

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

**«Московский Авиационный Институт»
(Национальный Исследовательский Университет)**

**Факультет №8 «Компьютерные науки и прикладная математика»
Кафедра 805 «Прикладная математика»**

**Реферат
по курсу «Вычислительные системы»
2 семестр**

Тема:

Разработка мобильного приложения «Список дел» для Android.

Автор работы:

студент 1 курса, гр. М8О-103Б-21

Ершов Д.В.

Руководитель проекта:

Севастьянов В.С.

Дата сдачи:

Москва 2022 г

Содержание:

Введение.....	Error! Bookmark not defined.
Структура приложения.....	4
Используемые технологии.	7
База данных.	8
Вывод.....	9

Введение.

Конечный результат проекта – это рабочее мобильное приложение, работающее на платформе Android. Финальная версия содержит следующие возможности:

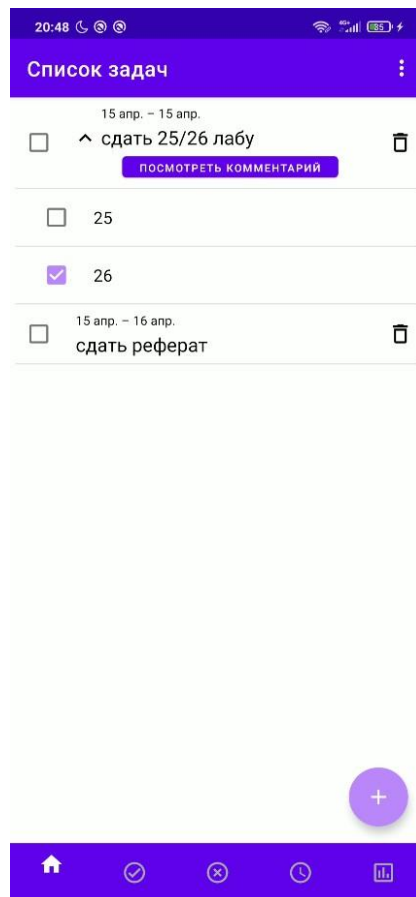
- Добавление задачи
 1. Добавление подзадач
 2. Добавление комментария
 3. Выбор время начала и конца выполнения задания (задание может быть запланировано на будущее)
- Выполнение задачи(задача будет перенесена в раздел «выполненные»)
- Проваливание задачи(если до конца установленного срока не выполнить задачу, то она будет перенесена в раздел «проваленные»)
- Редактирование задачи(редактирование доступно только для задач, находящихся в разделе «запланированные»)
- Получение уведомлений о новых задачах(в настройках можно настроить push уведомления о новых задачах)
- Просмотр статистики по выполнению/невыполнению задач за временные промежутки 7/14/30/90 дней
- Удаление статистики о выполнении/невыполнении задач

Структура приложения

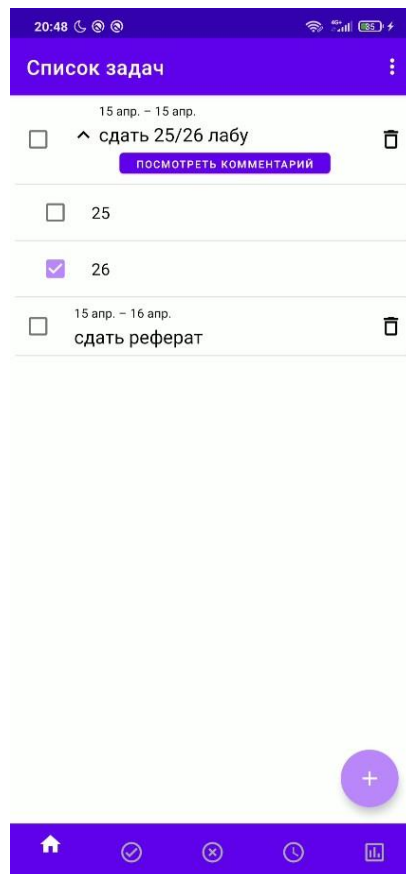
Еще до открытия приложения можно определить, качественно оно сделано или нет. Это можно понять, по наличию/отсутствию “SplashScreen”. SplashScreen это фоновое изображение, которое отображается на дисплее телефона в то время, когда приложение загружается. Если SplashScreen не реализован в приложение, то во время загрузки дисплей будет светиться белым цветом. Как выглядит SplashScreen в моём приложении:



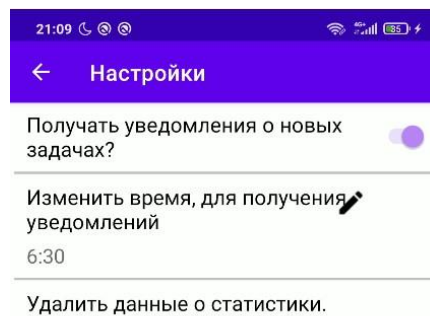
После открытия приложения открывается главная страница, которая содержит бар в нижней части экрана, на котором находятся кнопки для перехода в другие разделы приложения. Также в правом нижнем углу, чуть выше бара, расположена кнопка добавления задач, она будет оставаться доступной в любом разделе приложения. В верхнем баре располагается название приложения и справа от него кнопка, перехода в раздел настроек или просмотра информации об авторе. В центре экрана находятся активные задачи. (Иллюстрация находится на следующей странице.)



Раздел с добавлением задачи содержит несколько полей: поле ввода задачи, поле ввода комментария к задаче, поле выбора даты, поле добавления шагов к задаче. Обязательными для заполнения являются только 2 из 4 полей, а именно: поле названия задачи и поле выбора даты. Дата по умолчанию выбирается в формате «начать сегодня – закончить сегодня». Нажав на поле с датой, его можно будет поменять, выбрав дату на графическом представлении календаря или же введя дату в строку. Что касается строки названия задачи, то если оставить его пустым, то задача не будет добавлена. Так же стоит отметить, что при вводе первого шага, поле для ввода второго будет добавлено автоматически, при вводе второго шага, поле для добавления третьего шага также будет добавлено автоматически и так далее... Кроме того шаги можно удалять. Для добавления задачи необходимо нажать на плюсики в нижнем правом углу. (Иллюстрация находится на следующей странице)



Раздел настроек содержит три поля: первое отвечает за включение/отключение уведомлений, второе за настройку время получения уведомлений и третье за удаление собранной статистики.



Последним значимым разделом в приложении является раздел со статистикой о выполненных/проваленных задачах. Он содержит в своей верхней части выпадающий список, в котором можно выбрать промежуток времени, за который отображать статистику, по умолчанию стоит период 7 дней. Ниже расположена диаграмма, демонстрирующая отношение выполненных и проваленных задач. Ниже самой диаграммы находится график, так же отображающий отношение выполненных и проваленных

задач за указанный период по дням, Ось X – дни, ось Y – кол-во выполненных задач.



Используемые технологии.

В качестве языка программирования для нативной разработки под Android выбор стоял между двумя языками – это Java и Kotlin. Я выбрал Java, так как этот язык более старый, в то время, как Kotlin довольно молодой. Я вижу плюс в Java в том, что язык имеет большую популярность, на нём написано много приложений, написано много обучающих статей про разработку Android приложений на Java, у Java активное комьюнити, благодаря чему можно всегда получить ответ на форуме.

На роле среды разработки была использована AndroidStudio, которую можно назвать монополистом среди сред разработки для Android приложений.

Для разметки графических элементов на экране использовался xml.

Для хранения информации я использовал базу данных, а именно SQLite.

Также в процессе разработки очень активно использовал Android SDK и изучал особенности различных версий Android, т.к. приложение,

написанное для 10 Android не факт, что будет работать на 8 Android, и в обратную сторону тоже.

База данных.

Как уже было упомянуто, я использовал SQLite. База данных нужна в моём приложении для того, чтобы хранить данные после закрытия приложения, т.к. если приложение закрыть то все переменные понятное дело сотрутся из памяти. В моей базе данных находится несколько таблиц: таблица с активными задачами, таблица с проваленными задачами, таблица с запланированными задачами, таблица с выполненными задачами, таблица с подзадачами для задач из всех таблиц, таблица для данных о уведомлениях, таблица для статистики по выполненным/проваленным задачам.

Код создания таблиц:

```
@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL("create table " + TABLE_ACTIVE + "(" + COLUMN_ID + " Long primary key," +
        COLUMN_TASK + " text," + COLUMN_COMMENT + " text," + COLUMN_DATE_STRING + " text," +
        COLUMN_DATE_START + " long," + COLUMN_DATE_FINISH + " long" + ")");

    db.execSQL("create table " + TABLE_COMPLETED + "(" + COLUMN_ID + " Long primary key," +
        COLUMN_TASK + " text," + COLUMN_COMMENT + " text," + COLUMN_DATE_STRING + " text," +
        COLUMN_DATE_START + " long," + COLUMN_DATE_FINISH + " long" + ")");

    db.execSQL("create table " + TABLE_FAILED + "(" + COLUMN_ID + " Long primary key," +
        COLUMN_TASK + " text," + COLUMN_COMMENT + " text," + COLUMN_DATE_STRING + " text," +
        COLUMN_DATE_START + " long," + COLUMN_DATE_FINISH + " long" + ")");

    db.execSQL("create table " + TABLE_PLANNED + "(" + COLUMN_ID + " Long primary key," +
        COLUMN_TASK + " text," + COLUMN_COMMENT + " text," + COLUMN_DATE_STRING + " text," +
        COLUMN_DATE_START + " long," + COLUMN_DATE_FINISH + " long" + ")");

    db.execSQL("create table " + TABLE_SUBTASKS + "(" + COLUMN_ID + " Long primary key," +
        COLUMN_SUBTASKS + " text," + COLUMN_CHECKED + " boolean," +
        COLUMN_KEY + " text" + ")"); // COLUMN_KEY = PARENT TASK NAME

    db.execSQL("create table " + TABLE_ALARM_STATE + "(" + COLUMN_STATE + " int," + COLUMN_TIME_HOURS
        + " int," + COLUMN_TIME_MINUTES + " int" + ")");

    db.execSQL("create table " + TABLE_STATS + "(" + COLUMN_STATS_DATE + " long," + COLUMN_COM_OR_FAIL
        + " int" + ")");
}
```


Вывод.

Результатом работы стало мобильное приложение для Android, в котором можно фиксировать свои задачи и отслеживать их выполнение/невыполнение.

В процессе работы узнал очень много нового: Java(на очень базовом уровне, только синтаксис, ну еще знаю, что она память сама чистит, в отличие от Си), Android SDK для работы с Android, XML для разметки графической составляющей, основы работы с реляционными базами данных(SQLite).

Результат получился очень хорошим, хоть и не получилось сделать нормально работающие уведомления. Но и без них приложение классное.