**Отчет**

Тема: Разработка Android-приложений

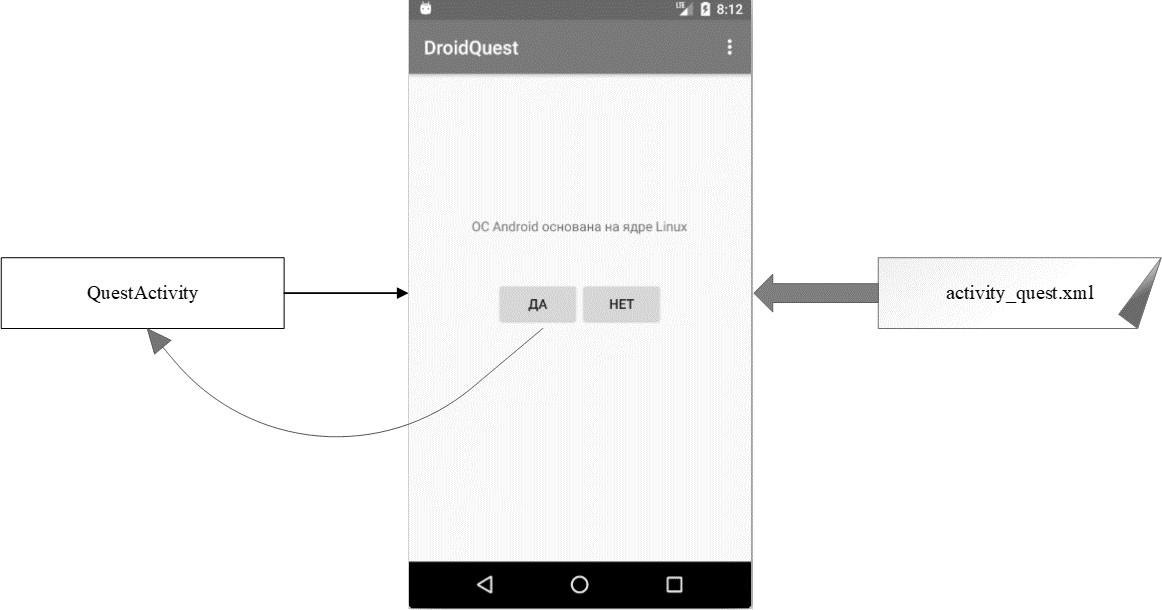
***1.Первое Android-приложение***



Рисунок 1. Итоговое Android-приложение. Тест.

*Основы построения приложения*

Приложение DriodQuest состоит из активности (activity) и макета  
(layout):  
 Активность представлена экземпляром Activity — класса из Android  
SDK. Она отвечает за взаимодействие пользователя с информацией на  
экране.  
Чтобы реализовать функциональность, необходимую приложению,  
разработчик пишет субклассы Activity. В простом приложении бывает  
достаточно одного субкласса; в сложном приложении их может  
потребоваться несколько. DriodQuest — простое приложение, поэтому в нем используется всего  
один субкласс Activity с именем QuestActivity. Класс QuestActivity управляет пользовательским интерфейсом, изображенным на рисунке 1.  
Макет определяет набор объектов пользовательского интерфейса и их  
расположение на экране. Приложение DriodQuest включает файл макета с  
именем activity\_quest.xml. Разметка XML в этом файле определяет  
пользовательский интерфейс, изображенный на рисунке 1.2

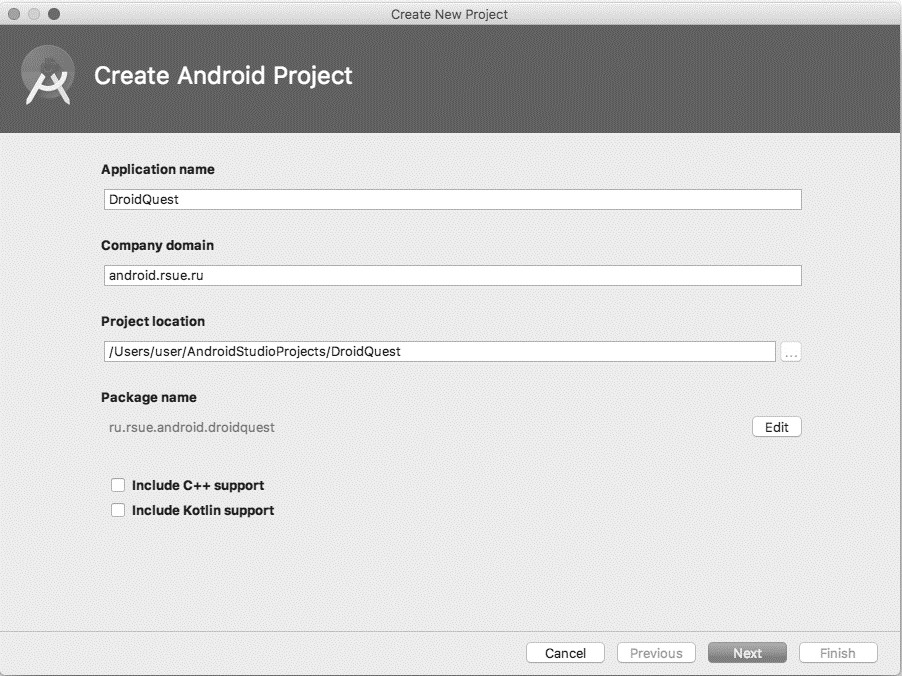
Рисунок 2.2

Первым этапом при создании Android-приложения является создание окружения для будущей работы с ним.

Рисунок 2.3

Здесь (рис. 2.3) даю название моему проекту, выбираю язык-программирования для работы с ним (Java, Kotlin). В моем случае выбор пал на Java. И теперь выбираем минимальную версию Android которая будет поддерживать приложение. Я выбрал версию «5.0 Lolipop».

На следующем этапе я выбрал пустую активность (Blank Activity / Empty Activity).



В интерфейсе QuestActivity будут задействованы пять виджетов: вертикальный виджет LinearLayout; TextView; горизонтальный виджет LinearLayout; две кнопки Button.

Определившись с виджетами я создал две кнопки True и False после чего присвоил им функцию, позволяющую определить правильность вопроса. Перед этим я заранее определил вопросы в string.xml, где также определяются все имеющиеся кнопки и остальные виджеты.

Вопросов было добавлено 5(в будущем их будет уже 10) и каждому вопросу был присвоен собственный ответ.

Следующим этапом была добавлена адаптивность. Она позволяет пользоваться приложением в горизонтальном положении.

Далее к кнопкам «Да» и «Нет» были добавлены кнопки «Далее» и «Назад». Позже они будут изменены в кнопки с иконками в виде стрелок. Они позволяют передвигаться по вопросам не показывая при этом верным или нет оказался эти самые пропущенные вопросы.

Теперь оставалось добавить кнопку «Солгать» или «Соврать» позволяющую показать ответ на выбранный вопрос. При выборе этой кнопке на экране владельца вылезает сообщение «Вы лжец». Указывающую на нечестность выбранного ответа.

***2. Создание нового проекта BookDepository (DeceitActivity)***

Создание нового проекта начинается ровно также, как и создание первого приложения. Данное приложение предназначено для учета прочтенных книг.

В приложении BookDepository пользователь создает запись о книге, которую планирует прочитать, с заголовком, датой и фотографией обложки. Также можно отправить отзыв о книге по электронной почте, опубликовать его в Twitter, Facebook или другом приложении. Сообщая о прочтенных книгах, пользователь мотивирует других больше читать. (рис 2)

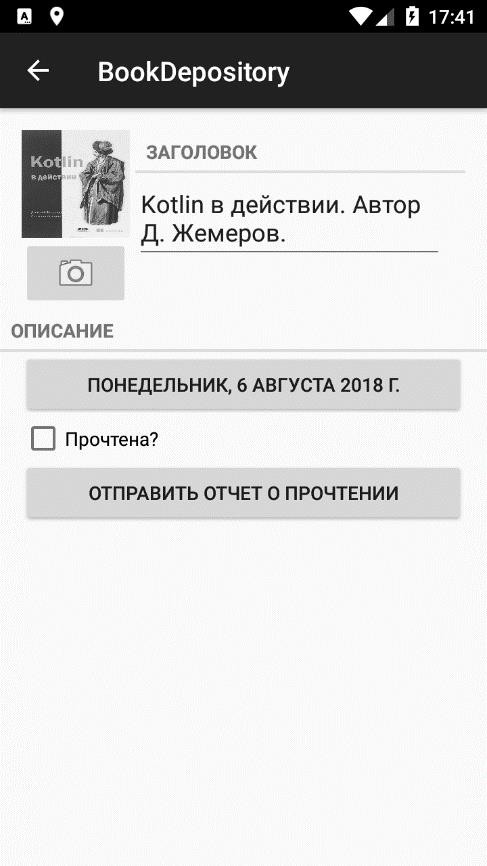
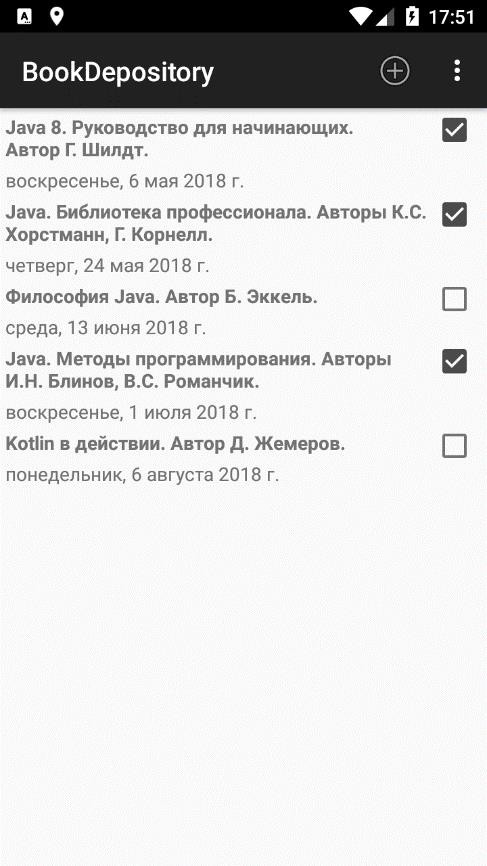


Рисунок 2 – Приложение BookDepository

BookDepository — сложное приложение, в нем используется интерфейс типа «список/детализация»: на главном экране выводится список желаемых к прочтению книг. Пользователь может добавить новую или выбрать существующую книгу для просмотра и редактирования информации.

На первом этапе будет реализовано представление детализации BookDepository.

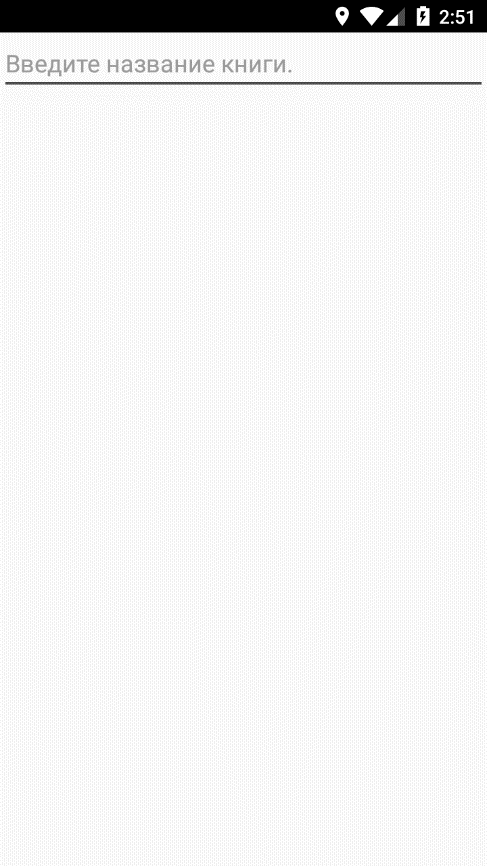


Рисунок 2.2 – Итоговое приложение BookDepository

Экранном, показанным на рисунке 2.2, будет управлять UI-фрагмент с именем BookFragment. Хостом экземпляра BookFragment является активность с именем BookActivity. Хост предоставляет позицию в иерархии представлений, в которой фрагмент может разместить свое представление. Фрагмент не может вывести представление на экран сам по себе. Его представление отображается только при размещении в иерархии активности. Класс BookFragment создает пользовательский интерфейс и управляет им, а также обеспечивает взаимодействие с объектами модели. В проект будут добавлены три класса: Book, BookFragment и BookActivity.

Экземпляр Book представляет одну книгу. В этом разделе описание книги будет состоять только из заголовка и идентификатора. Заголовок содержит текст (например, «Война и мир. Том 1.» или «Горе от ума.»), а идентификатор однозначно идентифицирует экземпляр Book.

Представление BookActivity состоит из элемента FrameLayout, определяющего место, в котором будет отображаться представление BookFragment.

Представление BookFragment будет состоять из элементов LinearLayout и EditText. BookFragment определяет поле для виджета EditText (mTitleField) и назначает для него слушателя, обновляющего уровень модели при изменении текста.

В приложении BookDepository был добавлен CheckBox который и позволит пользователю отметить прочитанную книгу. К дополнению к чекбоксу также была добавлена функция выводящая дату и время отмеченной прочтенной книги.

Как и предыдущему приложению, этому, была добавлена адаптивность под альбомную ориентацию.

Все книги выводятся списком с датой и чекбоксом.

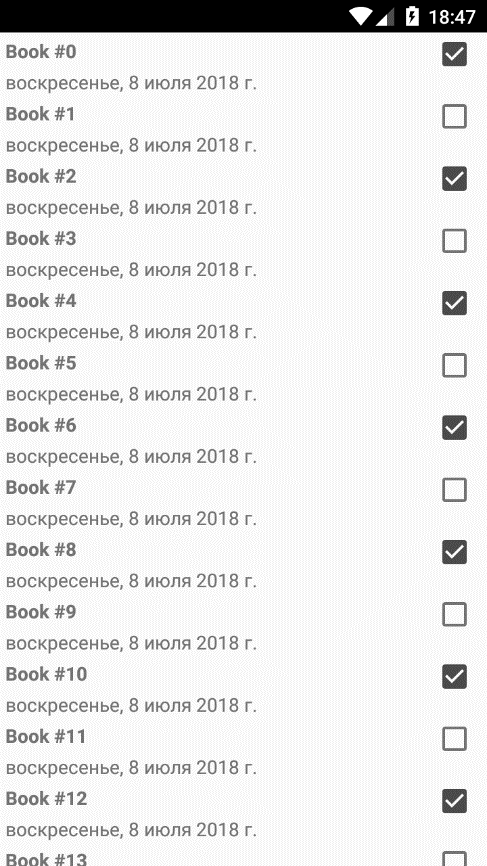


Рисунок 2.3

Книга выбранная в списке – рисунок 2.4

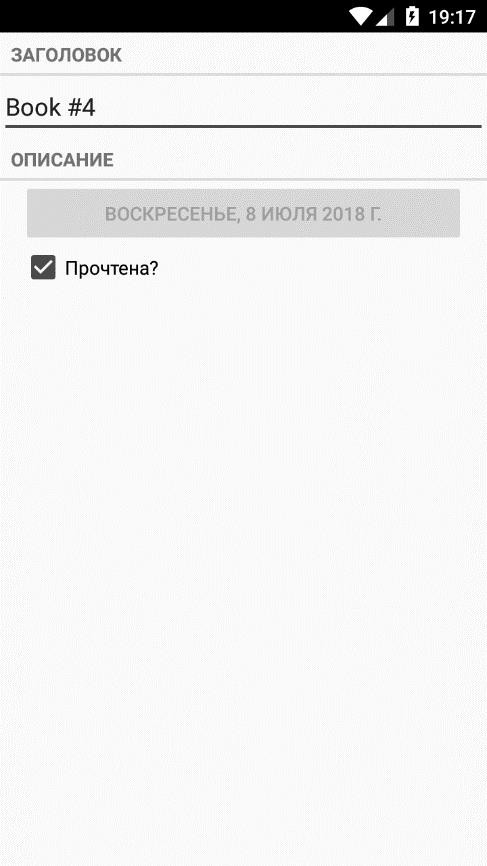


Рисунок 2.4

Также была создана активность ViewPager. Она позволяет листать. Элементы списка, проводя пальцем по экрану.

Далее было создано диалоговое окно позволяющее пользователю самостоятельно выбирать дат прочтения той или иной книги.

Начиная с версии Lollipop, диалоговые окна прошли визуальную переработку. AlertDialog в Lollipop и выше автоматически используют новый стиль. В более ранних версиях Android окна AlertDialog возвращаются к старому стилю. Для того чтобы диалоговые окна всегда отображались в новом стиле независимо от версии Android на устройстве пользователя, необходимо использовать библиотеку **AppCompat**.

Позже были добавлены различные косметические украшения для сайта, вроде иконок вместо текста.

Следующим этапом было добавление базы данных (в моем случае SqLite).

SQLite — реляционная база данных с открытым кодом, как и MySQL или Postgresql. В отличие от других баз данных, SQLite хранит свои данные в простых файлах. Была добавлена таблица books со строками : \_id, uuid, title, date, readed.

Далее была добавлена кнопка «отправить отчет о прочтении» при нажатии на которую пользователь может отправить отчет. При этом активность запускается из другого приложения на устройстве.

Похожа работа была сделана далее с камерой. Работа с которой начинается при нажатии на соответствующую кнопку с соответствующей иконкой.

***Заключение***

При написании данных приложений, я научился понимать код Java и создавать андроид приложения, подключать к ним базу данных и создавать связи с другими приложениями.