Министерство образования Новосибирской области ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ЧТЕНИЯ ЭЛЕКТРОННЫХ ДОКУМЕНТОВ

Пояснительная записка к курсовому проекту

ПМ.01Разработка модулей программного обеспечения для компьютерных систем

МДК 01.03 Разработка мобильных приложений

НАТКиГ.202300.010.000ПЗ

Разработал:

студент группы ПР-20.101

Фаруков В. С.

2023

**Содержание**

[Введение 3](#_Toc138507144)

[1 Исследовательский раздел 4](#_Toc138507145)

[1.1 Описание предметной области 4](#_Toc138507146)

[1.2 Образ клиента 5](#_Toc138507147)

[1.3 Сценарии 5](#_Toc138507148)

[1.4 Сбор и анализ прототипов 6](#_Toc138507149)

[2 Проектирование приложения 12](#_Toc138507150)

[2.1 UI/UX дизайн проекта 12](#_Toc138507151)

[2.2 Выбор технологии, языка и среды программирования 16](#_Toc138507152)

[3 Разработка мобильного приложения 17](#_Toc138507153)

[3.1 Разработка базы данных 17](#_Toc138507154)

[3.2 Разработка используемых плагинов 17](#_Toc138507155)

[3.3 Описание разработанных процедур и функций 18](#_Toc138507156)

[4 Тестирование 25](#_Toc138507157)

[4.1 Протокол тестирования дизайна приложения 25](#_Toc138507158)

[4.2 Протокол тестирования функционала приложения 25](#_Toc138507159)

[Заключение 31](#_Toc138507160)

[Библиография 33](#_Toc138507161)

[Приложение А (обязательное) Техническое задание 34](#_Toc138507162)

Введение

Мобильное приложение для чтения электронных документов позволяет читать электронные книги и другие текстовые документы на мобильных устройствах пользователей. Оно может использоваться как для личного чтения, так и для образовательных и профессиональных целей. Использование такого приложения позволяет пользователю читать и управлять документами так, как ему будет удобно. Это помогает ускорить процесс работы с документами.

Целью курсового проекта является создание мобильного приложения для чтения электронных документов.

Задачами курсового проекта в связи с указанной целью являются:

* изучение предметной области;
* рассмотрение приложения с точки зрения пользователя для выявления необходимых функций приложения;
* создание интерфейса и необходимого функционала приложения;
* тестирование приложения.

Объект исследования – приложение для чтения электронных документов.

Предмет исследования – изучение принципов функционирования и инструментов приложения.

# 1 Исследовательский раздел

## 1.1 Описание предметной области

Человеческое общество ⎯ это единый механизм. Каждый человек в этом механизме выполняет свою роль. В связи с этим человеческая деятельность становится очень разнообразной. Но, несмотря на это, мы можем сказать, что многие люди в современном мире так или иначе сталкиваются с технологиями в своей деятельности. Прежде всего, большинство из них работает с электронными документами в различных форматах. Это может быть связано с документацией, отчетностью и другими обязанностями. Кроме этого, приобщаться к использованию электронных документов начинают еще в школе. Детям могут отправлять задания в электронном формате для ознакомления с учебным материалом.

Помимо всего этого, люди могут проводить свой досуг, связывая его с чтением электронных книг. Для этого они так же могут использовать приложение. Электронные книги все больше набирают свою популярность в интернете. Их выбор становится все больше и разнообразнее. Хоть некоторые предпочитают держать в руках бумажные книги и чувствовать их физически, многие люди выбирают электронные книги.

Выбор электронных книг и документов вместо печатных можно объяснить несколькими причинами. Прежде всего важной характеристикой электронных книг является их компактность и совершенно незначительный вес, когда они находятся внутри гаджета читающего. Портативность стала важнейшим преимуществом книг в цифровых форматах. Теперь не нужно носить с собой несколько тяжелых книг, все необходимые книги могут поместиться в одном устройстве. Это сильно упрощает чтение книг. Человеку становится намного проще и удобнее держать и читать книгу или работать с документами, он меньше отвлекается от процесса чтения. Электронные книги становятся дешевле и доступнее для клиентов. Все потому, что издателю не приходится тратить деньги на материалы, перевозку и хранение книг до их продажи. Отсюда вытекает то, что на производство книг и документов уходит меньше сырья, а именно деревьев. Это безусловно благоприятно влияет на окружающую среду. Кроме этого, читать печатную книгу в темноте невозможно, а при слабом освещении становится некомфортно и вредно. Для электронной книги не нужен дополнительный источник света и ее удобно использовать при любом освещении.

В связи с этим возрастает потребность в приложениях, в которых удобно читать текстовые документы. В таких приложениях можно открывать различные форматы документов, настраивать яркость, менять внешний вид приложения, искать информацию в документах.

## 1.2 Образ клиента

У приложения для чтения документов нет возрастных ограничений. Пользователем может быть любой человек, умеющий читать. Это связано с тем, что работать с электронными документами в наше время необходимо во многих сферах деятельности. Это приходится делать офисным работникам, учителям, бухгалтерам и многим другим. Также работать с электронными документами учат еще в школе. Дети начинают использовать такие документы с раннего возраста.

## 1.3 Сценарии

Никита учится в школе, и ему часто приходится получать задания от учителей в электронном формате. Также Никите необходимо открывать готовые задания в таком виде для проверки. Для этого ему нужно приложение, котором он сможет открывать документы для чтения.

Андрей работает в офисе, и он постоянно работает с документами в электронном формате. Каждый день начальник отправляет Андрею план работы на день. Кроме этого, Андрею постоянно приходится оформлять отчетность по своей работе и отправлять ее начальнику. Иногда Андрею нужно читать готовую отчетность. Чтобы справиться с этими задачами Андрею понадобится приложение для чтения электронных документов.

Степан вышел на пенсию, и теперь у него появилось много свободного времени, которое он может тратить на то, что он любит. Свой досуг Степан предпочитает проводить за чтением книг. Также он иногда любит перечитать уже прочитанные книги. Для этого ему нужно приложение, в котором удобно читать в электронном формате. Поэтому приложение для чтения книг в различных форматах поможет Степану с комфортом проводить свое время.

## 1.4 Сбор и анализ прототипов

Работа с электронными документами очень актуальна в современном мире. Существует множество приложений, предназначенных специально для чтения таких документов. Популярные приложения для чтения книг предлагают пользователям множество функций, упрощающих работу с документами. Они позволяют открывать различные форматы документов, работать со списком документов и группировать документы.

### 1.4.1 Moon+ Reader

Приложение Moon+ Reader отличается широким выбором настроек интерфейса, что позволяет любому пользователю настроить ее под свои нужды и предпочтения. Можно изменить тип и размер шрифта, размер пробелов между словами. Настроить можно даже анимацию перелистывания страниц или включить автоматическую прокрутку. Присутствует несколько вариантов светлой и темной темы. Стоит отметить, что приложение поддерживает большинство существующих форматов, а именно: EPUB, PDF, DJVU, AZW3, MOBI, FB2, PRC, CHM, CBZ, CBR, UMD, DOCX, ODT, RTF, TXT, HTML, MD (MarkDown), WEBP, RAR, ZIP, а также OPDS-каталоги.

На рисунке 1 приведено меню приложения Moon+ Reader.

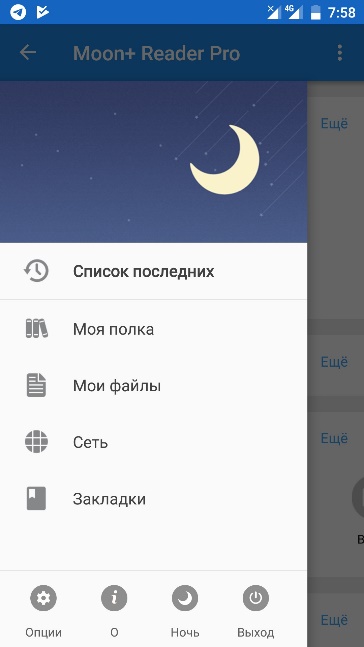


Рисунок 1 ⎯ Меню приложения Moon+ Reader

### 1.4.2 ReadEra

Приложение с удобным интерфейсом, а также с высокой степенью настройки. Можно настроить режим чтения и автоматического листания страниц. Главным отличием ReadEra является поддержка многодокументного режима, что позволяет читать несколько книг одновременно, открыв их в режиме разделенного экрана. А читает приложение довольно много форматов, включая FB2, PDF, EPUB, DOC, DOCX, RTF, DJVU, MOBI, AZW3, TXT, ODT, CHM, RAR и ZIP.

На рисунке 2 приведен внешний вид приложения ReadEra.

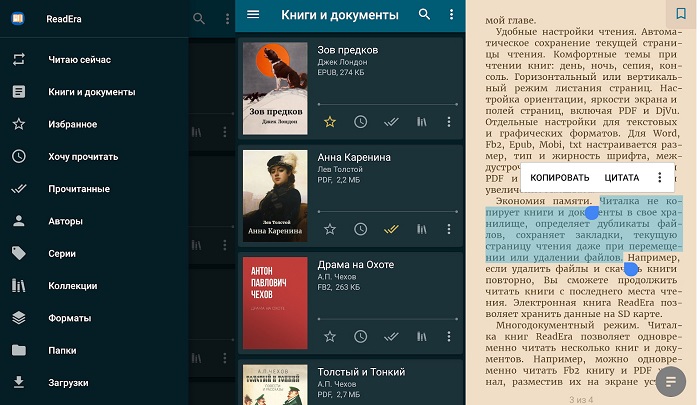


Рисунок 2 ⎯ Внешний вид приложения ReadEra

### 1.4.3 eBoox

Одним из самых стильных приложений для чтения на Android-устройствах является eBoox. Приятный и удобный интерфейс с возможностью выбора светлой и темной темы. Можно включить перелистывание кнопками громкости, автоматическое изменение подсветки. В приложении реализован поиск по тексту, выделение цитат и другие подобные функции. Поддерживаются такие форматы: FB2, EPUB, PDF, DOC, DOCX, MOBI, PRC, TXT, RTF, ODT, HTML, CBR, CBZ, ZIP и RAR.

На рисунке 3 показана библиотека всех книг в приложении eBoox.

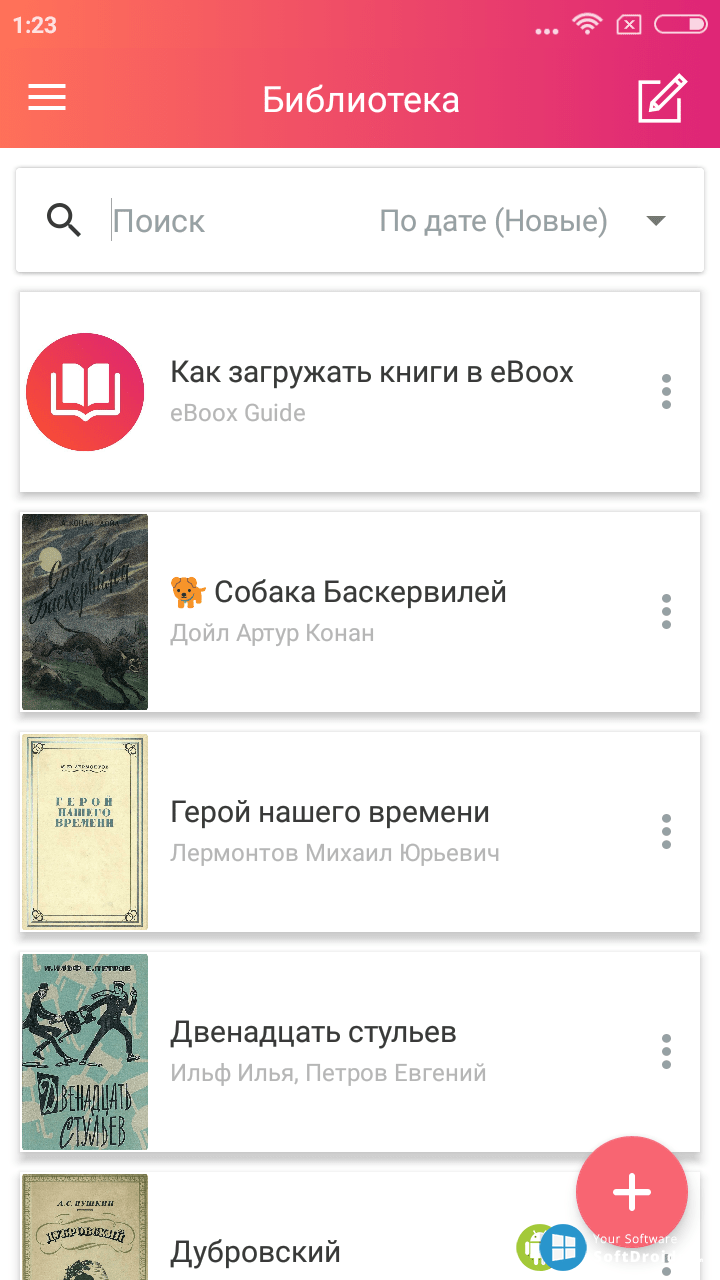


Рисунок 3 ⎯ Библиотека приложения eBoox

### 1.4.4 eReader Prestigo

Приятный минималистичный интерфейс, к тому же довольно удобный. Хорошая книжная полка с возможностью настройки внешнего вида. Также открывается возможность синхронизации между устройствами. eReader Prestigio, как и многие другие, поддерживает огромное количество форматов: FB2, EPUB, PDF, DJVU, MOBI, PDF, HTML, DOC, RTF, TXT, ZIP, а также аудиоформаты MP3, AAC, M4B. Присутствуют основные настройки: размер и тип шрифта, поля и отступы, ориентация, возможность выбора темы.

На рисунке 4 представлена библиотека всех книг в мобильном приложении eReader Prestigo.



Рисунок 4 ⎯ Библиотека приложения eReader Prestigo

### 1.4.5 BookMate

При первом запуске у пользователя сразу узнают, что ему интересно, какие книги он предпочитает. На основе этих данных сервис создает списки рекомендаций. Кроме того, прочитав книгу, можно оставить отзыв и впечатления о ней. Без регистрации можно читать бесплатные книги или те, что хранятся в памяти устройства (EPUB или FB2). Сама читалка предлагает минимальный набор настроек, которого в целом достаточно для комфортного чтения.

Библиотека приложения BookMate приведена на рисунке 5.

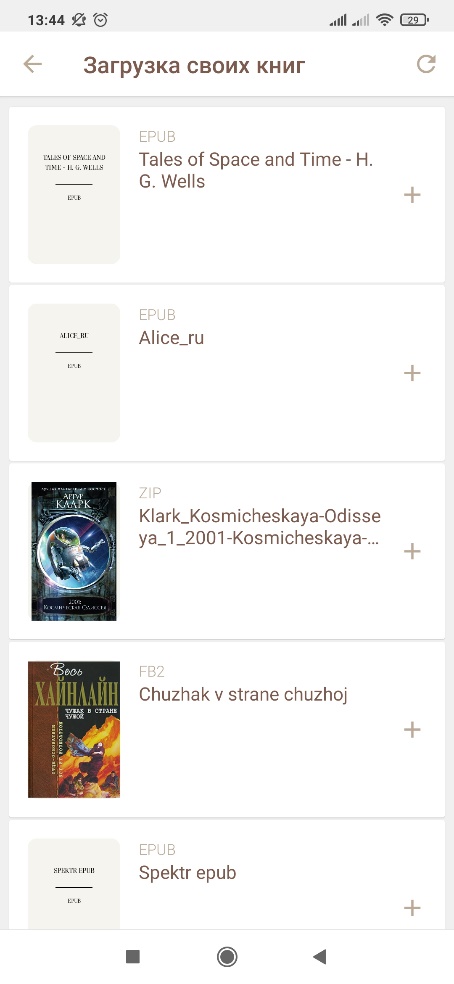


Рисунок 5 ⎯ Библиотека приложения BookMate

Можно сделать вывод, что положительными качествами разрабатываемого приложения должны быть:

* оснащенность необходимыми функциями;
* интуитивно понятный и удобный интерфейс;
* правильная цветовая гамма.

Проанализировав данные приложения и выделив положительные и отрицательные качества, можно сделать вывод, что при разработке планируется использовать положительные качества.

# 2 Проектирование приложения

## 2.1 UI/UX дизайн проекта

Дизайн приложения разработан в программе Figma.

При анализе и проектировании программного продукта определены следующие основные экраны:

* экран приветствия;
* экран с меню приложения;
* экран со всеми документами;
* экран с коллекциями;
* экран со списком документов определенной коллекции;
* экран для чтения документов.

Для приложения выбрана монохромная цветовая схема, то есть выбранные цвета являются оттенками одного основного цвета. Такие схемы позволяют создать эффект комфорта для глаз.

Цветовая схема составлена преимущественно из темных оттенков, так как по статистике такие цвета являются наиболее популярными среди пользователей.

Для данного приложения выбран графический логотип и помещен на фоне, соответствующем теме приложения. На рисунке 1 представлен логотип приложения.



Рисунок 6 – Логотип приложения «ReaDocs»

На экране приветствия пользователю представлен логотип приложения. Экран приветствия представлен на рисунке 7.



Рисунок 7 – Экран приветствия

Навигация между экранами осуществляется при помощи кнопок через экран меню. Из меню можно перейти на все основные экраны. Меню приложения представлено на рисунке 8.

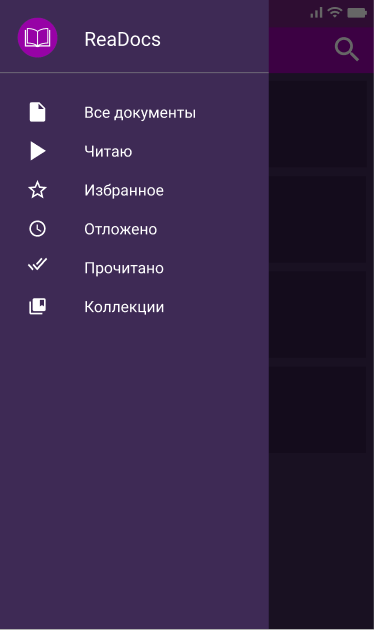


Рисунок 8 – Меню приложения

Все документы на устройстве пользователь может посмотреть на экране «Все документы». На экране со списком документов у пользователя есть возможность добавить документ в коллекции, перейти на страницу чтения документа, переименовать или удалить его. Экран со всеми документами представлен на рисунке 9.

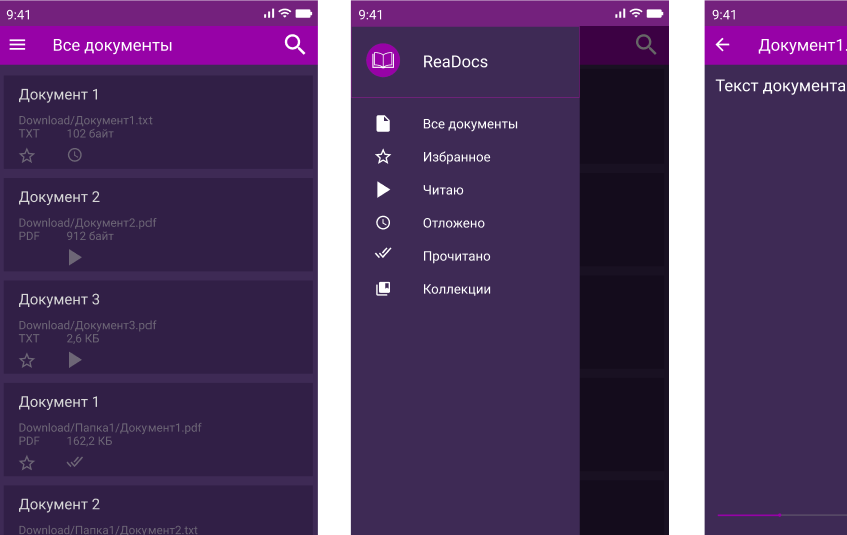


Рисунок 9 – Экран со всеми документами

На экране с коллекциями пользователь может создать коллекцию, просмотреть список созданных коллекций, переименовать или удалить коллекцию и перейти на экран просмотра списка документов в нужной коллекции. Экран с коллекциями представлен на рисунке 10.

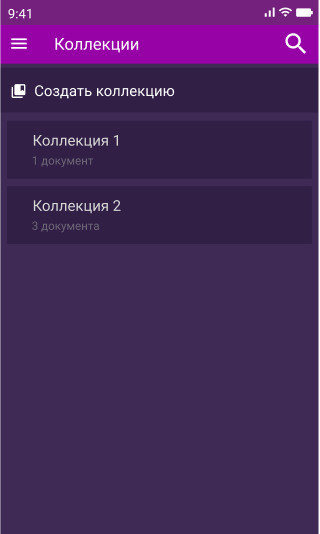


Рисунок 10 – Экран с коллекциями

После добавления в коллекцию по умолчанию документ в списке помечается знаком коллекции. Так, например, документы, которые пользователь пометил знаком «Избранное» расположены на соответствующем экране. На рисунке 11 показан экран со списком документов коллекции «Избранное».

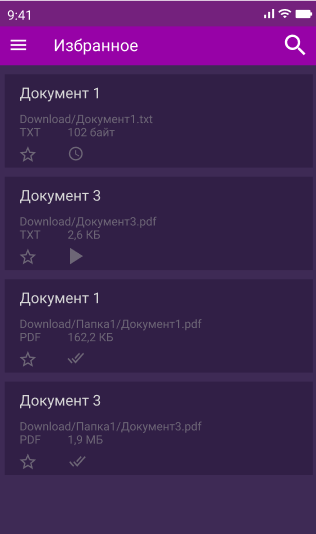


Рисунок 11 – Экран со списком документов коллекции «Избранное»

При нажатии на документ открывается экран чтения документа. На нем пользователь может изменять размер текста и перемещаться по документу. На рисунке 12 показан экран для чтения документов.

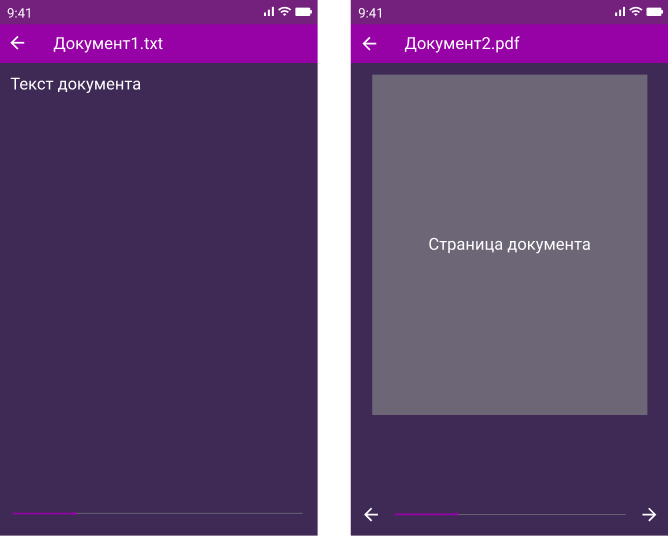


Рисунок 12 – Экран для чтения документов

## 2.2 Выбор технологии, языка и среды программирования

Для разработки приложения выбран язык программирования Java, так как он обладает следующими преимуществами:

* подходит для разработки как нативных, так и кроссплатформенных приложений;
* имеет огромное количество готовых библиотек;
* обладает сильным сообществом;
* полностью готов к работе с ним и широко распространен.

В качестве среды программирования было выбрано Android Studio, так как он обладает следующими преимуществами по сравнению с другими средами разработки:

* редактор кода, с которым удобно работать;
* поддержка системы контроля версий;
* скорость сборки приложения;
* предварительная проверка созданного приложения на предмет ошибок в нем;
* эмуляция устройств, возможность проводить тестирование и анализ кода.

# 3 Разработка мобильного приложения

## 3.1 Разработка базы данных

В приложении возникает необходимость создания базы данных, так как нужно хранить информацию о созданных пользователем коллекциях и вложенных в эти коллекции документах. Для этой цели было решено использовать SQLite, поскольку он позволяет хранить базу данных локально на устройстве пользователя.

База данных состоит из трех таблиц: "Документы", "Коллекции" и таблицы, связывающей их между собой. В таблице «Документы» необходимо хранить его название, путь, формат и размер. В таблице «Коллекции» необходимо хранить только ее название. Между таблицами " Documents " и " Collections " существует связь многие-ко-многим, так как один документ может находиться в нескольких коллекциях одновременно, а одна коллекция может содержать несколько документов. Для реализации этой связи была создана таблица "CollectionDocument".

На рисунке 13 представлена ER-диаграмма базы данных приложения.

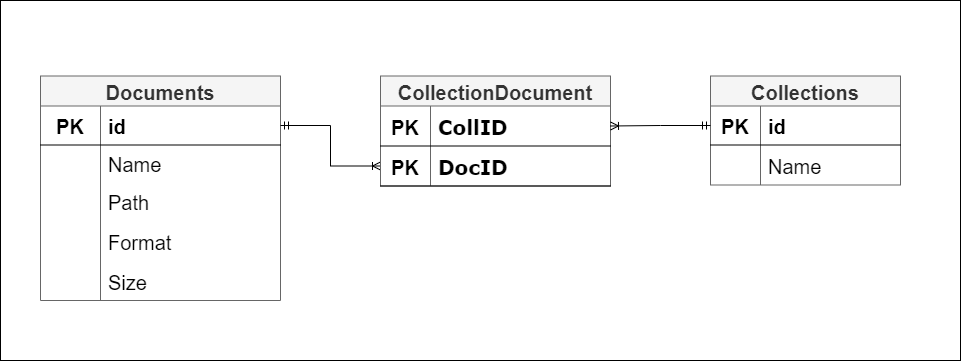


Рисунок 13 – ER-диаграмма базы данных

## 3.2 Разработка используемых плагинов

Ниже на рисунке 14 изображено название эмуляторов, используемых для запуска приложения в Android Studio.

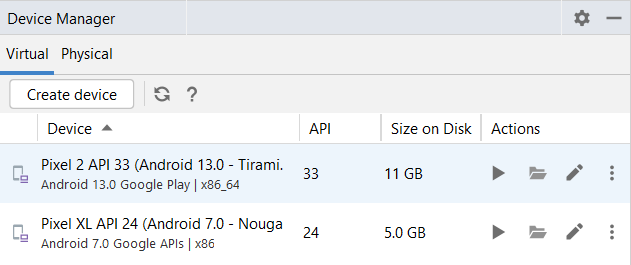


Рисунок 14 – Используемые эмуляторы

Для разработки были использованы встраиваемые плагины, которые создаются вместе с проектом, а также дополнительные плагины, которые нужны для функциональности приложения и изображены на Листинге 1.

Листинг 1 – Используемые плагины

implementation group: 'org.apache.poi', name: 'poi-ooxml', version: '3.17'  
implementation group: 'org.apache.xmlbeans', name: 'xmlbeans', version: '3.1.0'  
implementation 'javax.xml.stream:stax-api:1.0'  
implementation 'com.fasterxml:aalto-xml:1.2.2'  
  
implementation 'androidx.appcompat:appcompat:1.6.1'  
implementation 'com.google.android.material:material:1.9.0'  
implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
  
implementation(platform("org.jetbrains.kotlin:kotlin-bom:1.8.0"))  
implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.6.1'  
implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.6.1'  
  
implementation 'androidx.navigation:navigation-fragment:2.5.3'  
implementation 'androidx.navigation:navigation-ui:2.5.3'  
implementation 'androidx.legacy:legacy-support-v4:1.0.0'  
implementation 'androidx.recyclerview:recyclerview:1.3.0'  
testImplementation 'junit:junit:4.13.2'  
androidTestImplementation 'androidx.test.ext:junit:1.1.5'  
androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'

## 3.3 Описание разработанных процедур и функций

В приложении разработаны следующие функции:

* вывод списка документов текущей коллекции;
* добавление документа в коллекцию;
* переименование и удаление документа;
* чтение документа;
* вывод списка коллекций;
* создание коллекции;
* переименование и удаление коллекции;

Ниже на Листинге 2 представлен метод, который отвечает за формирование списка документов и его вывод. Он устанавливает адаптер с корректным внешним видом для списка документов, вызывает метод updateQuantityDocuments() для обновления количества документов в списке, устанавливает переходы на страницы чтения PdfActivity и TxtActivity при простом касании и открытие диалогового окна EditDocumentDialog при долгом касании. В списке нет документов, то метод выводит информацию об этом.

Листинг 2 – Метод для вывода списка документов текущей коллекции

public void formationViews() {  
 lvDocuments.setAdapter(adapter); //Установка адаптера в ListView  
 updateQuantityDocuments(); //Обновление вывода количества документов  
 //Открытие активности для просмотра документа при выборе документа из списка  
 lvDocuments.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
 Intent intent;  
 //Открытие активности для PDF-файла  
 if (listDocuments.get(i).getDocFormat().equals("PDF")) {  
 intent = new Intent(view.getContext(), PdfActivity.class);  
 }  
 //Открытие активности для TXT, DOC или DOCX файла  
 else{  
 intent = new Intent(view.getContext(), TxtActivity.class);  
 }  
 intent.putExtra("fileName", listDocuments.get(i).getDocName());  
 intent.putExtra("filePath", listDocuments.get(i).getDocPath());  
 startActivity(intent);  
 }  
 });  
 lvDocuments.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {  
 @Override  
 public boolean onItemLongClick(AdapterView<?> adapterView, View view, int i, long l) {  
 Intent intent = new Intent(view.getContext(), EditDocumentDialog.class);  
 intent.putExtra("fileName", listDocuments.get(i).getDocName());  
 intent.putExtra("filePath", listDocuments.get(i).getDocPathWithFolderNumber());  
 intent.putExtra("fileCollections", listDocuments.get(i).getDocCollections());  
 startActivity(intent);  
 return true;  
 }  
 });  
 //Вывод информации об отсутствии документов в коллекции  
 if(lvDocuments.getCount() == 0){  
 lvDocuments.setVisibility(View.*GONE*);  
 tvNoDocuments.setVisibility(View.*VISIBLE*);  
 } else {  
 lvDocuments.setVisibility(View.*VISIBLE*);  
 tvNoDocuments.setVisibility(View.*GONE*);  
 }  
}

Ниже на Листинге 3 представлен метод, который реализует добавление документа в коллекцию. Он получает имя коллекции и путь к документу и определяет индексы коллекции и документа с помощью функций getIdDocumentByPath() и getIdCollectionByName(). После метод создает связь и записывает полученные индексы в таблицу CollectionDocument базы данных.

Листинг 3 – Метод для добавления документа в коллекцию

//Добавление указанного документа в указанную коллекцию  
public void addDocumentInCollection(String nameColl, String pathDoc) {  
 int idDoc = getIdDocumentByPath(pathDoc); //Индекс документа в таблице Documents  
 int idColl = getIdCollectionByName(nameColl); //Индекс коллекции в таблице Collections  
 SQLiteDatabase database = getBaseContext().openOrCreateDatabase("ReaDocs.db", *MODE\_PRIVATE*, null);  
 database.execSQL("INSERT INTO CollectionDocument VALUES (" + idColl + ", " + idDoc + ")");  
 database.close();  
}

Ниже на Листинге 4 представлен метод, который используемый для переименования документов. Он вызывает метод проверки уникальности нового пути документа. Если путь уникален, имя и путь документа меняется на устройстве и в базе данных.

Листинг 4 – Метод для переименования документа

public void renameDocument(String newNameDoc, String newPathDoc, String oldPathDoc) {  
 //Получение файла по старому пути документа  
 String[] oldPath = oldPathDoc.split(" --- ");  
 String folderNumber = oldPath[0], oldPathDocWithoutFolderNumber = oldPath[1];  
 File oldFile = new File(oldPathDocWithoutFolderNumber);  
 //Получение файла по новому пути документа  
 String[] newPath = newPathDoc.split(" --- ");  
 String newPathDocWithoutFolderNumber = newPath[1];  
 File newFile = new File(newPathDocWithoutFolderNumber);  
 //Проверка уникальности пути документа  
 if (checkPathDoc(newPathDoc)) {  
 if (oldFile.renameTo(newFile)) {  
 renameDocumentInDatabase(newNameDoc, newPathDoc, oldPathDocWithoutFolderNumber, folderNumber);  
 //Уведомление об успешном изменении имени  
 Toast toast = Toast.*makeText*(getApplicationContext(),  
 "Имя файла успешно изменено на " + newNameDoc, Toast.*LENGTH\_SHORT*);  
 toast.show();  
 } else {  
 //Уведомление об ошибке при изменении имени  
 Toast toast = Toast.*makeText*(getApplicationContext(),  
 "Не удалось изменить имя файла", Toast.*LENGTH\_SHORT*);  
 toast.show();  
 }  
 } else {  
 //Уведомление о неуникальности имени  
 Toast toast = Toast.*makeText*(getApplicationContext(),  
 "Документ с таким именем уже существует", Toast.*LENGTH\_SHORT*);  
 toast.show();  
 }  
}

Ниже на Листинге 5 представлен метод для удаления документа. Он удаляет документ с устройства. Если документ удален успешно, то документ удаляется из базы данных и из всех коллекций в базе.

Листинг 5 – Метод для удаления документа

public void deleteDocument(String nameDoc, String path) {  
 //Получение файла по пути документа  
 String[] pathDoc = path.split(" --- ");  
 File file = new File(pathDoc[1]);  
  
 if (file.delete()) {  
 deleteDocumentFromCollections(path);  
 deleteDocumentFromDatabase(path);  
 //Уведомление об успешном удалении  
 Toast toast = Toast.*makeText*(getApplicationContext(),  
 "Файл " + nameDoc + " успешно удален", Toast.*LENGTH\_SHORT*);  
 toast.show();  
 } else {  
 //Уведомление об ошибке при удалении  
 Toast toast = Toast.*makeText*(getApplicationContext(),  
 "Не удалось удалить файл " + nameDoc, Toast.*LENGTH\_SHORT*);  
 toast.show();  
 }  
}

Ниже на Листинге 6 представлен класс для чтения Txt-документов. Он позволяет прочитать текст документа, изменить размер, перемещаться по тексту, вернуться в начальный размер по умолчанию при долгом нажатии по области текста.

Листинг 6 – Класс для чтения Txt-документов

public class TxtActivity extends AppCompatActivity {  
 private String path; //Путь к файлу  
 private TextView textViewTxt; //TextView для вывода документа  
 private int currentTextSize = 18; //Текущий размер шрифта документа  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_txt*);  
 //Представление кнопки Назад в ActionBar  
 if (getSupportActionBar() != null) {  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
 //Изменение цвета ActionBar  
 getSupportActionBar().setBackgroundDrawable(new ColorDrawable(getResources().getColor(R.color.*purple\_500*)));  
 }

getWindow().setNavigationBarColor(getResources().getColor(R.color.*purple\_900*)); //Установление цвета панели навигации  
 textViewTxt = findViewById(R.id.*textViewTxt*);  
 //Получение данных документа из Intent  
 path = getIntent().getStringExtra("filePath"); //Путь к файлу  
 setTitle(getIntent().getStringExtra("fileName")); //Установка заголовка активности  
 //Ползунок настройки масштаба (Минимум = 15, Максимум = 35, Начальное значение = 18)  
 SeekBar sbScale = findViewById(R.id.*sbScale*);  
 //Прослушиватель изменения значения ползунка  
 sbScale.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
 //Метод изменения значения ползунка  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
 //Изменение мастштаба  
 currentTextSize = seekBar.getProgress();  
 displayTxtDoc();  
 }  
 //Метод начала изменения значения ползунка  
 @Override  
 public void onStartTrackingTouch(SeekBar seekBar) {  
 }  
 //Метод окончания изменения значения ползунка  
 @Override  
 public void onStopTrackingTouch(SeekBar seekBar) {  
 }  
 });  
 //Возвращение в начальный масштаб при долгом касании  
 textViewTxt.setOnLongClickListener(new View.OnLongClickListener() {  
 @Override  
 public boolean onLongClick(View view) {  
 currentTextSize = 18;  
 sbScale.setProgress(18);  
 displayTxtDoc();  
 return false;  
 }  
 });  
 }  
 @Override public void onStart() {  
 super.onStart();  
 displayTxtDoc();  
 }  
 //Чтение txt-документа и вывод  
 private void displayTxtDoc(){  
 textViewTxt.setTextSize((float)currentTextSize); //Изменение размера шрифта  
 File file = new File(path);  
 try(FileInputStream fileInput = new FileInputStream(file)) {  
 byte[] bytes = new byte[fileInput.available()];  
 fileInput.read(bytes);  
 String fileText = new String(bytes);  
 textViewTxt.setText(fileText);  
 }  
 catch(IOException exception) {  
 Toast.*makeText*(this, "Документ защищен паролем.", Toast.*LENGTH\_SHORT*).show();  
 finish();  
 }  
 }  
}

Ниже на Листинге 7 представлен метод для создания коллекций. Он проверяет имя новой коллекции. Если имя уникально, то получается индекс для коллекции и коллекция добавляется в таблицу Collections базы данных.

Листинг 7 – Метод для создания коллекции

public void addCollection(String nameColl) {  
 if (nameColl.equals("")) {  
 Toast toast = Toast.*makeText*(getApplicationContext(),  
 "Имя коллекции не может быть пустым", Toast.*LENGTH\_SHORT*);  
 toast.show();  
 }  
 else if (checkNameColl(nameColl)) {  
 SQLiteDatabase database = getBaseContext().openOrCreateDatabase("ReaDocs.db", *MODE\_PRIVATE*, null);  
 Cursor queryColl = database.rawQuery("SELECT \* FROM Collections;", null);  
 int idColl = getIdCollectionToCreate();  
 database.execSQL("INSERT INTO Collections VALUES " +  
 "(" + idColl + ", '" + nameColl + "');");  
 queryColl.close();  
 database.close();  
 //Уведомление об успешном создании файла  
 Toast toast = Toast.*makeText*(getApplicationContext(),  
"Коллекция \"" + nameColl + "\" успешно создана", Toast.*LENGTH\_SHORT*);  
 toast.show();  
 } else {  
 //Уведомление о неуникальности имени  
 Toast toast = Toast.*makeText*(getApplicationContext(),  
"Коллекция с таким именем уже существует", Toast.*LENGTH\_SHORT*);  
 toast.show();  
 }  
}

Ниже на Листинге 8 представлен метод, который отвечает за формирование списка коллекций и вывод. Он устанавливает переходы на страницу просмотра списка документов в коллекции при простом касании и открытие диалогового окна редактирования при долгом касании. Если в списке нет коллекций, то метод выводит информацию об этом.

Листинг 8 – Метод для вывода списка коллекций

public void formationViews() {   
 lvCollections.setAdapter(adapter);  
 //Открытие активности для просмотра списка документов выбранной коллекции  
 lvCollections.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
 Intent intent = new Intent(view.getContext(), DocsFromCollectionActivity.class);  
 intent.putExtra("collName", listCollections.get(i).getCollName());  
 startActivity(intent);  
 }  
 });  
 lvCollections.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {  
 @Override  
 public boolean onItemLongClick(AdapterView<?> adapterView, View view, int i, long l) {  
 Intent intent = new Intent(view.getContext(), EditCollectionDialog.class);  
 intent.putExtra("collName", listCollections.get(i).getCollName());  
 startActivity(intent);  
 return true;  
 }  
 });  
 //Вывод информации об отсутствии коллекций  
 if(lvCollections.getCount() == 0){  
 lvCollections.setVisibility(View.*GONE*);  
 tvNoCollections.setVisibility(View.*VISIBLE*);  
 } else {  
 lvCollections.setVisibility(View.*VISIBLE*);  
 tvNoCollections.setVisibility(View.*GONE*);  
 }  
}

Рассмотренные листинги предоставляют ключевые функциональные возможности мобильного приложения для чтения и группировки документов. Комбинация этих функций позволяет создать интуитивно понятное и мощное приложение, которое удовлетворяет потребности пользователей в чтении электронных книг.

# 4 Тестирование

## 4.1 Протокол тестирования дизайна приложения

Для проведения тестирования дизайна приложение были выбраны устройства Xiaomi Redmi Note 10S c разрешение экрана 2400×1080 и Google Pixel 2 c разрешение экрана 1920×1080.

Проверка была проведена на:

* оптимальный размер кнопок;
* читабельный размер шрифта;
* приятное глазу цветовое сочетание;
* корректное расположение элементов.

В результате тестирования дизайна все пункты были пройдены.

## 4.2 Протокол тестирования функционала приложения

Для тестирования функционала были разработаны тест-кейсы, каждый из которых описывает небольшую последовательность действий с одним конкретным результатом. Благодаря тест-кейсам всегда известно, как и что протестировать оптимальным количеством проверок, и не забыть о нюансах, так как записан каждый шаг. Разработанные тест-кейсы представлены в таблицах от 1 до 9.

Таблица 1 – Test-case 1

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Дымовое тестирование | | |
| Функция: | Запуск приложения | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
|  | |  |  |
| Шаги теста: | |  |  |
| Нажатие на ярлык приложения | | Корректное открытие приложения с запросом разрешений для функционирования | пройден |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Постусловие: |  |  |
| Продолжение работы | Приложение работает и функционирует | пройден |

Таблица 2 – Test-case 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название: | Критическое тестирование | | | | |
| Функция: | Работа навигационного меню | | | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован | | |
| Предусловие: | |  |  | | |
| Запуск приложения | | Корректная работа приложения | пройден | | |
| Нажатие на кнопку меню | | Появление навигационного меню | пройден | | |
| Шаги теста: | |  |  | | |
| Нажатие на кнопку меню «Все документы» | | Перемещение на страницу со списком всех документов | пройден | | |
| Нажатие на кнопку меню «Избранное» | | Перемещение на страницу со списком документов коллекции «Избранное» | пройден | | |
| Нажатие на кнопку меню «Читаю» | | Перемещение на страницу со списком документов коллекции «Читаю» | пройден | | |
| Нажатие на кнопку меню «Отложено» | | Перемещение на страницу со списком документов коллекции «Отложено» | пройден | | |
| Нажатие на кнопку меню «Прочитано» | | Перемещение на страницу со списком документов коллекции «Прочитано» | пройден | | |
| Нажатие на кнопку меню «Коллекции» | | Перемещение на страницу со списком коллекций. | | пройден |
| Нажатие на кнопку меню «Помощь» | | Перемещение на страницу со вспомогательной информацией для пользователя | | пройден |
| Постусловие: | |  | |  |
| Продолжение работы | | Приложение работает и функционирует | | пройден |

Таблица 3 – Test-case 3

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Отображение списка всех документов | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения. Открытие главной страницы со всеми документами. | пройден |
| Шаги теста: | |  |  |
| Просмотр списка документов | | Отображение списка всех документов | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Таблица 4 – Test-case 4

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Добавление документа в коллекцию | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения. Открытие главной страницы со всеми документами. | пройден |
| Долгое нажатие на один из документов списка | | Открытие диалогового окна редактирования документа | пройден |
| Шаги теста: | |  |  |
| Выбор коллекции. Нажатие на выбранную коллекцию. | | Изменение цвета кнопки выбранной коллекции | пройден |
| Выход из окна редактирования документа | | Отображение знака коллекции на документе в списке |  |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Таблица 5 – Test-case 5

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Переименование документа | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения. Открытие главной страницы со всеми документами. | пройден |
| Долгое нажатие на один из документов списка | | Открытие диалогового окна редактирования документа | пройден |
| Шаги теста: | |  |  |
| Нажатие на кнопку «Переименовать» | | Открытие диалогового окна переименования документа | пройден |
| Ввод нового имени документа. Нажатие на кнопку «Сохранить» | | Изменение имени документа в списке документов | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Таблица 6 – Test-case 6

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Удаление документа | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения. Открытие главной страницы со всеми документами. | пройден |
| Долгое нажатие на один из документов списка | | Открытие диалогового окна редактирования документа | пройден |
| Шаги теста: | |  |  |
| Нажатие на кнопку «Удалить» | | Открытие диалогового окна подтверждения удаления документа | пройден |
| Подтверждение документа нажатием на кнопку «Удалить» | | Удаление документа с устройства и из списка документов | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Таблица 7 – Test-case 7

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Чтение документа | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения. Открытие главной страницы со всеми документами. | пройден |
| Шаги теста: | |  |  |
| Нажатие на один из документов списка | | Открытие окна чтения документа | пройден |
| Перемещение ползунка размера текста документа | | Корректное изменение размеров текста документов | пройден |
| Нажатие на кнопки для перемещения по тексту документа или вертикальная прокрутка | | Корректное перемещение по тексту документа | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Таблица 8 – Test-case 8

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Создание коллекции | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения. Открытие главной страницы со всеми документами. | пройден |
| Переход на страницу со списком коллекций в навигационном меню | | Открытие страницы со списком коллекций | пройден |
| Шаги теста: | |  |  |
| Нажатие на кнопку «Создать коллекцию» | | Открытие диалогового окна создания коллекции | пройден |
| Ввод имени для новой коллекции. Нажатие на кнопку «Создать» | | Сохранение коллекции в базе данных приложения и добавление в список коллекций | пройден |

Продолжение таблицы 8

|  |  |  |
| --- | --- | --- |
| Постусловие: |  |  |
| Продолжение работы | Приложение работает и функционирует | пройден |

Таблица 9 – Test-case 9

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Отображение списка коллекций | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения. Открытие главной страницы со всеми документами. | пройден |
| Переход на страницу со списком коллекций в навигационном меню | | Открытие страницы со списком коллекций | пройден |
| Шаги теста: | |  |  |
| Просмотр списка коллекций | | Отображение списка всех коллекций | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |
| Предусловие: | |  |  |

Заключение

В результате работы было разработано мобильное приложение, осуществляющее чтение электронных документов из памяти смартфона, работающего на операционной системе Android.

Данное приложение позволяет осуществить чтение документов в различных форматах, создание собственных коллекций и группировку документов по коллекциям. Это позволяет пользователю самостоятельно формировать удобный для него набор документов в каждой коллекции и тем самым ускорить работу с документами.

В ходе выполнения проекта были решены следующие задачи:

* проведены исследование предметной области и анализ целевой аудитории приложения. Исследование включало определение потребностей и предпочтений пользователей, их основные характеристики и требования;
* проведен обзор наиболее актуальных среди пользователей существующих приложений, связанных с чтением документов. Этот анализ позволил выявить преимущества для приложения, улучшить дизайн и функциональность приложения;
* разработана структура приложения, определяющая его основные модули, функциональные блоки и связи между ними. Это позволило логически организовать приложение и обеспечить эффективное взаимодействие между различными его частями;
* проведены исследования и изучены различные методы и инструменты для чтения электронных документов из памяти устройства Android. Это позволило определить наиболее эффективные и надежные способы чтения документов, которые могут быть использованы в приложении;
* проведено изучение методов взаимодействия с файловой системой Android, необходимых для управления документами. Это позволило разработать соответствующую функциональность, чтобы пользователи могли легко управлять своими документами, изменять имя и удалять их;
* выполнен этап разработки самого приложения, включая создание кода, реализацию функциональности и интеграцию всех компонентов. Результатом был готовый программный продукт, способный читать документы с устройства и группировать их согласно заданным конфигурациям.

Во время разработки возникли трудности с чтением документов формата docx. Данные проблемы были решены путем использования стороннего плагина Apache POI, при помощи которого стало возможным извлечение содержимого из документов docx и вывод на экран пользователя.

В дальнейшем планируется совершенствовать приложение и вносить следующие изменения:

* расширение функционала работы с содержимым документа;
* увеличение количества доступных для чтения форматов;
* функция выбора нескольких документов для совместного редактирования.

Библиография

Нормативно-правовые акты:

1 ГОСТ Р 2.105-2019. ЕСКД. Общие требования к текстовым документам. – Москва : Стандартинформ, 2019. – 36 с

Электронные ресурсы:

1 Android tools [Электронный ресурс]. – Получаем разрешение MANAGE\_EXTERNAL\_STORAGE для приложения. – URL: https://android-tools.ru/coding/poluchaem-razreshenie-manage\_external\_storage-dlya-prilozheniya/ (дата обращения: 25.11.2022)

2 Metanit [Электронный ресурс]. – Работа с базами данных SQLite. – URL: https://metanit.com/java/android/14.5.php (дата обращения: 02.12.2022)

3 GeeksforGeeks [Электронный ресурс]. – Different ways of Reading a text file in Java. – URL: https://www.geeksforgeeks.org/different-ways-reading-text-file-java/ (дата обращения: 15.12.2022)

4 Android tools [Электронный ресурс]. – Работа с PDF. Чтение PDF-файлов – URL: https://android-tools.ru/coding/rabota-s-pdf-chtenie-pdf-fajlov/ (дата обращения: 15.12.2022)

5 JavaDevBlog [электронный ресурс]. – Чтение документа Word в формате docx с помощью Apache POI. – URL: https://javadevblog.com/chtenie-dokumenta-word-v-formate-docx-s-pomoshh-yu-apache-poi.html (дата обращения: 20.12.2022)

Приложение А

(обязательное)

Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж

имени Б. С. Галущака»

Техническое задание

разработка мобильного приложения для ЧТЕНИЯ ЭЛЕКТРОННЫХ ДОКУМЕНТОВ

НАТКиГ.202300.010.000ПЗ

Выполнил:

студент группы ПР-20.101

Фаруков В. С.

2023**Содержание**

[Введение 36](#_Toc138507652)

[1 Назначение разработки 36](#_Toc138507653)

[2 Требования к мобильному приложению 37](#_Toc138507654)

[2.1 Требования к функциональным характеристикам 37](#_Toc138507655)

[2.2 Требования к надёжности 37](#_Toc138507656)

[2.3 Условия эксплуатации 37](#_Toc138507657)

[2.4 Требования к составу и параметрам технических средств 38](#_Toc138507658)

[2.5 Требования к информационной и программной совместимости 38](#_Toc138507659)

[2.6 Требования к защите информации 38](#_Toc138507660)

[2.7 Требования к маркировке и упаковке 38](#_Toc138507661)

[3 Требования к программной документации 38](#_Toc138507662)

[4 Технико-экономические показатели 38](#_Toc138507663)

[5 Стадии и этапы разработки 39](#_Toc138507664)

[6 Порядок контроля и приёмки 40](#_Toc138507665)

Введение

Настоящее техническое задание распространяется на разработку информационной системы «Разработка мобильного приложения для чтения электронных документов», используемого для просмотра и чтения электронных документов, группировки документов по коллекциям.

Наименование приложения: «ReaDocs».

Краткая характеристика области применения: приложение для чтения предназначено для чтения электронных книг и других текстовых документов на мобильных устройствах пользователей. Оно может использоваться как для личного чтения, так и для образовательных и профессиональных целей.

Основанием для проведения разработки является Протокол № Уч-34/1 от 7 марта 2023 года.

Наименование темы разработки – «Разработка мобильного приложения для чтения электронных документов».

Условное обозначение темы разработки – «Приложение для чтения – ReaDocs».

1 Назначение разработки

Основное назначение мобильного приложения заключается в следующих возможностях:

* просмотр списка документов;
* чтение документов;
* возможность группировки документов по коллекциям.

Работать с данной системой может пользователь – просматривает документы, читает, группирует документы.

2 Требования к мобильному приложению

2.1 Требования к функциональным характеристикам

Требования к составу выполняемых функций:

* просмотр списка документов;
* открытие документа для чтения;
* редактирование и удаление документов;
* создание коллекций;
* просмотр списка коллекций;
* редактирование и удаление коллекций;
* группировка документов по коллекциям;
* поиск документа или коллекции в списке;
* фильтрация списка документов.

Входные данные должны быть организованы в виде вводимого, в специальную форму, текста, соответствующего определённому шаблону. Данные, вводимые вручную, проверяются на корректность.

2.2 Требования к надёжности

Обеспечение устойчивого функционирования должно выполняться несколькими действиями:

* организация бесперебойного питания оборудования пользователя;
* использование лицензионного программного обеспечения.

Приложение должно контролировать входную информацию:

* соблюдение типов данных при заполнении полей;
* операции удаления, изменения и сохранения.

2.3 Условия эксплуатации

Обслуживание ИС включает в себя:

* информационное обслуживание – вывод информации;
* наличие электронных текстовых документов в памяти телефона.

2.4 Требования к составу и параметрам технических средств

Для работы приложения на устройстве требуется: объем свободной памяти не менее 1 ГБ, версия устройства Android 6 и выше.

2.5 Требования к информационной и программной совместимости

Проектирование взаимодействия с файловой системой должно быть выполнено в рамках разработки курсового проекта. При разработке взаимодействия с файловой системой должен быть использован язык программирования Java.

2.6 Требования к защите информации

Требования к защите информации не предъявляются.

2.7 Требования к маркировке и упаковке

Требования к маркировке и упаковке не предъявляются.

3 Требования к программной документации

Состав программной документации должен включать в себя:

* техническое задание;
* проектную документацию.

4 Технико-экономические показатели

Экономические преимущества разработки и ориентировочная экономическая эффективность не рассчитывается.

5 Стадии и этапы разработки

Таблица 1 – Стадии разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название этапа | Срок, даты | Отчётность |
| 1 | Исследование предметной области | 15.03.2023-28.03.2023 | Пояснительная записка |
| 2 | Выбор моделей, описывающих предметную область | 01.04.2023-03.04.2023 | Пояснительная записка |
| 3 | Разработка технического задания | 03.04.2023-05.04.2023 | Техническое задание |
| 4 | Изучение Java и SQLite | 01.04.2023-  18.04.2023 | Пояснительная записка |
| 5 | Анализ требований и уточнение спецификаций | 14.04.2023 | Спецификации программного обеспечения |
| 6 | Проектирование структуры мобильного приложения, проектирование компонентов (технический проект) | 15.04.2023-29.04.2023 | Схема структурная системы и спецификации компонентов |
| 8 | Кодирование клиентской части | 01.05.2023-20.06.2023 | Программный  Продукт |
| 9 | Тестирование компонентов.  Сборка и комплексное тестирование | 20.06.2023 | Тексты программных компонентов |
| 10 | Разработка программной  документации | 20.06.2023-21.06.2023 | Программная  документация |
| 12 | Защита | 29.06.2023 |  |

6 Порядок контроля и приёмки

Виды испытаний – защита курсового проекта.

Общее требования к приёмке:

* техническое задание;
* пояснительная записка;
* программный продукт;
* презентация.