

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ПЕТРА ВЕЛИКОГО

---

Институт компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

**Отчет**  
по курсовой работе «Разработка приложения Reminder»

Дисциплина  
«Проектирование мобильных приложений»

Выполнила:  
студентка группы 33531/3  
Кузовкина Е.О.

Преподаватель:  
Алексюк А.О.

Санкт-Петербург  
2018

## Содержание

Введение .....	3
Цели .....	3
Задачи .....	3
1.    Стартовый экран .....	3
2.    Главный экран .....	3
2.1.    Фрагменты отображения всех заметок.....	4
3.    Экран добавления нового напоминания .....	6
4.    Хранение данных .....	8
5.    Анализатор кода.....	9
6.    Тестирование .....	9
Заключение.....	9
Список литературы.....	10
Приложение 1.....	10

## Введение

Разработка под платформу Android — это интересный и трудоемкий процесс. Перед разработкой программного продукта необходимо понимать цель, задачи и ресурсы, которые требуются для реализации данного продукта. Необходимо принимать взвешенные решения начиная от выбора шаблона проектирования до распределения всех задач по времени, в целях более оптимизированной разработки проекта. Вся разработка происходила с использованием среды Android Studio. Код был написан на языках Java и Kotlin (привязка базы данных была реализована с помощью языка Kotlin).

## Цели

Целью работы является разработка Android приложения-напоминалки, которое может хранить заметки различных типов, напоминать о них и многое другое.

## Задачи

Необходимо реализовать:

- Стартовый экран;
- Главный экран, имеющий возможность отображать данные приложения в необходимом виде (в виде двух списков);
- Экран добавления записей, позволяющий выбирать дату и время, а также устанавливать приоритет;
- Базу данных для хранения информации пользователя, и взаимодействие этой базы с приложением;
- Провести тестирование;

## 1. Стартовый экран

Стартовый экран приложения реализован в виде всплывающей картинки, для ознакомления пользователя с тематикой приложения. При необходимости данный Splash Screen можно отключать и при последующих запусках приложения он не будет больше отображаться. Splash Screen является первым фрагментом главного экрана.

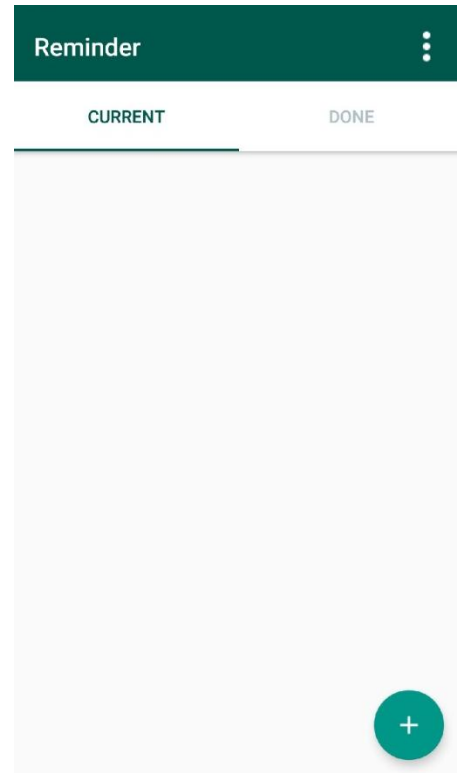
## 2. Главный экран

Реализация главного экрана сделана на основе одного Activity и трех Fragment'ов. Экран состоит из AppBarLayout'a, в котором находятся Toolbar,

где размещено название самого приложения, а также TabLayout для отображения списков текущих и выполненных заметок. Прокручивая вправо-влево можно просматривать списки Current Tasks и Done Tasks. Списки размещены в элементе ViewPager.



*Рисунок 1 Стартовый экран*



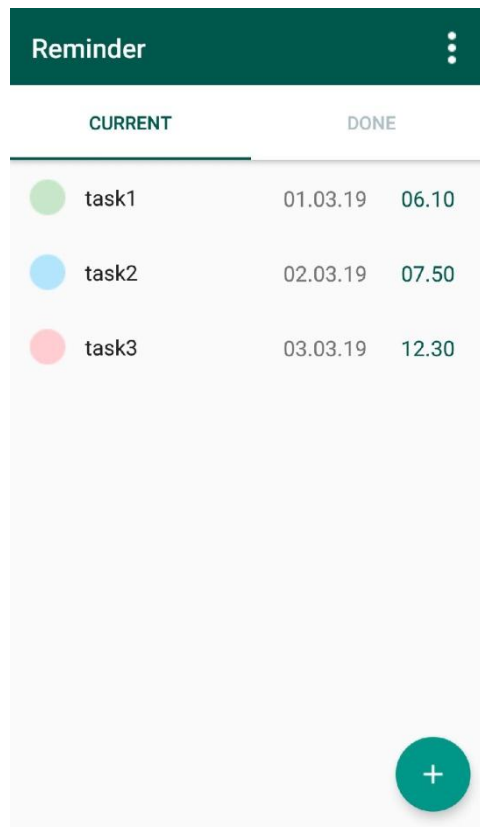
*Рисунок 2 Главный экран*

Такой тип меню был выбран из-за его эргономичности – пользователю всегда удобно дотянуться до кнопки добавления новой заметки или пролистнуть вправо или влево список напоминаний.

На рисунке 2 видно кнопку `FloatingActionButton`, которая позволяет перейти на экран создания напоминания.

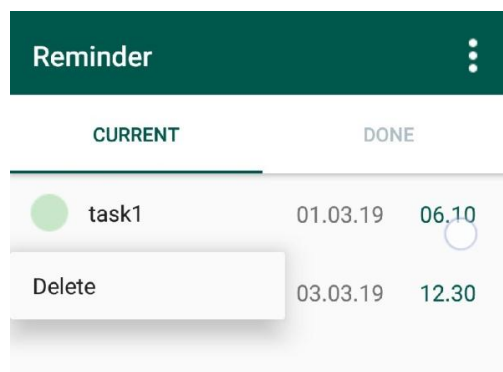
### **2.1.      Фрагменты отображения всех заметок**

Для того чтобы можно было отобразить все записи использовался элемент `RecyclerView` – представляющий абстракцию списка, который позволяет работать с большим количеством данных и отображать их на экране. Его особенность от своего предшественника `ListView` в том, что `RecyclerView` переиспользует графическое представление, а не создает отдельное – для каждого.



*Рисунок 3 Список текущих напоминаний*

На рисунке видно некоторое количество записей и заголовков для этого раздела, который не является элементом списка. Заметки как на этом фрагменте, так и в другом можно удалять долгим нажатием на нужную заметку.



*Рисунок 4 Удаление элемента списка*

Фрагмент сделанных заметок выполнен аналогично. Таким образом, тремя фрагментами главного экрана являются – стартовый экран, список текущих заметок и список выполненных заметок.

Так же пользователь может спокойно перемещать заметки между списками. При нажатии на индикатор приоритета заметки – заметка переместится в другой список, это позволяет легко перемещать напоминания и заметки между списками, если вдруг пользователь случайно пометил их как выполненные.

### 3. Экран добавления нового напоминания

При нажатии на `FloatingActionButton` всплывает новое окно. В данном окне пользователь указывает название или тему заметки, она является обязательным атрибутом для создания новой записи в листе заметок. Также по желанию пользователь может указать дату и время, когда нужно напомнить о каком-либо событии (например, идет речь о напоминании о празднике). Для этого были использованы такие фрагменты как `DataPickerFragment` и `TimePickerFragment`. Если указано время, то в назначенную минуту в списке уведомлений появится `Notification` от нашего приложения с темой напоминания.

Так же пользователь может указать приоритет заметки. Для удобства чтения было выбрано 3 вида приоритетов: низкий, обычный и высокий. По умолчанию заметка создается с низким приоритетом. Разные типы приоритетов могут помочь пользователю ориентироваться среди списка заметок и выполнять наиболее ценные в первую очередь. Так же данная реализация приоритетов удобна для составления листа покупок, где можно указать какие товары стоит купить в первую очередь.

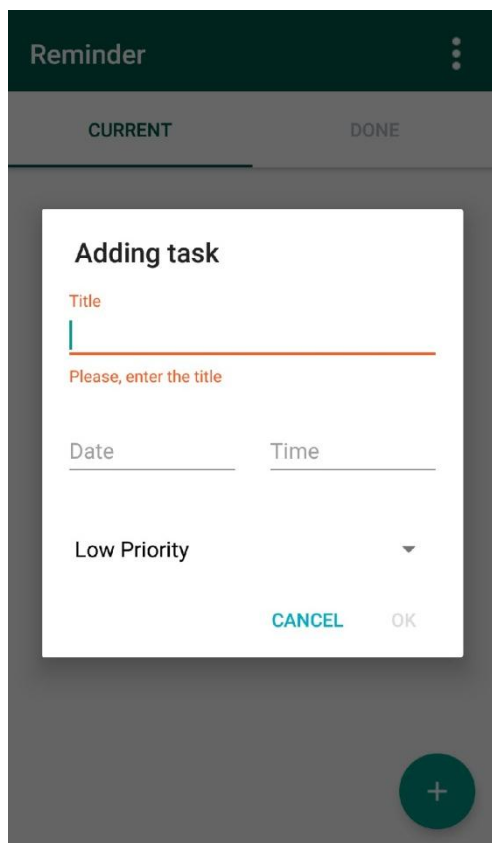


Рисунок 5 Добавление элемента списка

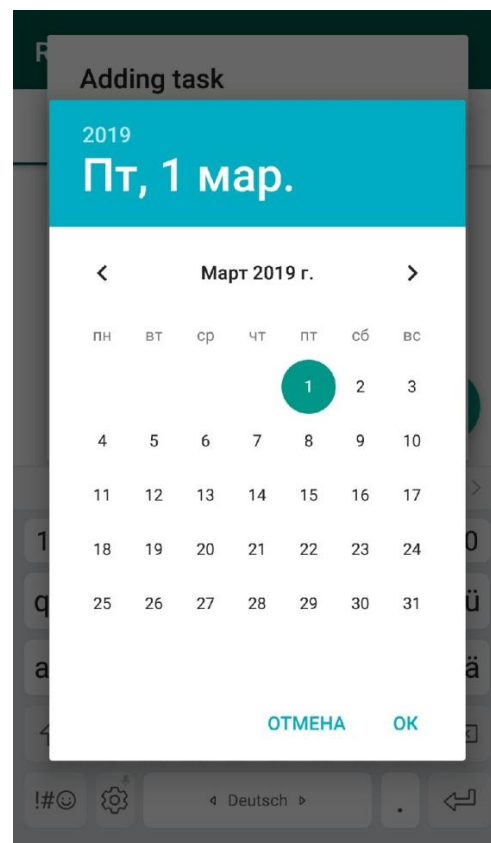


Рисунок 6 Выбор даты

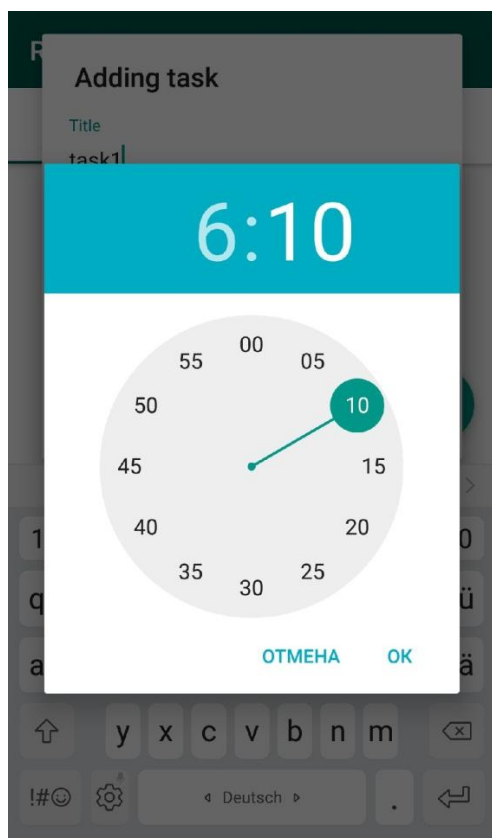


Рисунок 7 Выбор времени

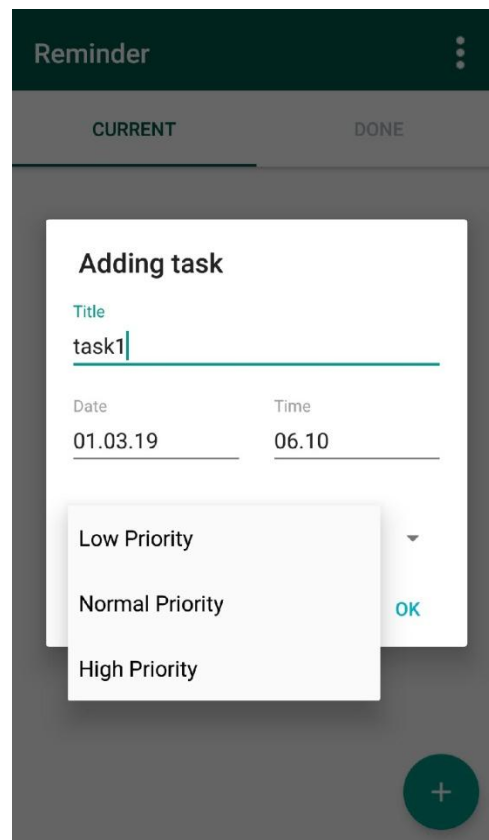
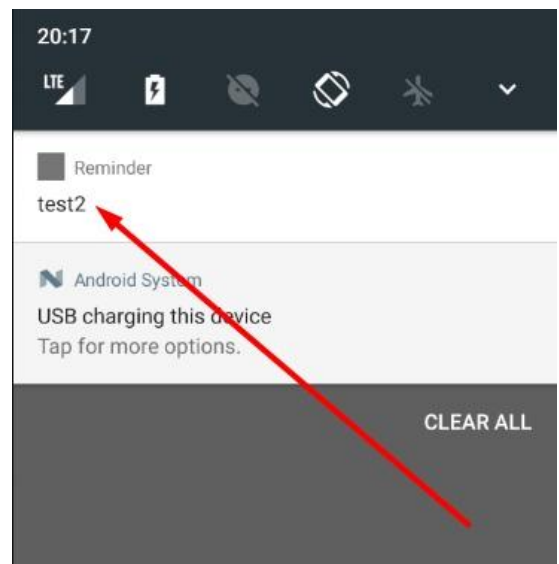
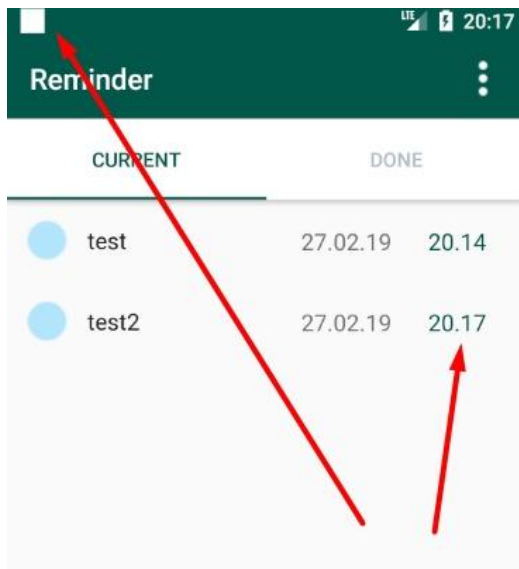


Рисунок 8 Выбор приоритета



Рисунки 9-10 Система оповещений

Для прослушивания таких сообщений был использован `BroadcastReceiver` — это компонент для получения внешних событий и реакции на них. Инициализировать передачи могут другие приложения или службы. Класс `BroadcastReceiver` является базовым для класса, в котором должны происходить получение и обработка сообщений.

Данный слушатель, получив сообщение отправляет пользователю уведомление, которое содержит необходимую для него информацию. Уведомления были реализованы с помощью компонента `Notification` и системного сервиса `NotificationManager`.

## 4. Хранение данных

Для хранения данных использовалась СУБД `SQLite`. База данных хранит в себе две таблицы с информацией, которую пользователь хотел сохранить. Одна таблица хранит основную информацию для записи и имеет следующую структуру:

Приоритет	Название заметки	Дата	Время
-----------	------------------	------	-------

Приоритет – приоритетность заметки

Название – основной текст напоминания (not null)

Дата – заголовок записи

Date – дата напоминания

Время – время напоминания



## 5. Анализатор кода

По завершению основной работы была запущена функция среды разработки Inspect Code, которая показала не малое количество предупреждений, но все они были не критичными (область видимости переменных, орфографические ошибки в строках, не вынесение строк в ресурсы, неиспользуемые переменные). Все предупреждения такого рода были устранены, оставшиеся предупреждения имеют сознательный характер.

## 6. Тестирование

Для создания UI тестов был использован фреймворк Espresso. Была протестирована небольшая часть кода для ознакомления с возможностями Android тестирования.

С созданием Unit тестов возникли проблемы, так как почти все, что стоило бы протестировать, получает на вход классы Android'a, JVM не может с этим работать.

## Заключение

По итогу, было разработано Android приложение-напоминалка Reminder. Во время разработки пришлось столкнуться со множеством проблем, которые замедляли процесс. Но благодаря которым, удалось узнать и изучить новое и интересное. Были изучены новые для меня элементы: RecyclerView, ToolBar и FloatingActionButton. Также была изучена работа с СУБД SQLite.

Вследствие медленной работы, не удалось полностью достигнуть конечной цели и реализовать все поставленные задачи. Например, реализовать изменение уже существующих заметок. Но я считаю, что основную функциональность приложения удалось реализовать, а именно: создание, удаление и сохранение заметок; отображение всей информации о заметке; установка и отображение напоминаний.

Разработка данного приложения позволила мне ознакомиться с основами программирования под Android. Несомненно, разработка мобильных приложений очень увлекательный процесс, так что возможно в будущем я попробую разработать уже более сложное приложение, основываясь на знаниях, полученных в данном курсе.

## Список литературы

1. Android Developers. <https://developer.android.com/> [Электронный ресурс]
2. Освой Android играючи. <http://developer.alexanderklimov.ru/android/> [Электронный ресурс]
3. Основы разработки android-приложений. <https://youtu.be/ac8y518CIng> [Видео ресурс]
4. Multiple selection RecyclerView Android.  
<https://learnpainless.com/android/recyclerview/multiple-selection-recyclerview-android> [Электронный ресурс]
5. Documentation SQLite. <https://sqlite.org/index.html> [Электронный ресурс]
6. Хранение данных. SQLite. <https://startandroid.ru/ru/uroki/vse-uroki-spiskom/74-urok-34-hranenie-dannyh-sqlite.html> [Электронный ресурс]

## Приложение 1

Ссылка на исходный код: <https://github.com/kuzo-liza/Reminder/>