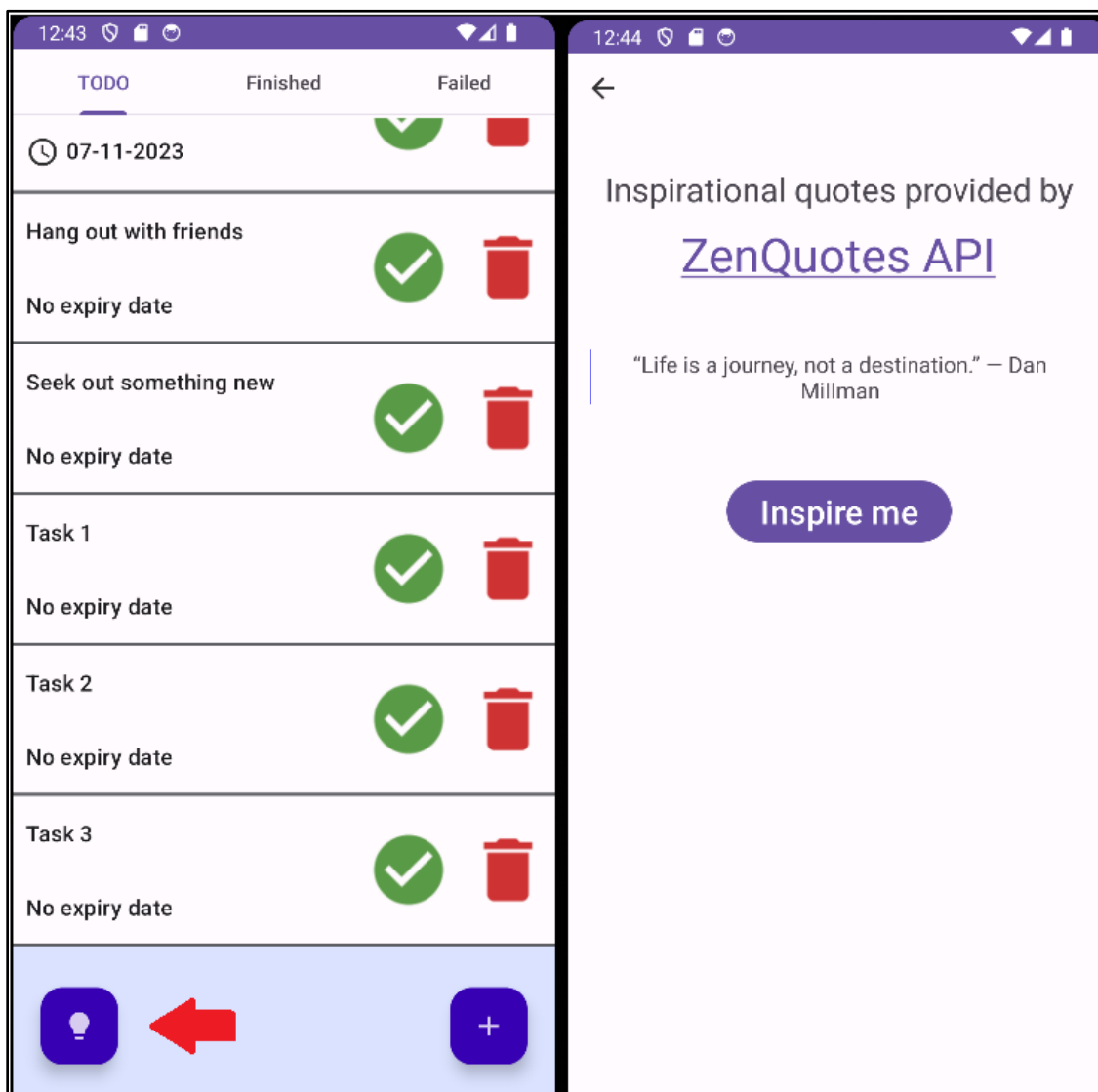
Всяка задача може да бъде изтрита, независимо от нейното състояние. Това става чрез натискане на червената кофа за боклук срещу задачата. Потребителят трябва да даде потвърждение, за да може изтриването да се случи.



Фигура 10 Изтриване на задача

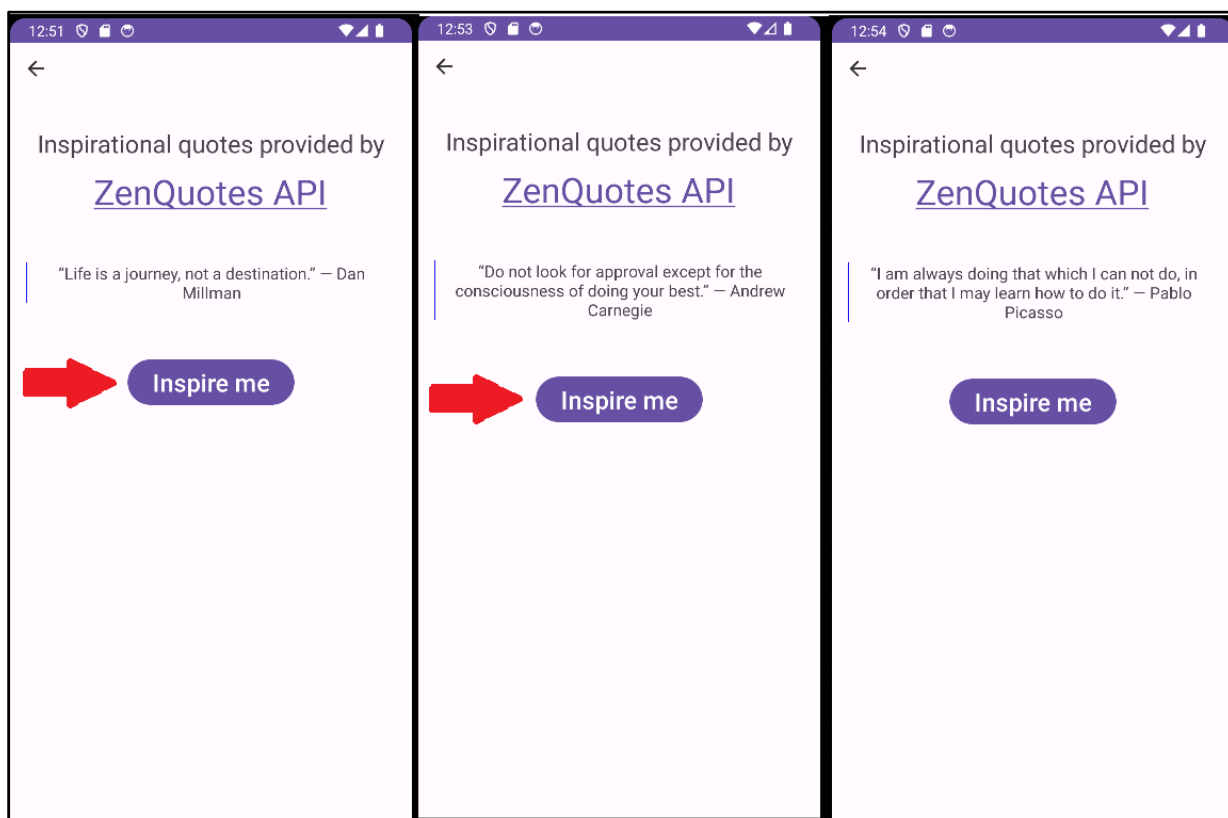
5.7 Извличане на мотивиращи цитати

Приложението дава възможност за извличане на цитати благодарения на Zen Quotes API [1]. Потребителят може да отвори изгледа с цитати чрез натискане на бутона с крушка, която се намира в долния ляв ъгъл на началния екран. След отваряне, изгледът ще покаже произволен цитат.



Фигура 11 Отваряне на екрана за мотивиращи цитати

След отваряне на екрана за цитати, потребителят може да получи нов произволен цитат при натискане на бутона „Inspire me“.



Фигура 12 Получаване на мотивиращи цитати

6. Имплементация

6.1 Извличане на задачи от базата данни

Отдолу е представен метода за извличане на задачи от базата данни. Той се извиква с параметър, който филтрира задачите според тяхното състояние. В зависимост от състоянието на извлечените задачи, се прилага подходяща сортировка за удобство на потребителя. Филтрирането и сортировката се извършват в базата данни за по-добра производителност. Накрая задачите се връщат като списък и това улеснява тяхната обработка.

```
@SuppressWarnings("Range")
public List<Task> selectTasks(@Nullable TaskStatus statusFilter) {
    //Default select query when no status filter is specified
    String selectQuery = "SELECT * FROM TASK ORDER BY EXPIRES_ON";

    if(statusFilter != null) {
        //Statuses correspond to TaskStatus enum values
        if (statusFilter.equals(TaskStatus.TODO)) {
            selectQuery = "SELECT * FROM " +
                "(SELECT * from 'TASK' " +
                "WHERE STATUS = 0 AND EXPIRES_ON >= unixepoch('now', 'start of day') ORDER BY EXPIRES_ON) " +
                "UNION ALL " +
                "SELECT * from 'TASK' WHERE STATUS = 0 AND EXPIRES_ON IS NULL";
        } else if (statusFilter.equals(TaskStatus.Finished)) {
            selectQuery = "SELECT * FROM 'TASK' WHERE STATUS = 1 ORDER BY COMPLETED_ON DESC";
        } else if (statusFilter.equals(TaskStatus.Failed)) {
            selectQuery = "SELECT * FROM 'TASK' " +
                "WHERE STATUS = 2 OR (EXPIRES_ON < unixepoch('now', 'start of day') AND STATUS = 0) ORDER BY EXPIRES_ON";
        }
    }

    Cursor c = _db.rawQuery(selectQuery, selectionArgs: null);
    List<Task> tasks = new ArrayList<>();

    while (c.moveToNext()) {
        Task currentTask = new Task();
        currentTask.setId(c.getInt(c.getColumnIndex(⌘ "ID")));
        currentTask.setTitle(c.getString(c.getColumnIndex(⌘ "TITLE")));
        currentTask.setDescription(c.getString(c.getColumnIndex(⌘ "DESCRIPTION")));
        currentTask.setExpiresOn(
            getLocalDateFromEpochSeconds(c.getLong(c.getColumnIndex(⌘ "EXPIRES_ON"))));
        currentTask.setCompletedOn(
            getLocalDateTimeFromEpochSeconds(c.getLong(c.getColumnIndex(⌘ "COMPLETED_ON"))));
        currentTask.setStatus(TaskStatus.getStatus(
            c.getInt(c.getColumnIndex(⌘ "STATUS"))));

        tasks.add(currentTask);
    }
    c.close();

    return tasks;
}
```

Фигура 13 Метод за извличане на задачи от базата данни

6.2 Десериализация на JSON обект до обект по избор

Когато ключовите стойности на извлечен JSON обект не са със смислени имена или се нуждаем само от някои негови стойности, можем да направим специализиран клас за десериализация.

```
public class QuoteJsonDeserializer implements JsonDeserializer<Quote> {  
    //https://docs.zenquotes.io/zenquotes-documentation/#response  
    @Override  
    public Quote deserialize(JsonElement json, Type typeOfT, JsonDeserializationContext context) throws JsonParseException {  
        JsonObject jsonObject = json.getAsJsonObject();  
        String quote = jsonObject.get("q").getAsString();  
        String author = jsonObject.get("a").getAsString();  
        String preformattedQuoteHTML = jsonObject.get("h").getAsString();  
  
        return new Quote(quote, author, preformattedQuoteHTML);  
    }  
}
```

Фигура 14 Специализиран клас за десериализация на JSON обекти

Този клас може да бъде комбиниран с Gson Converter за автоматично конвертиране до желания обект, след като данните са извлечени с помощта на Retrofit [11] [22] [23].

```
/**
 * Builds and returns a custom GSON converter factory for quotes.
 * @return Custom GSON converter factory for quotes
 */
1 usage  2 omin29
static GsonConverterFactory getGsonConverterFactory(){
    GsonBuilder gsonBuilder = new GsonBuilder();
    gsonBuilder.registerTypeAdapter(Quote.class, new QuoteJsonDeserializer());
    Gson gson = gsonBuilder.create();
    return GsonConverterFactory.create(gson);
}

/**
 * Prepares a Retrofit instance for interaction with the Zenquotes API.
 * @return Retrofit instance
 */
1 usage  2 omin29
static Retrofit getRetrofitInstance(){
    OkHttpClient client = new OkHttpClient().newBuilder().build();
    return new Retrofit.Builder()
        .baseUrl("https://zenquotes.io")
        .addConverterFactory(QuoteAPI.getGsonConverterFactory())
        .client(client)
        .build();
}
```

Фигура 15 Създаване на специализиран Gson Converter и неговата употреба

6.3 Извличане на цитати с Retrofit и тяхното визуализиране

Цитатите се извличат от Zen Quotes API с помощна на Retrofit [1] [11]. Ако има извлечени цитати, тогава се взема един от списъка с цитати и се визуализира. Когато няма цитати, се прави заявка към API-то, за да се извлекат 50 произволни цитата. Заявката е синхронна, но тя се прави в отделна нишка, за да не се блокира главната нишка, която е отговорна за функционирането на потребителския интерфейс. След успешно извличане на цитатите, те се зареждат в списъка, предвиден за тях. Накрая се визуализира цитат от списъка, но този път трябва изрично да се укаже, че помощният метод трябва да се изпълни в главната нишка, защото извикването се извършва от друга нишка.

```
public void getQuoteHandler(View v) {
    if(QuoteAPI.quotes.size() == 0) {
        Thread quoteFetchThread = new Thread()->{
            try {
                Retrofit retrofit = QuoteAPI.getRetrofitInstance();
                QuoteAPI api = retrofit.create(QuoteAPI.class);
                Call<List<Quote>> testCall = api.getQuotes();
                Response<List<Quote>> response = testCall.execute();

                if(response.body() != null) {
                    QuoteAPI.quotes.addAll((List<Quote>)response.body());
                    //This is run on the main thread which is allowed to modify the views
                    runOnUiThread(this::showQuote);
                }
            }
            catch (Exception e) {
                runOnUiThread()-> Toast.makeText(getApplicationContext(),
                    "There was a problem while fetching a quote. Please try ...",
                    Toast.LENGTH_LONG).show();
            }
        });
        quoteFetchThread.start();
    }
    else {
        showQuote();
    }
}
```

Фигура 16 Метод за извличане и визуализиране на цитати

```

private void showQuote() {
    if(QuoteAPI.quotes.size() > 0) {
        quoteTextView.setText(Html.fromHtml(QuoteAPI.quotes.remove(0).getPreformattedQuoteHTML(),
            Html.FROM_HTML_MODE_COMPACT));
        quoteTextView.setVisibility(View.VISIBLE);
    }
}

```

Фигура 17 Помощен метод за визуализация на цитат

6.4 Допълнителна информация

Допълнителна информация за имплементацията на функционалностите на приложението може да намерите в това GitHub хранилище [24].

7. Заключение

Тази курсова работа ми помогна да надградя моите умения за разработване на мобилни приложения. Крайният резултат беше задоволителен и оправда моите очаквания. Приложението наистина може да помогне на потребителя да бъде по-организиран и го зарежда положително. Въпреки това, все още има място за подобрения. В текущия си вид, потребителският интерфейс е статичен и не се съобразява с размерите на различните мобилни устройства. Това може да се промени, за да може всички потребители да имат еднакво качествено изживяване. Друг начин за подобряване на приложението би било увеличаването на начините за мотивация на потребителя. Някои потребители биха предпочели да се мотивират с цитати, други с изображения, трети с музика. Това може да се постигне чрез интеграцията на подходящи API-та.

8. Библиография

- [1] „Zen Quotes API,” [Онлайн]. Available: <https://zenquotes.io/>.
- [2] „JSON,” [Онлайн]. Available: <https://www.json.org/json-en.html>. [Отваряно на октомври 2023].
- [3] „Java Enums,” [Онлайн]. Available: https://www.w3schools.com/java/java_enums.asp. [Отваряно на октомври 2023].
- [4] „FragmentManager,” [Онлайн]. Available: <https://developer.android.com/reference/androidx/viewpager2/adapter/FragmentManager>. [Отваряно на октомври 2023].
- [5] „Activity,” [Онлайн]. Available: <https://developer.android.com/reference/android/app/Activity>. [Отваряно на октомври 2023].
- [6] „Download Android Studio & App Tools - Android Developers,” [Онлайн]. Available: <https://developer.android.com/studio>. [Отваряно на октомври 2023].
- [7] „What is Java and why do I need it?,” [Онлайн]. Available: https://www.java.com/en/download/help/whatis_java.html. [Отваряно на октомври 2023].
- [8] M. Rouse, „What is Android SDK? - Definition from Techopedia,” 6 октомври 2020. [Онлайн]. Available: <https://www.techopedia.com/definition/4220/android-sdk>. [Отваряно на октомври 2023].
- [9] „Gson - Overview,” [Онлайн]. Available: https://www.tutorialspoint.com/gson/gson_overview.htm. [Отваряно на октомври 2023].
- [10] „Maven Repository: com.squareup.retrofit2 >> converter-gson 2.9.0,” [Онлайн]. Available: <https://mvnrepository.com/artifact/com.squareup.retrofit2/converter-gson/2.9.0>. [Отваряно на октомври 2023].
- [11] Jintin, „Retrofit 2,” 3 декември 2017. [Онлайн]. Available: <https://medium.com/@jintin/retrofit-2-b0c80d5caadf>. [Отваряно на октомври 2023].
- [12] „Maven Repository: com.squareup.retrofit2 >> retrofit 2.9.0,” [Онлайн]. Available: <https://mvnrepository.com/artifact/com.squareup.retrofit2/retrofit/2.9.0>. [Отваряно на октомври 2023].
- [13] „SQLite Home Page,” [Онлайн]. Available: <https://www.sqlite.org/index.html>. [Отваряно на октомври 2023].
- [14] „SQLiteOpenHelper,” [Онлайн]. Available: <https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper>. [Отваряно на октомври 2023].

- [15] „ViewPager2,“ [Онлайн]. Available: <https://developer.android.com/reference/androidx/viewpager2/widget/ViewPager2>. [Отваряно на октомври 2023].
- [16] „Fragments,“ [Онлайн]. Available: <https://developer.android.com/guide/fragments>. [Отваряно на октомври 2023].
- [17] „RecyclerView,“ [Онлайн]. Available: <https://developer.android.com/reference/androidx/recyclerview/widget/RecyclerView>. [Отваряно на октомври 2023].
- [18] „DatePickerDialog,“ [Онлайн]. Available: <https://developer.android.com/reference/android/app/DatePickerDialog>. [Отваряно на октомври 2023].
- [19] Sanjeev, „Date picker in Android,“ [Онлайн]. Available: <https://stackoverflow.com/questions/3299392/date-picker-in-android>. [Отваряно на октомври 2023].
- [20] „What Is GitHub? A Beginner’s Introduction to GitHub,“ 18 октомври 2023. [Онлайн]. Available: <https://kinsta.com/knowledgebase/what-is-github/>. [Отваряно на октомври 2023].
- [21] „Let’s build from here,“ [Онлайн]. Available: <https://github.com/>. [Отваряно на октомври 2023].
- [22] J. Wharton, „Gson Converter,“ [Онлайн]. Available: <https://github.com/square/retrofit/tree/master/retrofit-converters/gson>. [Отваряно на октомври 2023].
- [23] E. Paraschiv, „Gson Deserialization Cookbook,“ 24 юни 2022. [Онлайн]. Available: <https://www.baeldung.com/gson-deserialization-guide#deserialize-json-with-non-matching-field-names-to-object>. [Отваряно на октомври 2023].
- [24] О. Солак, „Mobile-Project,“ октомври 2023. [Онлайн]. Available: <https://github.com/omin29/Mobile-Project.git>. [Отваряно на октомври 2023].

9. Списък на фигурите

Фигура 1 Дървовидна структура на проекта	4
Фигура 2 Главният изглед с различните списъци със задачи	6
Фигура 3 Отваряне на изгледа за създаване на задача	7
Фигура 4 Попълване на данни за задачата	8
Фигура 5 Създаване на задача	9
Фигура 6 Редактиране на задача.....	10
Фигура 7 Резултат от редакция на задача	11
Фигура 8 Маркиране на задача като приключена.....	12
Фигура 9 Маркиране на задача като незавършена	13
Фигура 10 Изтриване на задача	14
Фигура 11 Отваряне на екрана за мотивиращи цитати	15
Фигура 12 Получаване на мотивиращи цитати.....	16
Фигура 13 Метод за извличане на задачи от базата данни.....	17
Фигура 14 Специализиран клас за десериализация на JSON обекти	18
Фигура 15 Създаване на специализиран Gson Converter и неговата употреба	19
Фигура 16 Метод за извличане и визуализиране на цитати	20
Фигура 17 Помощен метод за визуализация на цитат	21