

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ  
КАЗАХСТАН  
Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»  
Кафедра IT-инжиниринг

**ДОПУЩЕН К ЗАЩИТЕ**

Заведующий кафедрой  
PhD, доцент

\_\_\_\_\_ Т.С. Картбаев  
« \_\_\_\_ » \_\_\_\_\_ 2019 г

**ДИПЛОМНЫЙ ПРОЕКТ**

На тему: Разработка мобильного приложения в среде Android «Библиотека»

Специальность 5В060200 – «Информатика»

Выполнила Иманбаева А.К.          Группа ИНФ-15-2  
Научный руководитель PhD, доцент Маликова Ф.У.

Консультанты:

по экономической части: к.э.н., доцент \_\_\_\_\_ А.И. Бекнешева  
« 09 » 04 \_\_\_\_\_ 2019 г.

по безопасности жизнедеятельности: ст. преп. \_\_\_\_\_ Ш.Ш. Бекбасаров  
« 02 » 04 \_\_\_\_\_ 2019 г.

по применению  
вычислительной техники: ст. преп. \_\_\_\_\_ М.Н. Майкотов  
« 14 » 05 \_\_\_\_\_ 2019 г.

Нормоконтролер: ст. преп. \_\_\_\_\_ Ж.К. Алимсентова  
« 16 » 05 \_\_\_\_\_ 2019 г.

Рецензент: к.т.н., доцент \_\_\_\_\_ Л.Ш. Балгабаева  
« \_\_\_\_ » \_\_\_\_\_ 2019 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ  
КАЗАХСТАН

Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В060200 – «Информатика»

**ЗАДАНИЕ**

на выполнение дипломного проекта

Студенту Иманбаевой Алуа Кайраткызы

Тема проекта: Разработка мобильного приложения в среде Android «Библиотека»

Утверждена приказом по университету № 124 от «26» октября 2018 г.

Срок сдачи законченной проекта «01» июня 2019 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): специальная литература, материалы сети Интернет, среда AndroidStudio, база данных SQLite, код реализации на языке JAVA.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

- Теоретические основы разработки мобильных приложений;
- Процесс разработки программного обеспечения;
- вопросы безопасности жизнедеятельности и охраны труда;
- экономическая эффективность работ по автоматизации проекта.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 19 таблиц, 27 иллюстрации и 8 схем.

Основная рекомендуемая литература:

- Использование мобильных устройств [Электронный ресурс]. – Режим доступа: <http://www.wi-life.ru/stati/wi-fi/marketingovye-statii-2/mobiledevices-use-aruba-research-results> (дата обращения: 15.05.2017).
- Рынок мобильных приложений в России и мире [Электронный ресурс]. – Режим доступа: [http://www.json.ru/poleznye\\_materialy/](http://www.json.ru/poleznye_materialy/)

3. Приложение для вуза «Мобильный РГППУ» [Электронный ресурс]. –Режим доступа:

[http://elar.rsvpu.ru/bitstream/123456789/20364/1/RSVPU\\_2017\\_341.pdf](http://elar.rsvpu.ru/bitstream/123456789/20364/1/RSVPU_2017_341.pdf)

5. Технологии разработки мобильных приложений [Электронный ресурс]. – Режим доступа: <https://rae.ru/forum2012/219/2709>(дата обращения: 15.05.2017).

Консультации по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	А.И. Бекишева	26.02-03.04.19	
Безопасности жизнедеятельности	Ш.Ш. Бекбасаров	26.02-03.04.19	
Программная часть	М.Н. Майкотов	19.05-14.05.19	
Нормоконтролер	Ж.К.Алимсеитова	1.03-15.05.19	

#### График подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Анализ современного состояния вопроса	15.11.2018	выполнено
Разработка информационной системы	23.01.2019	выполнено
Разработка программного обеспечения	04.02.2019	выполнено

Дата выдачи задания «24» сентября 2018 г.

Заведующий кафедрой \_\_\_\_\_ Т.С. Картбаев

Научный руководитель работы  Ф.У. Маликова

Задание принял к исполнению студент  А.К. Иманбаева

## **Аңдатпа**

Бұл дипломдық жобасы Android ортасында университет үшін «Кітапхана» мобильді қосымшасын әзірлеп, оқушыларға кітапханадағы кітаптарға тез қол жеткізуге мүмкіндік береді.

Бұл қосымша ұялы телефонда қажетті әдебиеттерді сақтауға мүмкіндік береді. Кітаптардан басқа, университет туралы жаңалықтар мен «Менің кітаптарым» қойындысы пайда болады, онда пайдаланушы қажетті кітаптарды немесе нұсқауларды қоса алады.

Бұл жобаның бағдарламалық жасақтамасы AndroidStudio ортасында жасалды, код Java-ның объектілі-бағытталған бағдарламалау тілінде жазылған. Негізгі бөлігі «Кітапхана» мобильді қосымшасын толығымен сипаттайды.

## **Аннотация**

В данном дипломном проекте разрабатывается мобильное приложение «Библиотека» для университета в среде Android, позволяющее студентам иметь быстрый доступ к книгам, которые есть в библиотеке.

Данное приложение даст возможность сохранять необходимую литературу в мобильном телефоне. Помимо книг, там будет отображаться новости университета и вкладку «Мои книги», куда пользователь сможет добавить нужные для себя книги или методические указания.

Программная часть в данном проекте разработана в среде Android Studio, код написан на объектно-ориентированном языке программирования Java.

В главной части полностью описывается мобильное приложение «Библиотека».

## **Annotation**

This graduation project is developing a mobile application "Library" for the university in the Android environment, allowing students to have quick access to books that are in the library.

This application will provide an opportunity to save the necessary literature in a mobile phone. In addition to books, there will be displayed the news of the university and the tab "My Books", where the user can add the necessary books or guidelines.

The software in this project was developed in the AndroidStudio environment, the code is written in an object-oriented programming language Java. The main part fully describes the mobile application "Library".

## Содержание

Введение.....	6
1 Анализ современного состояния вопроса.....	8
1.1 Анализ существующих систем.....	8
1.2 Постановка задачи дипломного проекта.....	12
2 Разработка информационной системы.....	13
2.1 Мобильное приложение.....	13
2.2 Web сайт и мобильное приложение.....	13
2.3 Функциональная структура ИС.....	14
2.4 Разработка бизнес-процессов ИС.....	18
2.5 Представление UML-диаграмм.....	19
3 Разработка программного обеспечения ИС.....	22
3.1 Обоснование выбора среды разработки.....	22
3.2 Структура программного обеспечения.....	28
3.3 Разработка интерфейса приложения(макет).....	29
3.4 Описание программы.....	31
3.4.1 Общие сведения.....	31
3.4.2 Описание логической структуры.....	31
3.4.3 Функциональное назначение.....	32
3.4.4 Используемые технические средства.....	35
3.5 Инструкция пользователю.....	35
4 Экономическая часть.....	44
4.1 Резюме.....	44
4.2 Трудоемкость разработки ПП.....	45
4.3 Расчет затрат на разработку ПП.....	47
4.4 Материальные затраты.....	47
4.5 Затраты на электроэнергию.....	47
4.6 Затраты на оплату труда.....	48
4.7 Социальный налог.....	49
4.8 Амортизация основных фондов.....	49
4.9 Прочие затраты.....	50
4.10 Смета на разработку ПП.....	53
4.11 Определение возможной цены ПП.....	54
5 Охрана труда и БЖД.....	57
5.1 Общее положение.....	57
5.2 Расчет вентиляции.....	58
5.3 Расчет тепловых нагрузок в помещении.....	58
5.4 Расчет теплового баланса помещения.....	62
5.5 Выбор кондиционера. Схема расположения.....	63
Заключение.....	66
Список использованной литературы.....	67
Приложение А.....	68
Приложение Б.....	70

## Введение

Сегодня все больше людей используют для обмена информацией не персональные компьютеры и ноутбуки, а современные мобильные телефоны, которые практически равны упомянутым устройствам. В результате разработка мобильных приложений становится все более популярной - над ними сегодня работает не меньше людей, чем над разработкой программного обеспечения для полнофункциональных компьютерных систем.

Развитие технологий и необходимость быстрого и удобного доступа пользователей к информации привели к тому, что мобильные телефоны, а именно приложения, специально предназначенные для них, используются в качестве основного источника доступа к информации и корпоративным ресурсам, таким как социальные сети, услуги связи, информационные порталы и многое другое.

Продуманный и оптимизированный продукт позволит не только упростить работу отдельного человека или компании в целом, но и решить массу бизнес-задач практически из любой точки мира. Например, интеграция такого продукта с массивными CRM-системами позволяет полностью контролировать работу организации или предприятия как в командировке или командировке, так и в отпуске.

Вскоре в IT-индустрии появятся программы для мобильных устройств.

Эта новая тенденция открывает доступ к мобильным устройствам для языков программирования, поэтому объем мобильных приложений и их доля на рынке растут.

Таким образом, актуальность темы дипломного проекта очевидна. В ближайшее время разработка и поддержка приложений на базе операционной системы Android будут наиболее востребованы на рынке программного обеспечения.

При разработке мобильных приложений наиболее популярными направлениями являются разработка под Android, а также разработка под iPhone / iPad / iOS. Эти платформы являются самыми популярными мобильными платформами сегодня.

В настоящее время самой распространенной операционной системой является Android OS. Android поддерживает большое количество устройств разных производителей. Основная причина распространения ОС Android - бесплатные инструменты разработки, тогда как разработка под IOS-системой требует высоких начальных затрат.

Из этого можно сделать вывод, что тема данного дипломного проекта является актуальной. В ближайшем будущем разработка и поддержка разработанных приложений на базе операционной системы Android будет более востребована на рынке программного обеспечения, чем другие операционные системы.

**Цель:** изучить и разработать удобное мобильное приложение, которое будет доступно для каждого, где можно будет получить доступ к книгам, методическим указаниям, которые есть в наличии.

**Задачи:**

1. Исследовать рынок, проанализировать уже существующие продукты.
2. Проанализировать выбранную предметную область.
3. Описать проектируемую разрабатываемую систему.

# 1 Анализ современного состояния вопроса

## 1.1 Анализ существующих систем

В настоящее время все больше и больше пользователей не хотят использовать сайты, которые не оптимизированы для просмотра на различных гаджетах. Всем известно, что качественное приложение для Android или iOS будет удобнее любого оптимизированного мобильного сайта. Поэтому многие разработчики пришли к выводу, что необходимо выпускать специализированное мобильное приложение.

Рассмотрим уже существующие мобильные приложения, связанные с поиском книг, чтением онлайн в приложении «Библиотека».

На рисунке 1.1 показано разработанное мобильное приложение «Библиотека ГГУ», которое было специально сделано для Гжельского государственного университета.

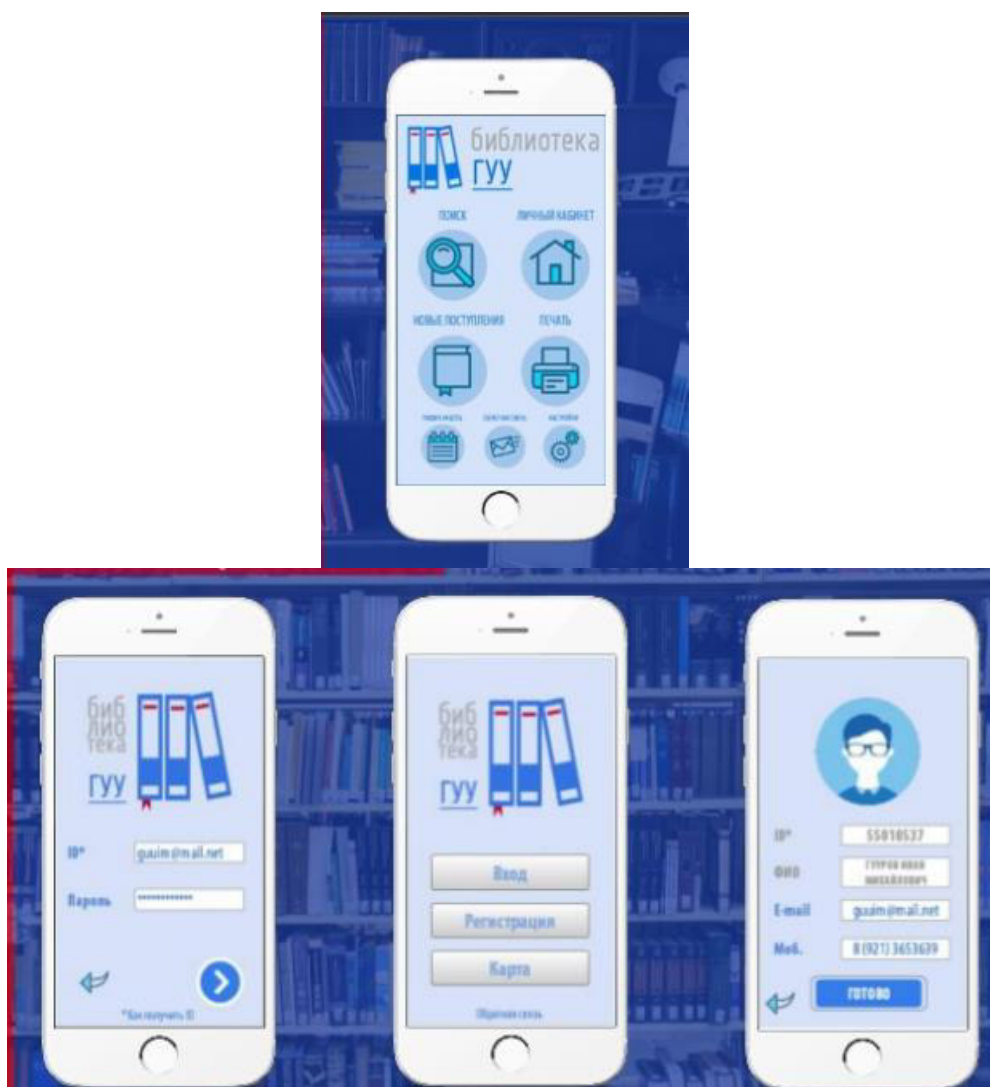


Рисунок 1.1 – Мобильное приложение «Библиотека ГГУ»



«Юрайта. Библиотека» - это одно из самых простых и удобных мобильных приложений для чтения книг. Основным преимуществом этого приложения является то, что вы можете читать загруженные книги в автономном режиме. Пользователю также предоставляется возможность выбрать шрифт и размер текста, что очень удобно для читателя. Есть много разных разделов, начиная от исследовательских книг до литературных произведений.

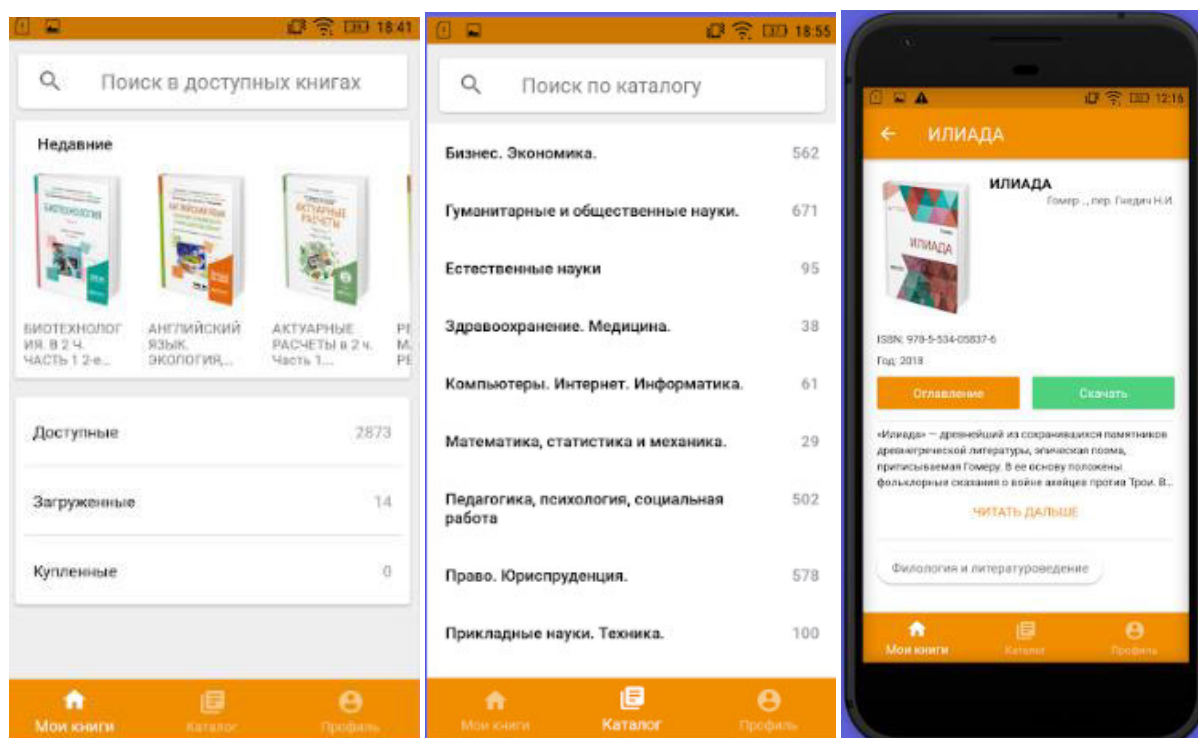


Рисунок 1.2 – Мобильное приложение «Юрайта. Библиотека»

Мобильное приложение «Моя библиотека» - это бесплатное приложение без рекламы для записи и создания каталога домашней библиотеки. Особенности приложения:

- добавление книг в библиотеку путем сканирования штрих-кодов (можно указать имя, автора, резюме, дату публикации, издателя, добавить обложку и т. д.);
- добавление книг в библиотеку по номеру ISBN или по ключевым словам;
- добавлять книги вручную;
- поиск книг в библиотеке;
- сортировка книг по имени, имени, категории (например, прочитанные и непрочитанные) и т.д. ;
- экспортировать всю библиотеку в один файл Excel;
- импортировать библиотеку из ранее экспортированного; - управление списком пожеланий.

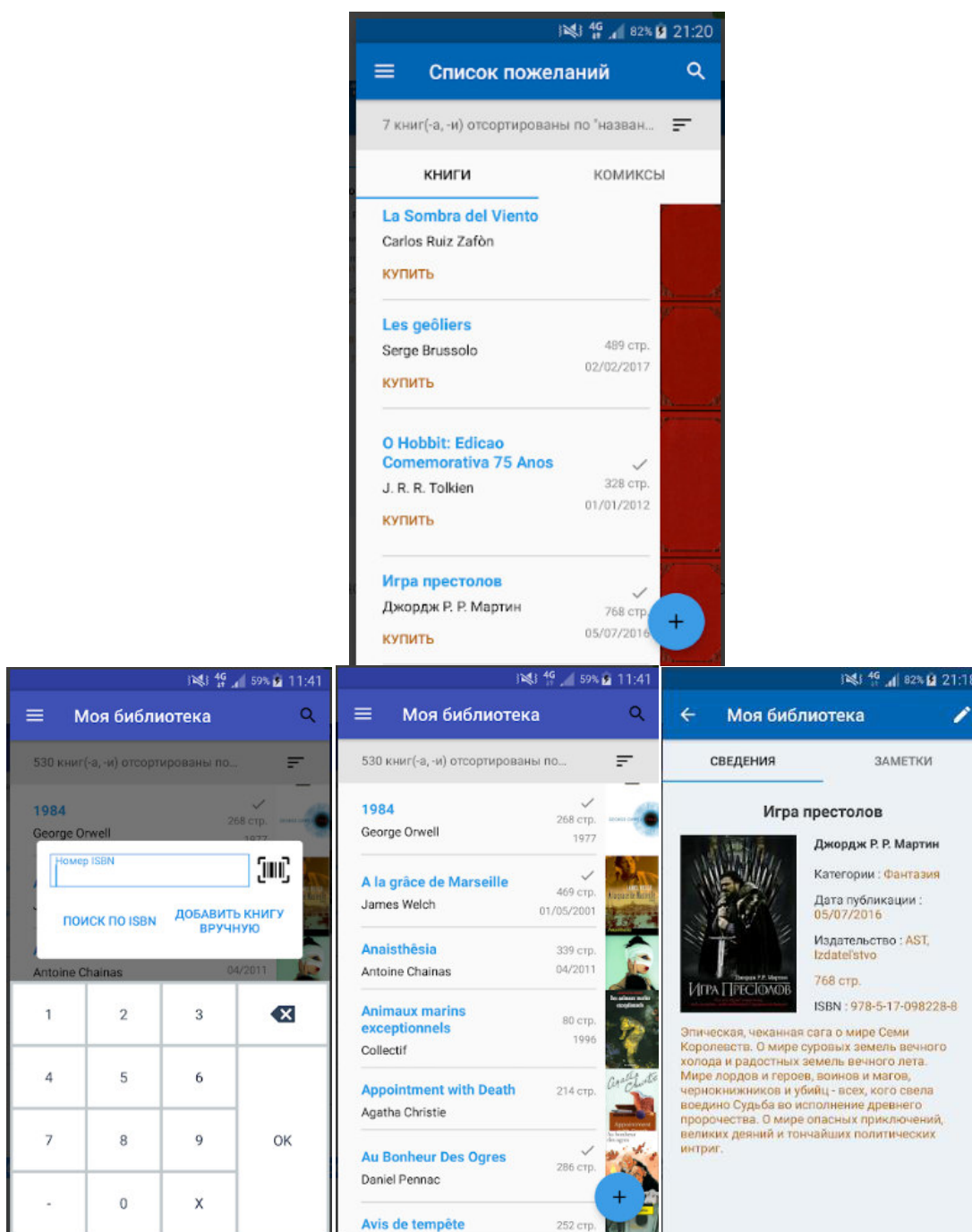


Рисунок 1.3 – Мобильное приложение «Моя библиотека»

Мобильное приложение ЭБС «Университетская библиотека онлайн», показанное на рисунке 1.4, осуществляет доступ ко всему контенту, на который оформлена подписка вашей организации. Читайте книги онлайн, или загружайте их на свое устройство для чтения без подключения к интернету.

В приложении сохранены важные функции сайта - личный кабинет, закладки, моя библиотека и т.д.

Для вашего удобства работы с контентом предусмотрен функционал закладок и цитат.

Приложение доступно для любого устройства, работающего на Android (от версии 4.1) и для iPad с IOS (от версии 8.0).

Многие книги имеют возможность скачивания на устройство и могут быть открыты только в приложении. На странице книги можно увидеть кнопки для скачивания и для чтения онлайн.

После скачивания можно читать загруженные издания даже без подключения к интернету.



Рисунок 1.4 – Мобильное приложение ЭБС

## 1.2 Постановка задачи дипломного проекта

Необходимо создать мобильное приложение на «AndroidStudio» где будет использоваться операционная система Android. Приложение должно будет иметь возможность вести поиск с фильтрами, по ключевым словам, и подача объявления путем его добавления в базу данных, которая находится в облаке.

База данных, которая будет привязана к приложению, должна иметь определенные требования для его полноценной работы:

- ввод данных через экранную клавиатуру девайса и их корректировку;
- обработка введенных данных;
- обработка поступающих данных со стороны сервера;
- вывод данных на дисплей девайса;

Исходные данные: числовые и текстовые, вводимые через диалоговые окна или взятые из локальной базы данных (БД) приложения.

Выходные данные: результирующий контент приложения, полученные со стороны сервера и сохраненный в локальной БД приложения, выводимый на дисплее девайса.

Будет осуществляться добавление следующих данных:

- данные о пользователе
- книги
- местонахождение

Вывод данных по поиску должно будет выдавать следующую информацию:

- личные данные о пользователе
- имеющиеся в наличии книги
- описание важных событий

*Требования к программному обеспечению*

Приложение должно будет реализовано на программе «AndroidStudio».

## **2 Проектирование информационной системы**

### **2.1 Мобильное приложение**

Мобильное приложение - это программа, установленная и работающая на телефоне, коммуникаторе, смартфоне и т. д.

Самыми первыми мобильными устройствами можно считать список контактов в телефоне и сервис для отправки / получения сообщений. Теперь, в связи с развитием сотовой связи и беспроводных технологий (Wi-Fi, WiMax, 4G), мобильные приложения ушли далеко вперед.

Выгоды:

- нет ограничений по SMS-рассылкам по длине, графической и видеоинформации;
- удобство продвижения приложений;
- возможность сбора дополнительных данных (местоположение, язык и т. д.);
- неисчерпаемые возможности для интерактивности.

Недостатки:

- недостаточно распространенные телефоны с поддержкой мобильных приложений;
- пользователь открывает приложение только тогда, когда он хочет, а не, например, когда приходит SMS;
- не очень широкая аудитория (в основном молодежь);
- при этом относительная сложность продвижения (требует особой компетенции, чтобы сделать приложение популярным).

Приложения могут быть предварительно установлены на устройстве во время производства, загружены пользователем через различные платформы распространения программного обеспечения или существовать в формате веб-приложения.

Основными операционными системами, на основе которых создаются мобильные приложения, являются Android, iOS, BlackBerry, HP webOS, Symbian OS, Bada от Samsung и Windows Mobile.

Каналы распространения:

- специализированный портал – AppStore, Android Market;
- через sms с порталов сотовых операторов;
- самостоятельный поиск и скачивание в Интернете.

Мобильные приложения позволяют не только вовлечь участников в интерактивный процесс, но и могут стать рекламным носителем.

### **2.2 Мобильный web сайт и мобильное приложение**

Существует два направления технической реализации проектов для мобильных устройств: мобильный веб-сайт и мобильное приложение. Мобильный веб-сайт будет специализированным веб-сайтом, адаптированным для просмотра и работы на мобильном устройстве.

Сайт может включать в себя интерактивные компоненты с использованием JavaScript, HTML5, новых браузерных API. В этом случае эта реализация называется веб-приложением. Кроме того, чтобы избежать путаницы, любая опция, которая работает внутри браузера мобильного устройства, будет называться веб-сайтом.

Мобильное приложение - это специально разработанное приложение для конкретной мобильной платформы (iOS, Android, Windows Phone). Как правило, приложение разрабатывается на языке высокого уровня и компилируется в собственный код ОС, который обеспечивает максимальную производительность.

Существует третий вариант - мобильное приложение, которое включает в себя компонент браузера. В этом случае часть мобильного приложения чаще всего используется для навигации и интеграции с ОС, а веб-компонент используется для отображения контента. Обычные пользователи часто не могут отличить эту опцию от родного мобильного приложения.

Одним из первых аргументов, которые приводят сторонники приложений – наиболее близкий к ОС и привычный для пользователей интерфейс. Действительно мобильное приложение наиболее тесно интегрировано с платформой и дает реализовать привычный отзывчивый интерфейс. С другой стороны web сайт с помощью хорошего форматирования и использования JavaScript может дать вполне понятный метод взаимодействия. На текущий момент отзыв web сайта значительно уступает приложению, но мощность мобильных устройств продолжает расти и сами браузеры существенно меняются в лучшую сторону. Кроме того, различные версии мобильных ОС могут диктовать свои стандарты, которых приходится придерживаться. При этом некоторые нововведения могут оказаться не совсем понятны обычным пользователям. Существенным в данном случае является то, что наиболее активными пользователями (теми кто выставляет рейтинг и делает комментарии в магазинах приложений) являются те, кто «фанатеет» от последних новшеств мобильной ОС. На это стоит обратить внимание при продвижении проекта – их можно использовать как союзников, помогающих распространению.

### **2.3 Функциональная структура ИС**

Информационная система является элементом большой системы и обеспечивает задачи управления информацией; следовательно, его структура и состав имеют большое значение для эффективной работы всей системы.

Информационная система (ИС) - это система связи для сбора, передачи, обработки информации об объекте, обеспечения сотрудников различного ранга информацией для выполнения ими своих функций управления. Подавляющее большинство информационных систем работают в диалоге с пользователем. В наиболее общем случае типичные программные компоненты, которые являются частью информационной системы, включают в себя: интерактивный ввод-вывод; логика диалога; прикладная логика

обработки данных; логика управления данными; операции с файлами и базами данных.

ИС независимо от сферы их применения, как правило, включают один и тот же набор компонентов: функциональные компоненты; компоненты системы обработки данных; организационные компоненты (рисунок 2.1.).



Рисунок 2.1 - Структура ИС предприятия

Функциональными компонентами являются предметные задачи, методы и алгоритмы их решения. Задача включает в себя набор исходных данных в виде атрибутов, связей, показателей, базы данных.

Компоненты системы обработки данных представляют собой набор систем, которые обеспечивают функционирование всех структур ИР.

Организационные составляющие - состав и структура организаций, обеспечивающих эффективное функционирование экономического объекта.

Состав функций, реализованных в ИС, подразделяется на информационные и управляющие функции (рисунок 2.2). В свою очередь, имеются: информационные функции централизованного управления и функции вычислительных и логических операций. Информационные функции должны включать:

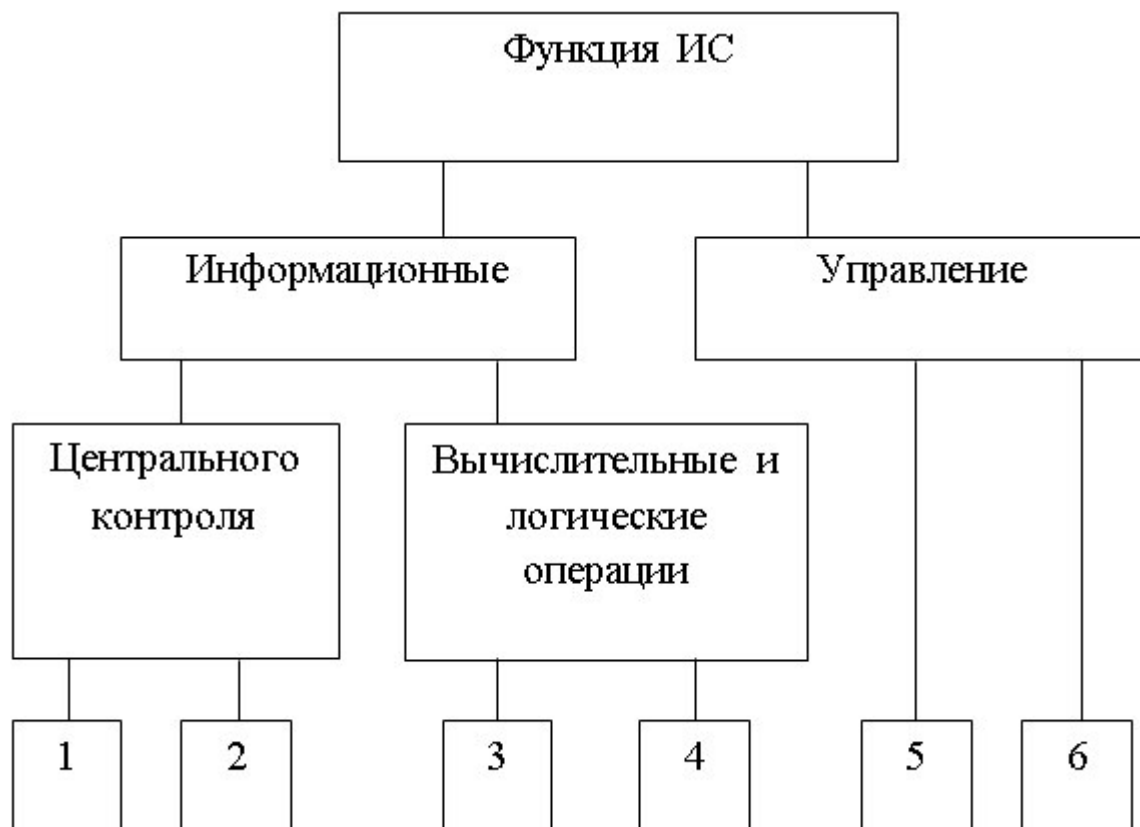


Рисунок 2.2 – Функциональная структура ИС

Функциональная структура ИС - это набор функциональных подсистем, наборов задач и процедур обработки информации, которые реализуют функции (задачи) системы и отражают их иерархию. Любая сложная функция делится на ряд небольших задач и передается непосредственному исполнителю.

Согласно теории, известно, что вся информационная система охватывает такие задачи, как сбор информации, ее хранение, обработка и выдача уже новой информации. Системы информационных систем делятся на функциональные подсистемы, которые зависят от группы факторов, которые связаны с предметной областью и целями, которые были установлены для информационной системы. Каждая из этих подсистем играет свою роль и решает определенные задачи. Кроме того, каждая задача имеет свой конечный результат и решение.

Таким образом, для правильного построения и реализации функциональной подсистемы вам необходимо спроектировать саму структуру существующей системы. Это необходимо для того, чтобы правильно определить взаимосвязь между элементами и продемонстрировать, какую задачу решает разработанное мобильное приложение в этой системе.

На рисунке 2.3 представлена функциональная структура информационной системы.





Рисунок 2.3 – Функциональная структура информационной системы

Подсистема «Программист-разработчик» состоит из программистов, которые работают над приложением. Они являются главным связующим звеном между мобильным приложением и его пользователями.

Функции данной подсистемы:

- поддерживать работу мобильного приложения;
- улучшать и выпускать новые улучшенные версии приложения.

Подсистема «Клиенты» состоит из пользователей мобильного приложения. Клиенты являются основой системы, так как именно они имеют напрямую влияние на успешность приложения. Следовательно, клиенты – это основные пользователи приложения и они располагают следующими действиями:

- открытие приложения с их мобильного телефона;
- регистрация в данном приложении;
- возможность просмотра новостей не только университета, но и новостей в мире;

- просмотр имеющейся литературы;
- добавление необходимых книг для изучения материала;

Подсистема «Мобильное приложение» - это основной источник потенциальных пользователей. В данной подсистеме содержатся:

- интерфейс и различные элементы функционирования отвечает за кнопки, вкладки и т.д.;
- обратная связь дает возможность пользователю оставить комментарии для улучшенной работы приложения;
- база данных, а именно сервер, хранится вся информация и сведения приложения.

Исследовав всю функциональную структуру информационной системы, можно прийти к выводу, что разрабатываемые системы играют важную роль для ее участников. Если даже одна система будет удалена, система не будет работать корректно, что приведет к большой потере потенциальных пользователей.

## **2.4 Разработка UML-диаграмм**

Приложение для ОС Android состоит из набора активностей, каждой из которых соответствует экран приложения. Каждая активность представлена в проекте классом, реализованном на языке Java, хранящемся в одноименном файле с расширением .java. Каждой активности соответствует xml файл-описание. В xml-файле описано в виде xml кода расположение визуализируемых объектов. При запуске активности система Android автоматически распознает размер экрана мобильного устройства и приводит выводимый контент в соответствие с разметкой, описанной в xml файле. Таким образом, одна и та же активность будет выглядеть одинаково независимо от диагонали используемого устройства. Также, для каждого приложения Android должен существовать xml файл, в котором в виде xml кода будут прописаны минимальные требования к системе, а также активность, вызываемая при запуске приложения.

Приложение работает с базой данных MySQL, которая находится на сервере. MySQL использует парадигму клиент-сервер. Для связи между клиентом и сервером используются запросы PHP и JSON. Для получения данных, отправки формы, подключения к базе данных были созданы отдельные файлы PHP, что позволяет легко вносить изменения в базу данных. Такой подход сокращает время отклика и упрощает работу разработчиков.

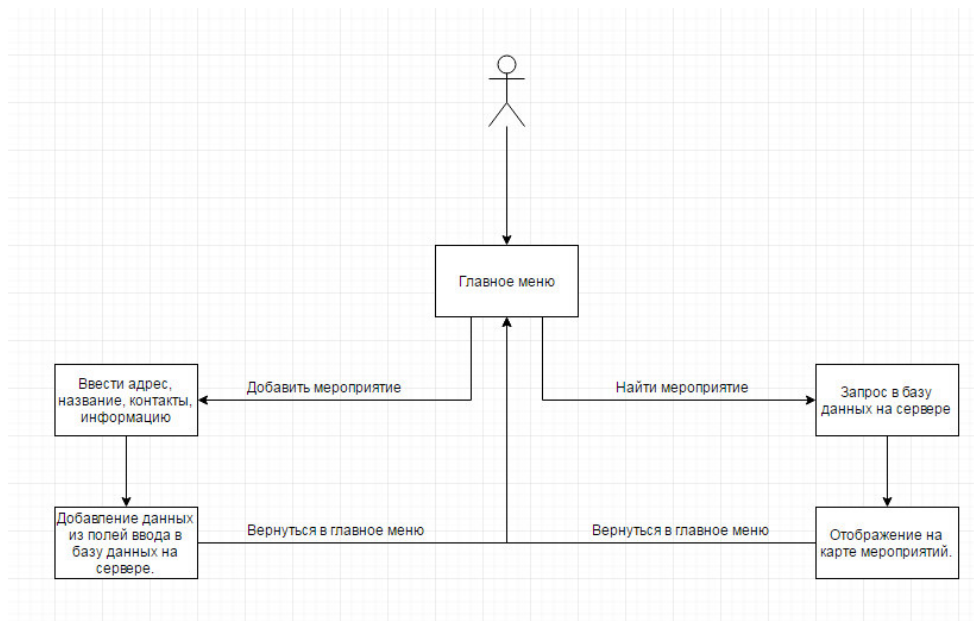


Рисунок 2.4 – Пошаговая схема работы приложения

Таким образом, пользователь может добавить событие или посмотреть на карту. При выборе добавления события загружается действие «Добавить событие», которое обращается к базе данных и отправляет данные в базу данных. При выборе поиска события загружается действие «Найти события», которое обращается к базе данных и считывает данные из столбцов, отображает события. Выход из этих действий осуществляется кнопкой Назад.

## 2.5 Бизнес-процессы ИС

Для построения функциональной модели данной предметной области использовалась методология САДТ, отражающая функциональную структуру объекта, то есть выполняемые им действия и связи между этими действиями. Графические блоки и дуги диаграммы отражают функциональный блок, а интерфейсы входы / выходы представлены дугами. Взаимодействие блоков друг с другом выражается с помощью интерфейсных дуг, описывающих, когда и над какой функцией выполняется и как она управляется. На рисунке 1 показан блок А0, представляющий всю систему, имя блока - это имя всей системы. Информация о книгах и читателях базы данных обрабатывается по этой схеме, а результатом работы всей системы является выдача отчетов о результатах ее деятельности. Управление системой осуществляет библиотекарь, который вводит поступающую информацию в базу данных и управляет ею через автоматизированную систему.



Рисунок 2.5 - SADT диаграмма - Блок А0. 1

Следующим шагом построения SADT диаграммы служит более детальное рассмотрение системы путем декомпозиции всей системы на более мелкие подзадачи. На рисунке 2.6 схематично в качестве блоков показаны основные процессы системы:

Блок А1 - Регистрация читателя. В этом блоке заносится информация в базу данных о новых читателях. Входящей информацией служат индивидуальные данные читателя, вносимые библиотекарем, в процессе регистрации читателю присваивается уникальный номер. Результатом работы этого процесса является добавление новой записи в таблицу зарегистрированных читателей.

Блок А2 - Регистрация книг. Предназначен для ввода поступивших книг в библиотеку, при регистрации книг так же присваивается уникальный номер. После выполнения процесса, зарегистрированные книги готовы к выдаче читателям.

Блок А3 - Выдача книг. При выполнении этого процесса библиотекарем заносится запись о выдаче книги читателю, в которой указывается, кто и какую книгу взял, и отмечается дата занесения записи.

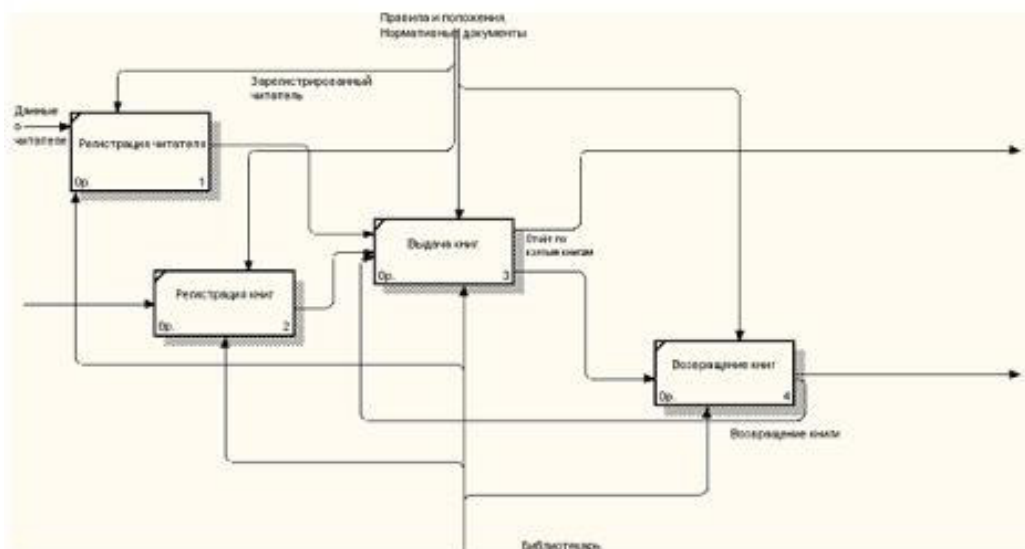


Рисунок 2.6 - Схема подзадачи системы.

## 3 Разработка программного обеспечения ИС

### 3.1 Обоснование выбора среды разработки

**Android Studio.** Существует несколько самых распространенных операционных систем для смартфонов, такие как, IOS, Android, WindowsPhone, BlackBerry, Simbian. Для создания моего приложения выбран Android, потому что:

1. Android – операционная система с открытым исходным кодом.
2. Распространенность ОС Android, указана на рисунке 3.1
3. Доступ к разработке любому пользователю.
4. Абсолютно бесплатная для разработки.

#### Доля устройств на различных ОС

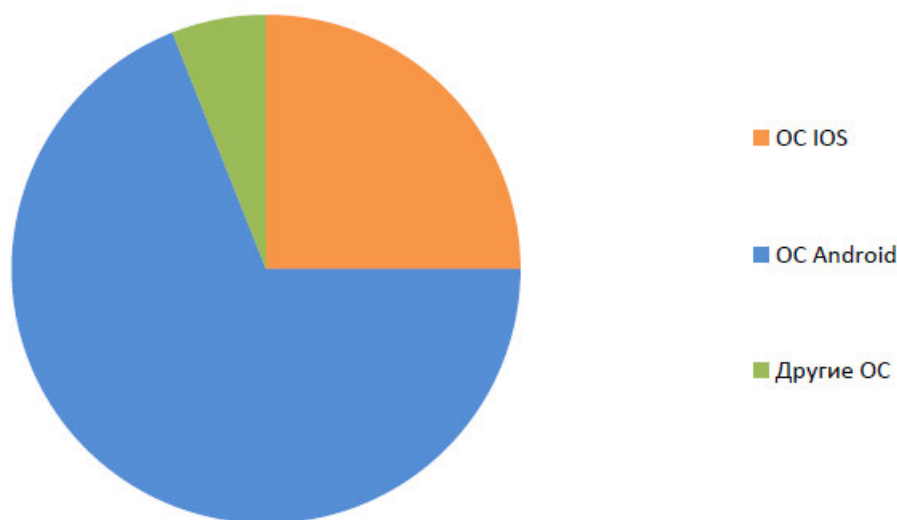


Рисунок 3.1 – Доля устройств на различных ОС

Android Studio — это интегрированная среда разработки (IDE) для работы с платформой Android, анонсированная 16 мая 2013 года на конференции Google I/O.[7]

IDE находилась в свободном доступе начиная с версии 0.1, опубликованной в мае 2013, а затем перешла в стадию бета-тестирования, начиная с версии 0.8, которая была выпущена в июне 2014 года. Первая стабильная версия 1.0 была выпущена в декабре 2014 года, тогда же прекратилась поддержка плагина Android DevelopmentTools (ADT) для Eclipse.

AndroidStudio, основана на программном обеспечении IntelliJ IDEA от компании JetBrains, официальное средство разработки Android приложений. Данная среда разработки доступна для Windows, OS X и Linux.

За счет своей лаконичности по сравнению с XML, формат JSON может быть более подходящим для сериализации сложных структур. Если говорить о

веб-приложениях, в таком ключе он уместен в задачах обмена данными как между браузером и сервером (AJAX), так и между самими серверами (программные HTTP-интерфейсы).[12]

Поскольку формат JSON является подмножеством синтаксиса языка JavaScript, то он может быть быстро десериализован встроенной функцией eval(). Кроме того, возможна вставка вполне работоспособных JavaScript-функций. В языке PHP, начиная с версии 5.2.0, поддержка JSON включена в ядро в виде функций json\_decode() и json\_encode(), которые сами преобразуют типы данных JSON в соответствующие типы PHP и наоборот.

В качестве значений в JSON могут быть использованы:

- Объект — это неупорядоченное множество пар ключ: значение, заключённое в фигурные скобки «{ }». Ключ описывается строкой, между ним и значением стоит символ «:». Пары *ключ-значение* отделяются друг от друга запятыми.

- Массив (одномерный) — это упорядоченное множество значений. Массив заключается в квадратные скобки «[ ]». Значения разделяются запятыми.

- Число.

- Литералы *true*, *false* и *null*.

- Строка — это упорядоченное множество из нуля или более символов юникода, заключенное в двойные кавычки. Символы могут быть указаны с использованием escape-последовательностей, начинающихся с обратной косой черты «\» (поддерживаются варианты \", \\, \/, \t, \n, \r, \f и \b), или записаны шестнадцатеричным кодом в кодировке UTF-8 в виде \uFFFF.

Технология Java включает в себя язык программирования Java, средства трансляции исходного текста программы - исходного кода - в специальную форму, пригодную для исполнения компьютером, и средства исполнения Java-программ на различных платформах, то есть в различных операционных системах и на различном аппаратном обеспечении. Основная особенность JAVA-технологии в том, что преобразованная на этапе трансляции в специальный код Java-программа полностью "машинонезависима".

Если исполняемый код, полученный из программ на других распространённых языках, обычно не пригоден для исполнения компьютером "другой платформы", то к исполняемому коду Java такое ограничение не относится. Правда, необходимо, чтобы для "целевой платформы" имелась реализация так называемой Java-машины - среды исполнения JAVA-программ.

JAVA-технологии, активно продвигаемые компанией SUN, получили широкое распространение (далеко не только в web-решения). А платформонезависимость Java, позволившая интегрировать средства исполнения Java-программ в браузеры, работающие в самых разных операционных системах, определила распространение Java в качестве элемента Web-технологий.

JAVA используется для создания сложных интерактивных элементов, связанных с web-сайтом. Например, на Java реализуются сложные инструменты для работы с базами данных, размещёнными в Web. Или графические интерфейсы, требующие вывода сложных интерактивных элементов. И, конечно, многое другое, от сетевых шахматных программ до средств редактирования звуковых файлов.

**SQLite** — легковесная встраиваемая реляционная база данных. Исходный код библиотеки передан в общественное достояние. В 2005 году проект получил награду Google-O'Reilly Open Source Awards<sup>[2]</sup>.

*Дизайн.* Слово «встраиваемый» означает, что SQLite не использует парадигму клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а предоставляет библиотеку, с которой программа компонуется и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором исполняется программа. Простота реализации достигается за счёт того, что перед началом исполнения транзакции запись весь файл, хранящий базу данных, блокируется; ACID-функции достигаются в том числе за счёт создания файла журнала.

Несколько процессов или потоков могут одновременно без каких-либо проблем читать данные из одной базы. Запись в базу можно осуществить только в том случае, если никаких других запросов в данный момент не обслуживается; в противном случае попытка записи оканчивается неудачей, и в программу возвращается код ошибки. Другим вариантом развития событий является автоматическое повторение попыток записи в течение заданного интервала времени.

В комплекте поставки идёт также функциональная клиентская часть в виде исполняемого файла `sqlite3`, с помощью которого демонстрируется реализация функций основной библиотеки. Клиентская часть работает из командной строки, позволяет обращаться к файлу БД на основе типовых функций ОС.

Благодаря архитектуре движка возможно использовать SQLite как на встраиваемых системах, так и на выделенных машинах с гигабайтными массивами данных.

*Ограничения.* Старые версии SQLite были спроектированы без каких-либо ограничений, единственным условием было то, чтобы база данных умещалась в памяти, в которой все вычисления производились при помощи 32-разрядных целых чисел. Это создавало определённые проблемы. Из-за того, что верхние пределы не были определены и соответственно должным образом протестированы, то частенько наружу вылезали ошибки при использовании SQLite в достаточно экстремальных условиях. И поэтому, в



новых версиях SQLite были введены пределы, которые теперь проверяются вместе с общим набором тестов.

Это может быть полезным, если SQLite используется в веб-приложениях, так как уменьшенные пределы могут предотвратить DoS-атаки со стороны недоверяемых внешних клиентов.

*Использование.* Сама библиотека SQLite написана на C; существует большое количество привязок к другим языкам программирования, в том числе C++, Java, C#, VB.NET, Python, Perl, PHP, Tcl (средства для работы с Tcl включены в комплект поставки SQLite), Ruby, Haskell, Scheme, Smalltalk, Lua и Parser, а также ко многим другим. Полный список существующих средств размещён на странице проекта<sup>[3]</sup>.

Простота и удобство встраивания SQLite привели к тому, что библиотека используется в браузерах, музыкальных плеерах и многих других программах.

В частности, SQLite используют:

- Среда для запуска приложений Adobe Integrated Runtime, правда частично;

- Gears;

- Фреймворк Qt;

- Платформа XUL на движке Gecko 1.9, XULRunner 1.9 и, потенциально, все приложения, основанные на этой платформе, в том числе Mozilla Firefox (начиная с версии 3.0) и Mozilla Thunderbird (начиная с версии 3.0). В качестве примеров XUL-приложений можно привести Songbird и SQLite Manager<sup>[4]</sup>.

- Skype<sup>[5]</sup>;

- Некоторые модели GPS-навигаторов Garmin

Многие программы поддерживают SQLite в качестве формата хранения данных (особенно в Mac OS и iPhone OS, Android), в том числе:

- 1С:Предприятие 7.7 (с помощью внешнего компонента<sup>[6]</sup>);

- Banshee;

- F-Spot;

- Gajim;

- Miranda IM (с помощью плагина драйвера базы данных<sup>[7]</sup>);

- Safari;

- Opera (Начиная с версии 10.50);

- AIMP;

- XnView;

- Mozilla Firefox

- qutIM

SQLite доступен на любом Android-устройстве, его не нужно устанавливать отдельно.

SQLite поддерживает типы TEXT (аналог String в Java), INTEGER (аналог long в Java) и REAL (аналог double в Java). Остальные типы следует конвертировать, прежде чем сохранять в базе данных. SQLite сама по себе не

проверяет типы данных, поэтому вы можете записать целое число в колонку, предназначенную для строк, и наоборот. Представлены в таблице 3.1.

Таблица 3.1 – Тип данных в SQLite

Тип	Описание
NULL	пустое значение
INTEGER	целочисленное значение
REAL	значение с плавающей точкой
TEXT	строки или символы в кодировке
NUMERIC	здесь можно хранить булевы значения, а также время и дату
BLOB	бинарные данные

Работа с SQLite напрямую теперь считается устаревшим подходом. На Google I/O 2017 был представлен новый механизм доступа к базе данных через специальную обёртку Room

Работа с базой данных сводится к следующим задачам:

- Создание и открытие базы данных
- Создание таблицы
- Создание интерфейса для вставки данных (insert)
- Создание интерфейса для выполнения запросов (выборка данных)
- Закрытие базы данных

С помощью утилиты ADB можно запустить SQLite на эмуляторе и работать с базами данных напрямую.

Напомню, что можно активировать команду **sqlite3** для одной из перечисленных баз данных, введя следующую команду:

```
#sqlite3 /data/data/com.android.providers.contacts/databases/contacts.db
```

Для завершения работы с **sqlite3** напишите:

```
sqlite> .exit
```

Обратите внимание: приглашение для adb - это #, а приглашение для sqlite3 - это **sqlite>**.

Перечислим некоторые важные команды. Чтобы просмотреть список таблиц:

```
sqlite> .tables
```

Быстрый доступ к главной таблице:

```
SELECT name FROM sqlite_master  
WHERE type IN ('table', 'view') AND name NOT LIKE 'sqlite %'  
UNION ALL  
SELECT name FROM sqlite_temp_master  
WHERE type IN ('table', 'view')  
ORDER BY 1
```

Таблица **sqlite\_master** - это главная таблица (master table), в которой отслеживаются таблицы и виды, содержащиеся в базе данных. Следующая

команда распечатывает инструкцию **create** для таблицы **people**, находящейся в базе данных **contacts.db**:

```
.schema people
```

Это один из способов, позволяющих узнать названия всех столбцов, которые содержатся в таблице базы данных. Можно скопировать базу данных на локальный компьютер и изучать её в более комфортных условиях. Чтобы переместить файл **contacts.db**, можно дать следующую команду:

```
adb pull /data/data/com.android.providers.contacts/databases/contacts.db  
d:/somelocaldir/contacts.db
```

**Язык программирования JAVA.** На сегодняшний момент язык Java является одним из самых распространенных и популярных языков программирования. Первая версия языка появилась еще в 1996 году в недрах компании Sun Microsystems, впоследствии поглощенной компанией Oracle. Java задумывался как универсальный язык программирования, который можно применять для различного рода задач. И к настоящему времени язык Java проделал большой путь, было издано множество различных версий. Текущей версией является Java 12, которая вышла в марте 2019 года. А Java превратилась из просто универсального языка в целую платформу и экосистему, которая объединяет различные технологии, используемые в целого ряда задач: от создания десктопных приложений до написания крупных веб-порталов и сервисов. Кроме того, язык Java активно применяется для создания программного обеспечения для целого ряда устройств: обычных ПК, планшетов, смартфонов и мобильных телефонов и даже бытовой техники. Достаточно вспомнить популярность мобильной ОС Android, большинство программ для которой пишутся именно на Java.

**Особенности Java.** Ключевой особенностью языка Java является то, что его код сначала транслируется в специальный байт-код, независимый от платформы. А затем этот байт-код выполняется виртуальной машиной JVM (Java Virtual Machine). В этом плане Java отличается от стандартных интерпретируемых языков как PHP или Perl, код которых сразу же выполняется интерпретатором. В то же время Java не является и чисто компилируемым языком, как C или C++.

Подобная архитектура обеспечивает кроссплатформенность и аппаратную переносимость программ на Java, благодаря чему подобные программы без перекомпиляции могут выполняться на различных платформах - Windows, Linux, Mac OS и т.д. Для каждой из платформ может быть своя реализация виртуальной машины JVM, но каждая из них может выполнять один и тот же код.

Java является языком с Си-подобным синтаксисом и близок в этом отношении к C/C++ и C#. Поэтому, если вы знакомы с одним из этих языков, то овладеть Java будет легче.

Еще одной ключевой особенностью Java является то, что она поддерживает автоматическую сборку мусора. А это значит, что вам не надо

освобождать вручную память от ранее использовавшихся объектов, как в C++, так как сборщик мусора это сделает автоматически за вас.

Java является объектно-ориентированным языком. Он поддерживает полиморфизм, наследование, статическую типизацию. Объектно-ориентированный подход позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений.

Язык Java активно используется для создания мобильных приложений под операционную систему Android. При этом программы компилируются в нестандартный байт-код, для использования их виртуальной машиной Dalvik (начиная с Android 5.0 Lollipop виртуальная машина заменена на ART). Для такой компиляции используется дополнительный инструмент, а именно Android SDK (Software Development Kit), разработанный компанией Google.

Разработку приложений можно вести в среде Android Studio, NetBeans, в среде Eclipse, используя при этом плагин Android Development Tools (ADT), или в IntelliJ IDEA. Версия JDK при этом должна быть 5.0 или выше.

8 декабря 2014 года Android Studio признана компанией Google официальной средой разработки под ОС Android.

### **3.2 Структура программного обеспечения**

Из определения в частности следует, что в информационной системе имеется два компонента: программное обеспечение и электронное информационное хранилище.

Любая информационная система рассчитана на использование ее в какой-либо профессиональной области. Значит ИС рассчитана на взаимодействие, с какими либо пользователями. Причем под пользователями в общем случае следует понимать не только людей, но и другие информационные системы, с которыми данная ИС обменивается информацией.

Если программа взаимодействует с пользователем и данными, значит, какая-то ее часть (логика) отвечает за это взаимодействие. С другой стороны, как мы впоследствии будем неоднократно убеждаться, структура хранения данных практически никогда не совпадает со структурой данных, представляемых пользователю. Следовательно, в программном обеспечении должно быть предусмотрено преобразование информации из одного формата к другому и обратно. Вот эта часть программного обеспечения, которую мы выделили чисто логически и принято называть бизнес логикой.

Формально информационные системы можно разделить на автономные и сетевые. Причем к сетевым информационным системам мы относим и такие, которые эпизодически синхронизируют свои информационные хранилища с другими хранилищами посредством каких-либо каналов связи (в том числе и посредством переносных устройств внешней памяти). Спрос на автономные ИС не велик, в силу почти полного отсутствия несетевых компьютеров.

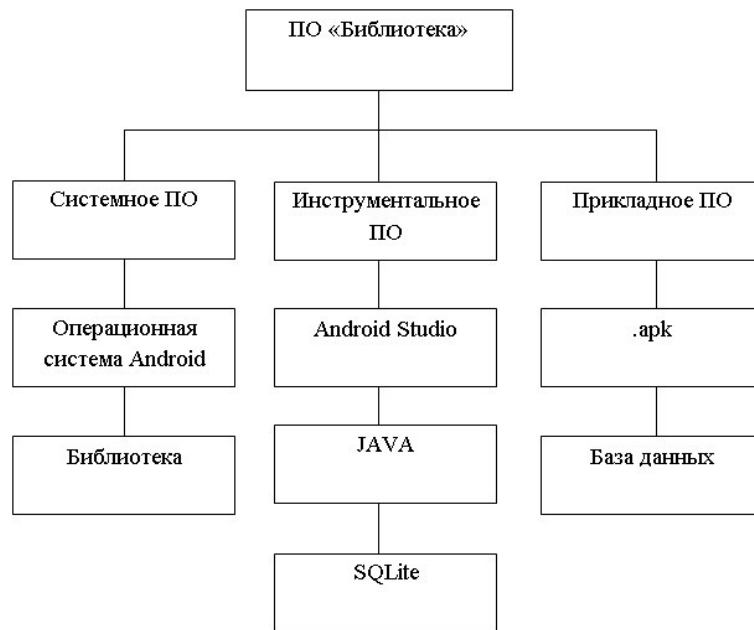


Рисунок 3.2 – Структура программного обеспечения

### 3.3 Разработка интерфейса приложения (прототип)

Для создания макета был использован сервис Mockplus. Данное приложение имеет одну существенную для современного мира особенность— это скорость и простота в выполнении. Как правило, Mockplus используется для desktopbasedapplication, но он также может быть легко использован для мобильных и веб-проектов. Чистый интерфейс в разы повышает эффективность работы, потому что проектировщик, во время работы, сосредоточивает свое внимание на самой конструкции и в меньшей степени на функциях. При помощи различных шаблонов был разработан макет, представленный на рисунках 3.9, 3.10, 3.11.

При запуске данного мобильного приложения пользователь увидит главную страницу (рис 3.2), на которой он может зайти или зарегистрироваться, а также посмотреть по карте местонахождение библиотеки.



Рисунок 3.3 – Меню мобильного приложения

Нажав на кнопку «Регистрация» пользователь попадает в меню, где должен заполнить свои данные, по желанию добавить фотографию, чтоб пользоваться приложением.

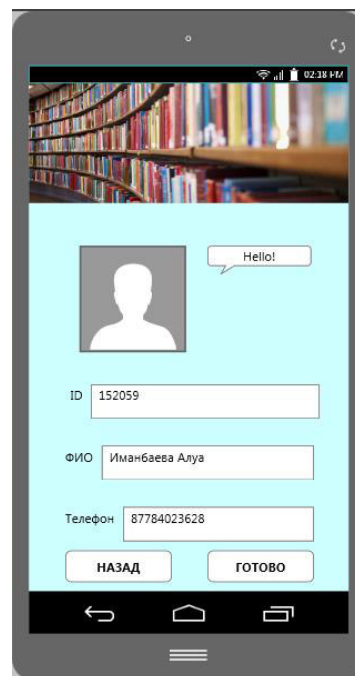


Рисунок 3.4 – Данные пользователя

При нажатии на кнопку «Вход» пользователь должен ввести свои данные (IDи пароль), чтоб зайти и пользоваться приложением.

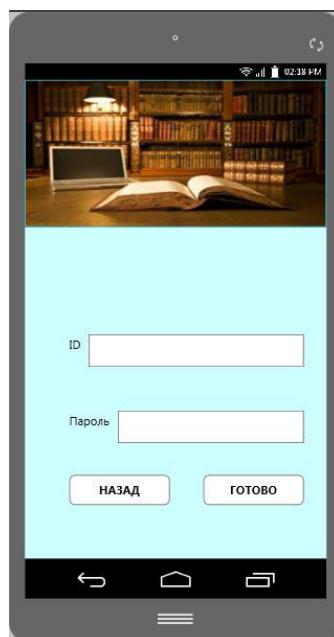


Рисунок 3.5 – Вид кнопки «Вход»

### 3.4 Описание программы

#### 3.4.1 Общие сведения

Для начала рассмотрим все имеющиеся модули, разрабатываемые в проекте. Программа реализована в программной среде Android Studio.

- SplashActivity;
- MainActivity;
- LibraryFragment;
- NewsFragment;
- ProfileFragment;
- SettingsFragment;
- LoginActivity;
- AutorizationActivity.

#### 3.4.2 Функциональное назначение

Данный программный продукт служит для хранения книг, как методических указаний, так и обычной литературы разных жанров. Также просмотр новостей, имеющихся в университете.

Модуль SplashActivity выводит на экран мобильного телефона загрузку в виде круга и двигающейся в нем волны, которая задерживается на 5 секунд и дальше переходит в главное меню.

Модуль MainActivity отображает окно работы с программой в виде шторки, куда можно перейти в другие вкладки: вкладка профиля, вкладка библиотеки, вкладка настроек, вкладка новостей и вкладка выхода.

Модуль ProfileFragment выводит на экран окно авторизации пользователя. Соответственно авторизация происходит путем ввода логина и пароля. Регистрация происходит через SMS-регистрацию.

Модуль LibraryFragment выводит на экран вкладку библиотеку, где имеется список литературы

Модуль NewsFragment отображает на экране вкладку новости, где рассказывается о интересных событиях в университете.

Модуль SettingsFragment выводит на экран вкладку настройки, где пользователь имеет возможность поменять фон и язык.

Модуль LoginActivity позволяет войти пользователю, прописав свое имя и пароль.

Модуль AutorizationActivity позволяет новым пользователям зарегистрироваться в данном приложении. Данное окно появляется при выходе пользователя из приложения.

### 3.4.3 Описание логической структуры

Логическая структура модуля SplashActivityс привязкой к строкам текста приведено в таблице 3.2.

Таблица 3.2 – Логическая структура модуля SplashActivity

12-33	Создаем класс SplashActivity и объявляем переменную mWaveLoadingView
17-31	Вызов метода onCreate
20-22	Инициализация и настройка WaveLoadingView
23-31	Вызов метода задержки на 5 секунд с последующим переходом в активность MainActivity

Логическая структура модуля MainActivityс привязкой к строкам текста приведено в таблице 3.3.

Таблица 3.3 – Логическая структура модуля MainActivity

39-213	Создаем класс MainActivity и объявляем переменные
54-120	Вызов метода onCreate
58-76	Вызов метода initialization, установка боковой панели и данных в нее и открытие фрагмента «библиотека»
77-117	Установка логики поведения кнопок бокового меню
131-158	Создание методов, позволяющих открыть фрагменты в главном меню
168-177	Создание метода initialization, позволяющего инициализировать переменные



### Продолжение таблицы 3.3

179-212	Создание метода <code>setLabel</code> , позволяющего прописывать текстовые значения в зависимости от установленного языка интерфейса
---------	--

Логическая структура модуля `LibraryFragments` привязкой к строкам текста приведено в таблице 3.4.

Таблица 3.4 – Логическая структура модуля `LibraryFragment`

42-357	Создание фрагмента <code>LibraryFragment</code>
92-139	Вызов метода <code>onStart</code> , инициализация элементов, таких как <code>gridView</code> , <code>gridViewAdapter</code> , для отображения элементов плиткой, добавление кнопок для загрузки и удаления информации в базу данных, устанавливаем список <code>books</code> в <code>gridView</code>
157-198	Создаем класс <code>GridViewAdapter</code> для обеспечения заполнения списка <code>gridView</code> элементами из списка <code>books</code>
200-237	Метод <code>createBook</code> позволяет создавать записи в базе данных
239-242	Метод <code>deleteBook</code> позволяет удалить записи из базы данных
244-288	Метод <code>displayDataBaseInfo</code> позволяет получать данные из базы данных и записывает их в лист объектов класса <code>Book</code> , который мы декларировали в начале фрагмента: <code>List&lt;Book&gt; books</code>
289-356	Метод <code>setGenre</code> создается для установления данных в <code>gridView</code> в зависимости от выбранного в выпадающем списке жанра

Логическая структура модуля `NewsFragments` привязкой к строкам текста приведено в таблице 3.5.

Таблица 3.5 – Логическая структура модуля `NewsFragment`

59-64	Описание метода <code>onStart</code> , инициализация переменной <code>mWebView</code> и вызов метода <code>openWebView</code>
80-86	Создание метода <code>openWebView</code> , установка настроек и загрузка страницы URL

Логическая структура модуля `ProfileFragment` с привязкой к строкам текста приведено в таблице 3.6.

Таблица 3.6 – Логическая структура модуля `ProfileFragment`

16-87	Создание фрагмента <code>ProfileFragment</code> и объявление переменных
61-70	Инициализация основных объектов: изображение и поля для отображения текста; запись данных в эти объекты

Логическая структура модуля SettingsFragment с привязкой к строкам текста приведено в таблице 3.7.

Таблица 3.7 – Логическая структура модуля SettingsFragment

25-157	Создание фрагмента SettingsFragment
69-89	Создание метода onStart, инициализация объектов интерфейса: текстовое поле, спиннер и кнопка, вызов метода setLanguage, установка языка фрагмента в зависимости от условия
91-133	Создание метода setLanguage, установка поведения программы при нажатии кнопки в зависимости от выбранного поля выпадающего меню

Логическая структура модуля LoginActivity с привязкой к строкам текста приведено в таблице 3.8.

Таблица 3.8 – Логическая структура модуля LoginActivity

15-55	Создание активности LoginActivity
21-54	В методе onCreate инициализируем объекты и переменные, устанавливаем по нажатию кнопки переход в главное меню, в случае неправильно введенных данных вызов всплывающего меню, в зависимости от установленного в настройках языка интерфейса, сообщаем о неверно введенных данных

Логическая структура модуля AuthorizationActivity с привязкой к строкам текста приведено в таблице 3.9.

Таблица 3.9 – Логическая структура модуля AuthorizationActivity

15-73	Создание активности AuthorizationActivity, в данной активности создаем нового пользователя
21-49	В методе onCreate инициализируем объекты и переменные, устанавливаем по нажатию кнопки переход в главное меню
51-72	Метод fillStrings устанавливает в зависимости от языка интерфейса текстовые значения

### 3.4.5 Используемые технические средства

В ходе разработки программного обеспечения были использованы следующие технические средства: ноутбук Lenovo B50-10, 512 Мб ОЗУ, также используется лазерный принтер HP LaserJet 1020.

### 1.5 Инструкция пользователю

Данное приложение создано для выполнения работ на мобильных телефонах с операционной системой Android 3.0 и версии выше.

#### *Установка приложения*

Для того, чтобы установить приложение «Библиотека» на телефон необходимо:

1. Нажмите на кнопку *Установить* для загрузки приложения.
2. После того, как приложение загрузилось, нажмите на кнопку *Открыть*. Приложение начнет загружаться (Рисунок 3.6)

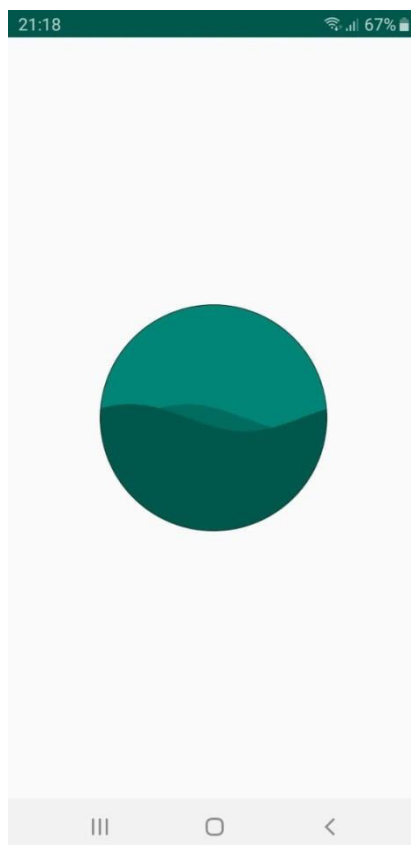


Рисунок 3.6 – Запуск приложения

Далее, щелкнув на иконку, которая расположена в меню мобильного телефона, откроется окно авторизации, в котором пользователю необходимо ввести свои данные (рисунок 3.7).

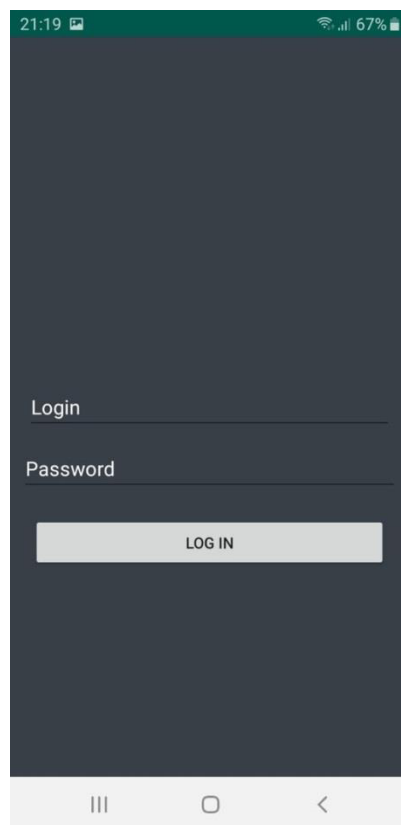


Рисунок 3.7 – Окно Входа

Если вы открыли данное приложение первый раз, то нажав кнопку «Регистрация», можете зарегистрироваться, прописав свое Имя и Пароль (рисунок 3.8).

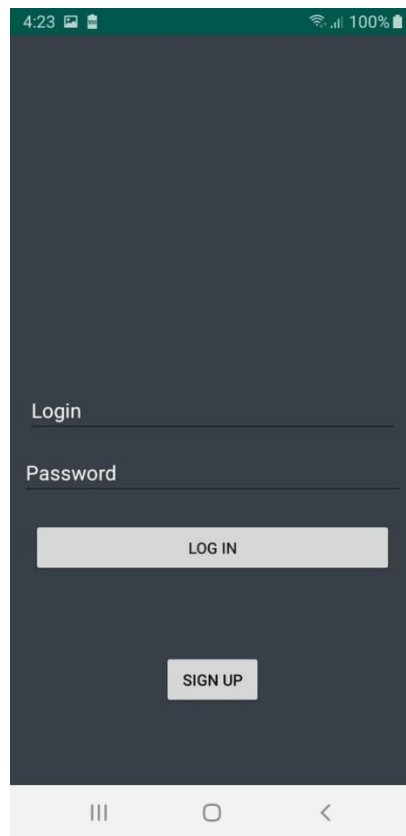


Рисунок 3.8 – Окно регистрации пользователя

Для того чтобы начать работу в приложении, пользователю необходимо ввести свои данные в окне авторизации и нажать кнопку Войти. После нажатия на кнопку начинается загрузка главного окна, в котором отображено меню со следующими пунктами (рисунок 3.9) [13]:

- Профиль;
- Библиотека;
- Новости;
- Настройки;
- Выход.

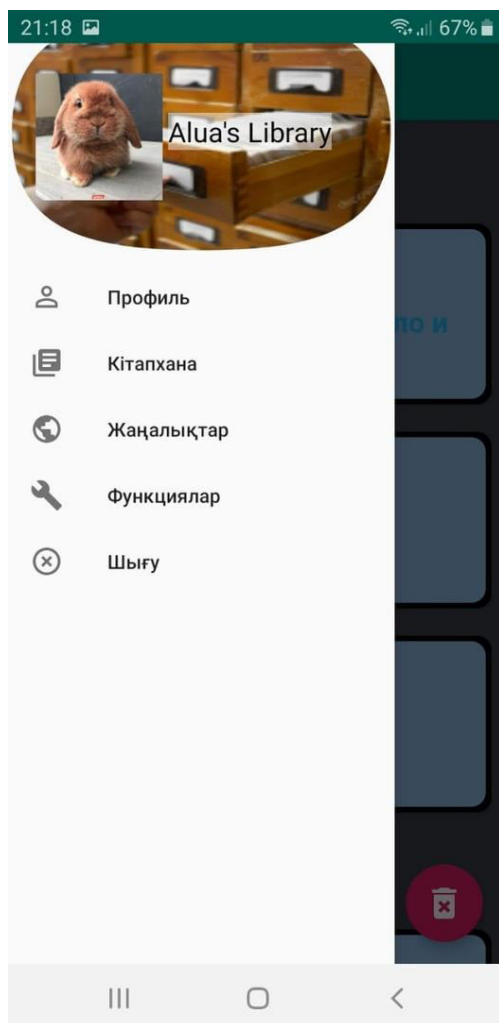


Рисунок 3.9 – Отображение главного меню

Пункт «Профиль» выводит на экран имя пользователя и его фотографию (Рисунок 3.10).

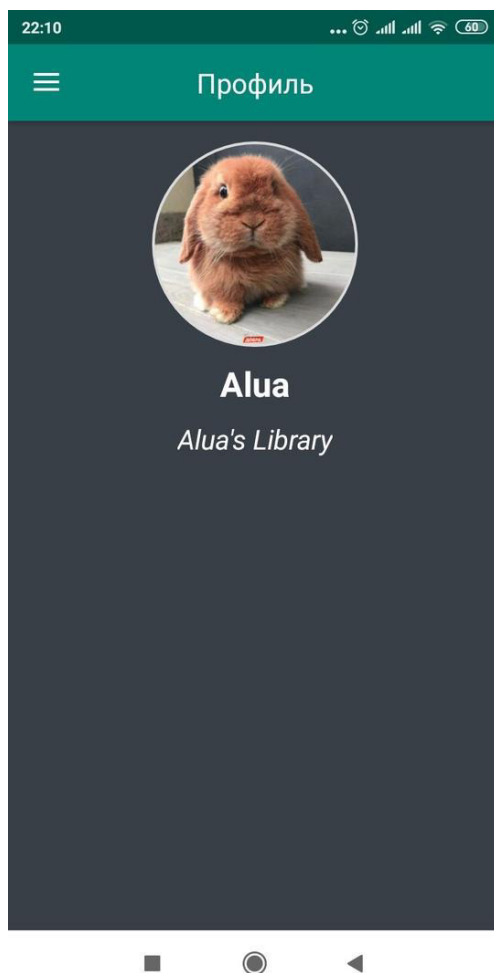


Рисунок 3.10 – Окно профиля

Пункт «Библиотека» выводит на экран доступную литературу. Нажав на необходимую книгу можно перейти к тексту для чтения. Также для удобства каждая книга разделена на главы и жанры (Рисунок 3.11).

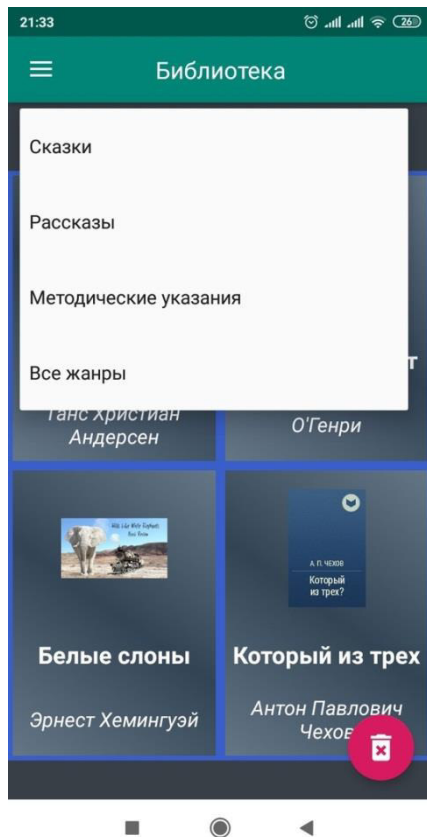
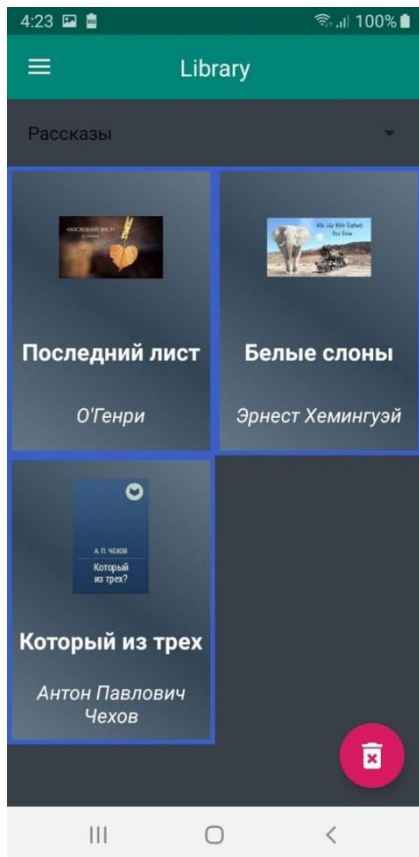


Рисунок 3.11 – Окно Библиотека



Пункт новостей отображает на экран все происходящие новости в университете. Там публикуются не только новости из ВуЗа, но и происходящее в мире в целом (рисунок 3.12).

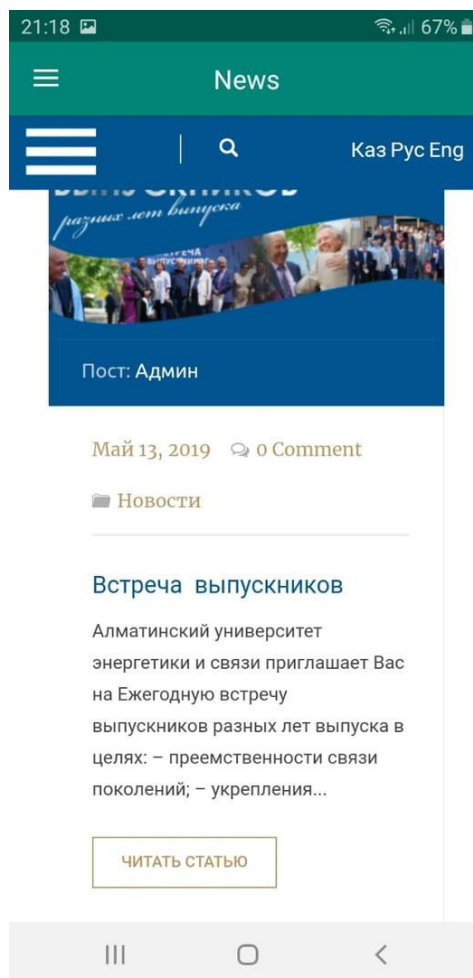


Рисунок 3.12 – Окно новостей

В пункте «Настройки» имеется возможность смены языка на русский, казахский и английский (рисунок 3.13).

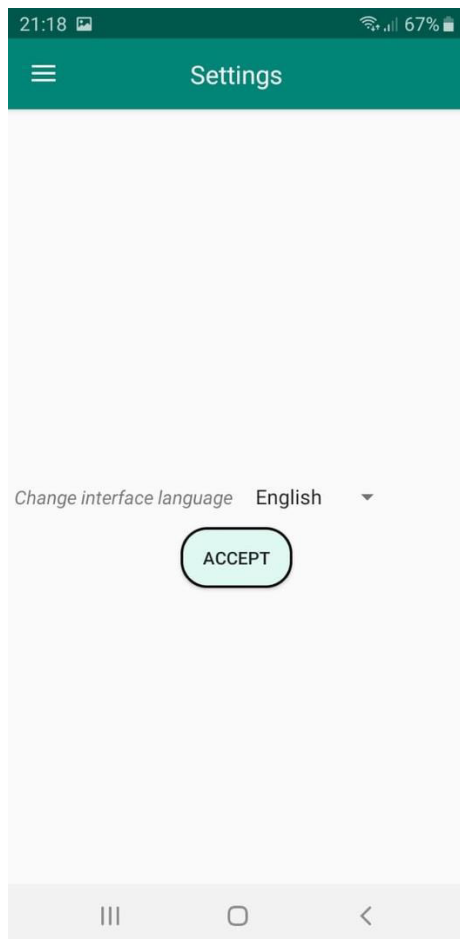


Рисунок 3.13 – Меню настроек

Пункт «Выход» необходим для выхода пользователя из приложения (рисунок 3.14).

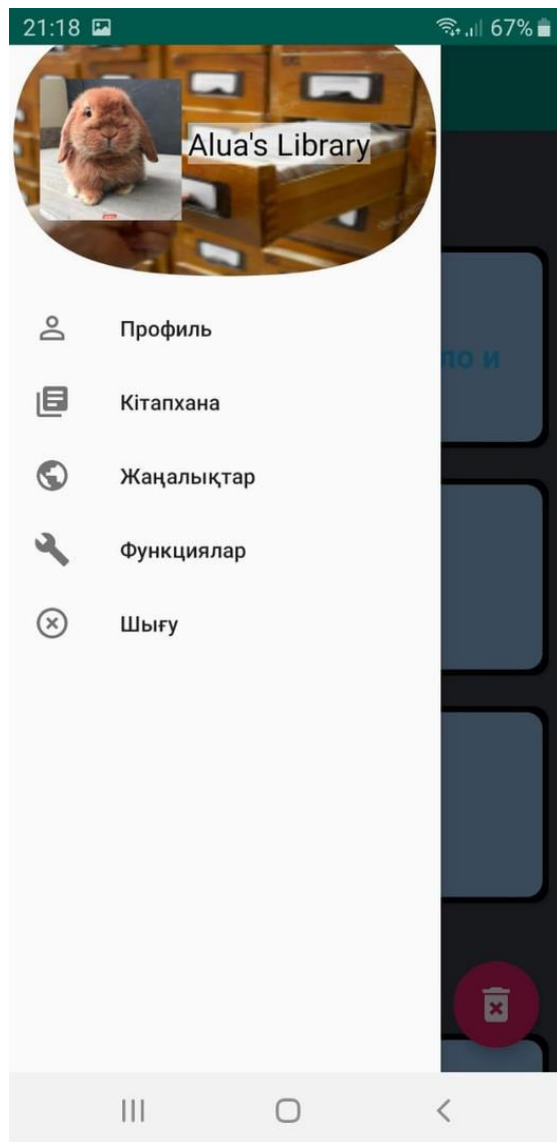


Рисунок 3.14 – Выход из приложения

## **4 Экономическая часть**

### **4.1 Резюме**

На сегодня все больше людей используют для обмена информацией не персональные компьютеры и ноутбуки, а современные мобильные телефоны, которые по функционалу практически сравнялись с упомянутыми устройствами. Как следствие, разработка мобильных приложений становится все более и более востребованной – над ними сегодня работает не меньше людей, чем над разработкой программного обеспечения для полнофункциональных компьютерных систем.

Развитие технологий и потребность пользователей в быстром и удобном доступе к информации привели к тому, что мобильные телефоны, а именно, специально разработанные приложения для них, используются в качестве основного источника доступа к информационным и корпоративным ресурсам, таким как социальные сети, коммуникационные службы, информационные порталы и прочее.

В этом проекте разработано мобильное приложение «Библиотека». Создание мобильного приложения на данный момент является довольно так и частым явлением, так как сейчас это очень удобно просматривать интересующую для тебя область в доступном мобильном приложении. Целью данного проекта является изучение и разработка удобного мобильного приложения, которое будет доступно для каждого, где можно будет получить доступ к книгам, методическим указаниям, которые есть в наличии. Основной функцией данного приложения является удобство при чтении скачанных книг, запоминание страниц, на которой остановился пользователь, а также удобный и приятный интерфейс.

Процесс создания любого мобильного приложения необходимо разделить на отдельные этапы, которые в свою очередь делятся на подразделы.

Все этапы разработки были выполнены одним человеком с использованием сторонней помощи.

В условиях сегодняшнего рынка, крайне необходимо, чтобы расходы, вложенные в исследование программного продукта, окупили себя. В этой главе производится организационно-экономический вычисление коммерческой цены разработки программного обеспечения. Расчеты учитывают расходы на исследование и введение ПО, расходы до внедрения и после, экономию расходов.

Разрабатывая новый программный продукт, необходимо правильно рассчитать проектную стоимость, которая напрямую зависит от сроков его исполнения. Существуют трудности в оценке затрат на разработку программного обеспечения, так как необходимо учитывать множество разнообразных факторов: сложность и объем решаемой задачи, стоимость и доступность необходимых ресурсов, необходимый уровень абстракции при разработке и др.

Разработка проектов программных средств связана со значительными затратами ресурсов (трудовых, материальных, финансовых). В связи с этим создание и реализация каждого проекта программного обеспечения нуждается в соответствующем технико-экономическом обосновании

Для того, чтобы определить трудоемкость разработки программного проекта, необходимо составить список всех основных этапов и видов работ, которые должны быть выполнены. Определение трудоемкости (затрат времени) на разработку ПП вызывает такие же трудности, что и нормирование любого творческого труда, содержащего технические (рутинные) элементы. Творческие элементы труда программистов практически не нормируются, они могут быть определены либо на основе экспертных оценок опытных программистов, либо жестко заданными сроками разработки, в которые программист обязан найти решение.

Определение затрат на разработку ПП производится путем составления соответствующей сметы, которая включает следующие статьи: материальные затраты, затраты на оплату труда, социальный налог, амортизация основных фондов, прочие затраты. Прямые материальные затраты включают в себя фактическую себестоимость основных материалов, израсходованных на производство.

Первичными документами для составления «сметы затрат» являются: план проведения работ по модернизации и реконструкции – график основных этапов и содержание работ, смета на приобретение оборудования для проведения работы, смета на приобретение основных и вспомогательных материалов, смета на покупку комплектующих изделий, расчет затрат на потребляемую электроэнергию, а также определение суммы амортизационных отчислений, используемых оборудования и приборов.

## **4.2 Трудоемкость разработки ПП**

Основными задачами планирования работ являются:

- определение объема предстоящих работ;
- взаимная увязка работы и установление рациональной последовательности предстоящих работ;
- установление сроков выполнения работ.

Работы по планированию сводятся к составлению перечня работ, определению их трудоемкости, расчету длительности цикла работ, обоснования сметы затрат на проведение работ.

Полный перечень работ с разделением их по этапам выполнения проекта следует оформить в виде таблицы, фрагмент которой показан в таблице 4.1.

Таблица 4.1 – Распределение работ по этапам и видам и оценка их трудоемкости

Этапы разработки ПП	Вид работы на данном этапе	Трудоемкость разработки ПП	
		Чел. х час	Час х день
Анализ требований	Формирование цели и задач проекта, выделение базовых сущностей и взаимосвязей между ними	1 х 16	8 х 2
Анализ рынка	Анализ подобных проектов и выявление заинтересованности в данном продукте	1 х 16	8 х 2
Проектирование	Получение технических заданий, назначение требований к пользовательскому интерфейсу, оценка и подбор оборудования	1 х 24	8 х 3
Реализация	Экспериментирование и анализ, строение прототипов, как целой системы, так и ее частей	1 х 48	8 х 6
Тестирование продукта	Тестирование и дополнение в системе	1 х 40	8 х 5
Подготовка инструкции	Подготовка полной инструкции для работы с продуктом	1 х 8	8 х 1
Внедрение и поддержка	Установка программного обеспечения, обучение пользователей, исправление выявленных ошибок.	1 х 40	8 х 5
Итого трудоемкость выполнение проекта		1 х 192	8 х 24

### 4.3 Расчет затрат на разработку ПП

Для определения затрат на разработку ПП нужно составить смету, которая включает следующие статьи:

- материальные затраты;
- затраты на оплату труда;
- социальный налог;
- амортизация основных фондов;
- прочие затраты.

### 4.4 Материальные затраты

Затраты на основные и вспомогательные материалы относятся к материальным затратам. Расчет затрат на материальные ресурсы и стоимость оборудования производится по форме, приведенной в Таблице 4.2

Таблица 4.2 – Стоимость оборудования и ПО

№	Наименование	Описание	Цена за единицу, тг	Сумма, тг
1	Ноутбук	LenovoB50-10	60000	60000
2	Операционная система	MicrosoftWindows 7 Максимальная	45000	45000
3	Программа для кода	Android Studio	бесплатно	бесплатно
4	Антивирус	ESET NOD32 Cyber Security Pro	9990	9990
5	Принтер	HP LaserJet1020	25000	25000

Таблица 4.3 – Затраты на материальные ресурсы

№	Наименование	Описание	Цена за единицу, тг	Сумма, тг
1	Бумага	A4	1 200	2 400
2	Картридж		3 000	3 000

### 4.5 Затраты на электроэнергию

Эта глава включает затраты на технологические нужды, которые приведены в Таблице 4.4. Общая сумма затрат рассчитывается по формуле (4.1).

$$Z_3 = \sum_{i=1}^n M_i * K_i * T_i * C, \quad (4.1)$$

С 1 января 2019 года цена на электроэнергию по тарифу ТОО «АлматыЭнергоСбыт» составляет 15,90 тенге за 1 кВтч без НДС

Таблица 4.4 – Затраты на технологические нужды

Наименование оборудования	Паспортная мощность, кВт	Коэффициент использования мощности	Время работы оборудования для разработки ПП, ч	Цена электроэнергии г/кВт*ч	Сумма, тг
Ноутбук	0,3	0,7	192	15,90	641,088
Принтер	0,25	0,7	24	15,90	66,78
Итого затраты на электроэнергию					707,868

#### 4.6 Затраты на оплату труда

Эта статья затрат учитывает выплаты по заработной плате за выполненную работу, исчисленные на основании тарифных ставок и должностных окладов в соответствии с принятой в организации – разработчике системой оплаты труда. Затраты на оплату труда рассчитывают по форме, приведенной в таблице 4.5.

Общая сумма затрат на оплату труда рассчитывается по формуле (4.2).

$$Z_{тр} = \sum_{i=1}^n ЧС_i * T_i, \quad (4.2)$$

Часовая ставка работника, рассчитанная по формуле, равняется – 500 тг/час.

Ежемесячная заработная плата начинающего инженер-программиста, который участвовал в разработке ПП = 115 200 тг.

Таблица 4.5 – Затраты на оплату труда

Категория работника	Трудоемкость разработки ПП, чел. х ч	Часовая ставка, тг/ч	Сумма, тг
Разработчик	1 х 192	600	115 200
ИТОГО затрат на оплату труда			115 200



#### 4.7 Социальный налог

Отчисления на социальные нужды учитывает 9,5 % от затрат на оплату труда всех работников, однако пенсионные отчисления (10% отЗтр) не облагаются социальным налогом.

Обязательные пенсионные отчисления составят:

$$\text{ОПВ} = 115\,200 * 10\% = 11\,520 \text{ (тенге)}.$$

Отсюда, сумма социального налога составит:

$$\text{СН} = (115\,200 - 11\,520) * 9,5\% = 9\,849,6 \text{ (тенге)}.$$

#### 4.8 Амортизация основных фондов

По статье «Амортизация основных фондов» рассчитываются амортизационные отчисления, исходя из стоимости основных средств, используемых в процессе разработки программного продукта, сроков эксплуатации оборудования и годовой нормы амортизации.

Амортизация отчисления определяются согласно Таблице 4.6. Сумма амортизационных отчислений вычисляется по формуле (4.3).

$$Z_{\text{ам}} = \frac{C_{\text{обор}} * H_a * N}{100 * 12 * t}, \quad (4.3)$$

где  $H_a$  – норма амортизации %;

$C_{\text{обор}}$  – первоначальная стоимость оборудования;

$N$  – время использования оборудования;

$t$  – количество рабочих дней в месяце.

Необходимо учитывать, что в стоимость ОФ также входят затраты на такие вещи как: доставка, монтаж, установка программного обеспечения и оборудования. Норма амортизации для линейного способа начисления вычисляется по формуле (4.4).

$$H_{ai} = \frac{100}{T_{hi}}, \quad (4.4)$$

Использование ОФ варьируется от 3 до 10 лет. Все используется в течении 7 лет. Программное обеспечение – 3 года. Используя формулу (4.4), заполним Таблицу 4.6 для отображения амортизации основных фондов.

$$H_{A1} = 100/7 = 14,29\%.$$

$$H_{A3} = 100/3 = 33,33\%.$$

Расчеты амортизации:

$$Z_{ам} = (60000 \times 0,1429 \times 24) / (1 \times 12 \times 24) = 714,5 \text{ тг};$$

$$Z_{ам} = (45000 \times 0,3333 \times 24) / (1 \times 12 \times 24) = 1249,87 \text{ тг};$$

$$Z_{ам} = (9990 \times 0,3333 \times 24) / (1 \times 12 \times 24) = 277,47 \text{ тг};$$

$$Z_{ам} = (25000 \times 0,1429 \times 24) / (1 \times 12 \times 24) = 297,71 \text{ тг}.$$

Таблица 4.6 – Амортизация основных фондов

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Время работы оборудования и ПО для разработки ПП, д	Сумма, тг
Ноутбук	60 000	14,29	24	714,5
MicrosoftWindows Максимальная 7 64 bit	45000	33,33	24	1249,87
Антивирус	9990	33,33	24	277,47
Принтер	25 000	14,29	24	297,71
ИТОГО амортизация основных фондов				2539,55

#### 4.9 Прочие затраты

Статья «Прочие затраты» представляет собой расходы за аренду помещения, коммунальные услуги, затраты на электроэнергию, рекламу и другие хозяйственные и организационные расходы.

Арендная плата за 1 кв.м. площади определяется:

$$A_{п} = C \times S \times K1 \times K2 \times K3 \times K4 \times K5 \times K6;$$

где  $A_{п}$  – ставка арендной платы за пользование помещением;

$C$  – базовая ставка арендной платы за имущественный наем 1 кв.м. - 1,5 МРП;

$S$  – арендная площадь, квадратный метр;

Таблица 4.7 – Виды коэффициентов

№ п/п	Вид коэффициентов	Размер коэффициента
1	Коэффициент, учитывающий тип строения К1:	
	административное	1
	производственное	0,8
	складское, котельная	0,6
2	Коэффициент, учитывающий наличие инженерных сетей и коммуникаций К2:	
	Для помещений со всеми инженерно-техническими устройствами При отсутствии каких-либо видов коммуникаций уменьшается на 0,1 за каждый вид	1,0
3	Коэффициент, учитывающий степень комфортности помещения К3:	
	Здание или помещение в здании	1,0
	Цокольный этаж здания или полуподвал	0,7
	Подвальное помещение	0,4
4	Коэффициент, учитывающий вид деятельности арендатора К4:	
	Оказание услуг:	
	- бытовые и культурно-досуговые услуги, тренажерные залы, фитнес- клубы, СТО, мастерские	2,0
	- ритуальные услуги, прокат книг	1,0
	Оказание услуг в сфере общественного питания:	
	- кафе, рестораны	3,0
	- буфеты и столовые	2,0
	- буфет и столовая в организациях образования	0,5
Оказание услуг в области образования (внешкольная, кружковая деятельность для учащихся, компьютерные классы), физкультуры и спорта для детей и юношества, здравоохранения и культуры	1,0	
Прочие виды деятельности	1,0	

Продолжение таблицы 4.7

5	Коэффициент, учитывающий организационно-правовую форму нанимателя К5:	
	<ul style="list-style-type: none"> <li>- для структур с частной формой собственности (акционерные общества, товарищества с ограниченной ответственностью)</li> <li>- для общественных организаций и объединений, неправительственных организаций,</li> <li>- для индивидуальных предпринимателей для организации производственной деятельности и развития сферы услуг населению, за исключением торгово-закупочной (посреднической) деятельности,</li> <li>- для государственных учреждений:</li> <li>- финансируемых из бюджета и арендующих имущество, находящееся в коммунальной собственности акимата иного уровня бюджета</li> <li>- финансируемых из бюджета и арендующих имущество, находящееся коммунальной собственности акимата соответствующего уровня бюджета и для районных коммунальных предприятий</li> <li>- для иных, не учтенных выше</li> </ul>	<ul style="list-style-type: none"> <li>2,0</li> <li>1,5</li> <li>0,9</li> <li>без взимания арендной платы</li> <li>1,0</li> </ul>
6	Коэффициент, учитывающий территориальное расположение К6:	
	Районный центр	1,0

К1 = 1,0;

К2 = 1,0;

К3 = 1,0;

К4 = 1,0;

К5 = 1,0;

К6 = 1,0.

Тогда:

$A_{\text{п}} = 1,5 \times 2525 \times 18 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 = 68\ 175$  тг

Расходы на электроэнергию представлены в таблице 4.8

Таблица 4.8 – Затраты на электроэнергию

Наименование оборудования	Кондиционер	Освещение
Паспортная мощность	0,9	0,9
Коэффициент использования мощности	0,9	0,7
Время работы оборудования для разработки ПП, ч	192	192
Цена электроэнергии тг/кВт*ч	15,90	15,90
Сумма, тг	2472,768	1923,264
Итого затраты на электроэнергию		4 396,032

При разработке моего программного продукта использовались расходы для размещения рекламы в Интернет, что составило 6300 тенге в месяц и соответственно расходы за Интернет в сумме 5300 тенге за месяц. Арендная плата за месяц составляет 68 175 тенге. Расходы за электроэнергию 4 396,032 тенге в месяц. Итого по прочим затратам сумма составляет 88 567,064 тенге.

#### 4.10 Смета затрат на разработку ПП

Рассчитав все возможные затраты, связанные с созданием программного обеспечения, опираясь на расчеты, полученные в пунктах 4 – 8 составила смету общих затрат и отразила ее в Таблице 4.9.

Таблица 4.9 – Смета затрат на разработку ПП

Статья затрат	Сумма, тг
Оплата труда	115 200
Социальный налог	9 849,6
Электроэнергия	707,868
Амортизация основных фондов	2 539,55
Прочие затраты	88 567,064
ИТОГО по смете	216 864,082

#### 4.11 Определение возможной (договорной) цены ПП

Величина возможной (договорной) цены ПП устанавливается на основе эффективности, качества и сроков её выполнения на уровне, отвечающим экономическим интересам заказчика (потребителя) и исполнителя и вычисляется по формуле (4.5).

$$Ц_{д} = З_{\text{нир}} \left( 1 + \frac{P}{100} \right), \quad (4.5)$$

P – средний уровень рентабельности ПП принимается в размере 20%.

$$Ц_{д} = 216\,864,082 * (1 + 0,2) = 260\,236,898 \text{ (тенге)}.$$

Далее определяется цена реализации с учетом налога на добавленную стоимость (НДС), ставка (НДС) устанавливается законодательно. Налоговым Кодексом РК на 2019 год ставка НДС установлена в размере 12%.

Цена реализации с учетом НДС рассчитывается по формуле (4.6):

$$Ц_{р} = Ц_{д} + Ц_{д} * \text{НДС}, \quad (4.6)$$

$$Ц_{р} = 260\,236,898 + 260\,236,898 * 0,12 = 291\,465,326 \text{ (тенге)}.$$

Рассчитанная возможная цена ПП составляет 291 465,326 тенге.

#### 4.12 Оценка эффективности внедрения программных средств

Экономическая эффективность до внедрения веб-приложения рассчитывалась и осуществлялась одним работником.

Затраты на решение задачи без использования программного средства рассчитываются по формуле (4.7):

$$З_{\text{тр}} = ФЗП_{р} + ОТ_{з/п}, \quad (4.7)$$

где  $ФЗП_{р}$  – фонд заработной платы группы лиц, решающих данную задачу;

$ОТ_{з/п}$  – отчисления на социальные нужды (9,5%) [9].

Фонд заработной платы работников определяется по формуле (4.8):

$$ФЗП_{р} = ЗП_{р} * N * 12, \quad (4.8)$$

где  $ЗП_{р}$  – оклад работника, тенге/месяц;

$N$  – количество работников [9].

Оклад работника составляет 115200 тенге в месяц  
Исходя из этого, фонд заработной платы сотрудников за год составляет:

$$\Phi ЗП_p = 115\,200 * 12 = 1\,382\,400 \text{ тг.}$$

$$ОТ_{з/п} = (\Phi ЗП_p - \Phi ЗП_p * 10\%) * 9,5\% = (1\,382\,400 - 1\,382\,400 * 0,1) * 0,095 = 118\,195,2 \text{ тг.}$$

Подставив полученный результат в формулу (4.8) и рассчитаем затраты на решение задачи без использования программного продукта:

$$З_{тр} = 1\,382\,400 + 118\,195,2 = 1\,500\,595,2 \text{ тг.}$$

Годовые затраты машинного времени на решение задачи определяются по формуле (4.9):

$$З_m = K * q * 12 \text{ ,} \quad (4.9)$$

где  $K$  – количество часов использования ПК в месяц;  
 $q$  – стоимость часа аренды сервера (146 тенге/час) [9].

С учетом 8 часового рабочего дня, а также 24 рабочих дней в месяц, получаем часы использования ПК в месяц  $K=192$  час. Исходя из этого получим:

$$З_m = 192 * 146 * 12 = 336\,384 \text{ тг.}$$

Скорость печати одного документа 0,1 минута (т.е. 0,0016 часа).  
Годовые затраты для печати результата с принтера  $K_{печ}$  определяются:

$$З_п = t_п * N_э * q \text{ ,} \quad (4.10)$$

где  $t_п$  – время на печать одного экземпляра;

$N_э$  – количество экземпляров в год;

$q$  – стоимость часа машинного времени (146 тенге/час) [9].

Учитывая, что в день примерно печатается 3 документа и 24 рабочих дней в месяц, то получим 864 экземпляров в год.

Исходя из этого, годовые затраты на печать составляют:

$$З_п = 0,0016 * 864 * 146 = 202 \text{ тг}$$

Суммарные затраты после внедрения программного продукта определяются по формуле:

$$Z_{\text{ом}} = Z_{\text{м}} + Z_{\text{п}} , \quad (4.11)$$

Подставив значения, получим:

$$Z_{\text{ом}} = 336\,384 + 202 = 336\,586 \text{ тг.}$$

Экономия затрат от внедрения программного продукта определяется по формуле [9]:

$$\mathcal{E} = Z_{\text{тр}} - Z_{\text{ом}} , \quad (4.12)$$

где  $Z_{\text{тр}}$  – затраты до внедрения системы;

$Z_{\text{ом}}$  – затраты после внедрения системы [9].

Подставив значения получим следующее:

$$\mathcal{E} = 1\,500\,595,2 - 336\,586 = 1\,164\,009,2 \text{ тг.}$$

Срок окупаемости программного продукта определяется по формуле [9]:

$$T_{\text{ок}} = C/\mathcal{E} , \quad (4.13)$$

где  $C$  – затраты на разработку и внедрение системы, тенге;

$\mathcal{E}$  – экономия затрат от внедрения системы, тенге/год [9].

Подставив значения, получим:

$$T_{\text{ок}} = 291\,465,326 / 1\,164\,009,2 = 4 \text{ (месяца).}$$

$E_{\text{н}}$  – нормативный коэффициент эффективности капитальных вложений ( $E_{\text{н}} = \text{прибыль/затраты}$ ):

$$E_{\text{н}} = 1\,164\,009,2 / 291\,465,326 = 3,99$$

Вывод по разделу: сделав полный анализ вышеприведенных вычислений, можно прийти к тому, что внедрение мобильного приложения, который был разработан в данном проекте принесет экономический эффект. По результатам расчета экономической эффективности проектирования и внедрения средства автоматизации также можно сказать, что это прибыльно. Хотя выгода и косвенная, но, как правило, заметная в средне и долгосрочной перспективе.

Также, проанализировав рынок, можно сделать вывод, что внедрение данного программного продукта является выгодным, так как является востребованным и очень интересным пользователям.

В результате расчета, затраты на создание данного мобильного приложения составили 291 465,326 тенге. Эти расходы окупаются за 4 месяца. Годовой программный продукт позволяет нам экономить до 1 164 009,2 тенге



в год, что говорит о целесообразности и успешности внедрения данного программного продукта.

## 5 Безопасность жизнедеятельности

### 5.1 Общее положение

В данном дипломном проекте рассматриваются элементы разработки мобильного приложения для библиотеки в среде «Android». Делается это с целью облегчения процесса поиска необходимых книг для чтения, а также изучения материала. Предложенное мобильное приложение значительно облегчает всю процедуру от начала до конца. Это обуславливается тем, что весь процесс происходит в онлайн режиме. Мобильное приложение будет применяться в отделе информационных технологий. Отдел находится на 5 этаже в кабинете №513.

Проанализировав условия труда в данном кабинете, можно сказать, что в данной комнате сидят более 10 человек, имеется серверный отдел, а также хорошее освещение, не только искусственного, но и естественного. Но, что касается, вентиляции, в кабинете установлен только один кондиционер, который имеет недостаточной мощности для необходимого комфорта в помещении. В расчетной части выясним оптимальную мощность, используя внешние и внутренние тепловые нагрузки. Для расчётов понадобятся фактические данные, которые приведены в таблице 5.1.

Таблица 5.1 – Фактические данные

Город	Алматы
Параметры помещения (Д x Ш x В), м	10 x 6 x 3
Кол-во оборудования, шт	11
Мощность $P_{об}$ , кВт/ч	0,5
КПД, $\eta$	0,9
Данные по ист. света: мощ. N ос.уст., Вт/м <sup>2</sup>	36
Вид источника света	люминесцентные лампы
Число сотрудников: мужчины	6
женщины	3
Кол-во окон	1
Площадь 1 окна, м <sup>2</sup>	10
Расположение	Север
Вид	жалюзи, пластиковый переплет загрязнение незначительное
Расчетное время суток, ч	13-14

Температура в помещении, °С: летом	25
зимой	20
Вид положения работы	сидя

## 5.2 Расчет вентиляции

### Задание:

- считать количество воздуха, необходимое для подачи в помещение;
- проверьте уже установленные кондиционеры в этой комнате;
- по найденному значению количества воздуха выбрать подходящую модель кондиционера;
- привести основные характеристики выбранного кондиционера;
- обеспечить компоновку кондиционера в помещении и подачу воздуха;
- сделать выводы и предложения.

## 5.3 Расчет тепловых нагрузок в помещении

В помещениях различного назначения в основном действуют тепловые нагрузки, возникающие вне помещения (наружные); а также тепловые нагрузки, возникающие внутри зданий (внутренние).

### 5.3.1 Наружные тепловые нагрузки

$$Q_{огр} = V_{пом} \cdot X_o \cdot (t_{Нрасч} - t_{Врасч}), Вт \quad (5.1)$$

где  $V_{пом}$  – объем помещения, м<sup>3</sup>:

$$V_{пом} = 10 \cdot 6 \cdot 3 = 180 \text{ м}^3;$$

$X_o$  – удельная тепловая характеристика, Вт/м<sup>3</sup>°С:

$$X_o = 0.42 \text{ Вт/м}^3 \text{°С};$$

$t_{Нрасч}$  – наружная температура (параметр А). Для холодного периода – средняя температура самого холодного месяца в 13 часов, для теплого периода – средней температуре самого жаркого месяца в 13 часов.

$t_{Врасч}$  – внутренняя температура, выбирается с учетом комфортных условий или технологических требований, предъявляемых к производственным процессам.

Для теплого времени года

$$t_{Нрасч} = 29,4 \text{ °С}$$

$$t_{Врасч} = 25 \text{ °С}$$

$$Q_{огр} = 180 \cdot 0,42 \cdot 4,4 = 332,64 \text{ Вт}$$

Для холодного времени года

$$t_{Нрасч} = -9 \text{ °С}$$

$$t_{Врасч} = 20 \text{ °С}$$

$$Q_{огр} = 180 \cdot 0,42 \cdot (-29) = -2192,4 \text{ Вт}$$

Избыточная теплота солнечного излучения в зависимости от типа стекла почти до 90% поглощается средой помещения, остальная часть отражается. Максимальная тепловая нагрузка достигается при максимальном уровне излучения, которое имеет прямую и рассеянную составляющие.

Интенсивность излучения зависит от ширины местности, времени года и времени суток.

Теплопоступление от солнечного излучения через остекление определяется по формуле (5.2):

$$Q_p = (q^I F_o^I + q^II F_o^{II}) \cdot \beta_{с.з.} \quad (5.2)$$

где  $q^I$ ,  $q^{II}$  – тепловые потоки от прямой и рассеянной солнечной радиации, Вт/м<sup>2</sup>;

$F_o^I$ ,  $F_o^{II}$  – площади светового проема, облучаемые и необлучаемые прямой солнечной радиацией, м<sup>2</sup>;

$\beta_{с.з.}$  – коэффициент теплопропускания.

$$\beta_{с.з.} = 0.15$$

При отсутствии наружных затеняющих козырьков, ребер и т. д. для периода облучения остекления солнцем, когда его лучи проникают через окно в помещение  $F_o^I = F_o$ ;  $F_o^{II} = 0$ , (3) [1]:

$$Q_p = q^I F_o \cdot \beta_{с.з.} = (q_{пр} + q_{рп}) \cdot K_1^f \cdot K_2 \cdot \beta_{с.з.} \cdot n \cdot S_o, \text{ Вт} \quad (5.3)$$

$q_{пр}$ ;  $q_{рп}$  – тепловые потоки от прямой рассеянной радиации, Вт/м<sup>2</sup>. По таблице 5.1 для Ю, после полудня в 12-13 ч. при расположении 3:

Таблица 5.1 - Поверхностная плотность теплового потока (прямой и рассеянной) солнечной радиации в июле, Вт/м<sup>2</sup>

Географическая широта, градус	Часы до полудня	Ориентация вертикального светового проема								Часы после полудня	
		С	СВ	В	ЮВ	Ю	ЮЗ	З	СЗ		
44	5-6	С	СВ	В	ЮВ	Ю	ЮЗ	З	СЗ	18-19	
		84	222	292	72	—	—	—	—		
	8-9	42	53	58	40	23	22	22	23	15-16	
		—	256	490	398	66	—	—	—		
	9-10	71	101	121	108	79	60	59	60	14-15	
		—	84	371	387	162	—	—	—		
	10-11	64	80	100	101	81	63	60	62	13-14	
		—	2	193	305	245	—	—	—		
			60	71	81	86	84	67	60	64	

	11-12	—	—	<u>37</u>	<u>214</u>	<u>288</u>	<u>73</u>	—	—	12-13
		59	67	72	79	85	77	65	65	

$$q_{вп} = 73 \text{ Вт/м}^2; q_{вр} = 77 \text{ Вт/м}^2;$$

$F_0 = nS_0 = 1 \cdot 10 = 10 \text{ м}^2$  – площадь светового проема ( $n$  – число окон;  $S_0$  – площадь 1 окна);

$K_1$  – коэффициент затемнения остекления переплетами ( $K_1^C$  – для облученных проемов). По таблице 5.2:

Таблица 5.2 – Коэффициент ( $K_1$ ) затемнения остекления переплетами

Заполнение светового проема	$K_1$
Остекление в пластиковом переплете:	
одинарное	0,8
двойное	0,72
Остекление без переплетов:	
одинарное	1,0
двойное	0,9
Остекление в деревянных переплетах	
одинарное	0,65
двойное	0,6

$$K_1^C = 0.72;$$

$K_2$  – коэффициент загрязнения остекления. По таблице 5.3:

Таблица 5.3 - Коэффициент загрязнения остекления ( $K_2$ )

Степень загрязнения остекления	$K_2$
Значительное	0,85
Умеренное	0,9
Незначительное	0,95
Чистое стекло	1

$$K_2 = 0.95.$$

Тогда:

$$Q_p = (73 + 77) \cdot 0.72 \cdot 0.95 \cdot 0.72 \cdot 0.15 \cdot 10 = 110,8 \text{ Вт}$$

### 5.3.2 Внутренние тепловые нагрузки

Внутренние нагрузки в жилых, офисных или относящихся к сфере обслуживания помещениях слагаются в основном из тепла:

- выделяемого людьми;
- выделяемого лампами и осветительными, электробытовыми приборами;
- выделяемого компьютерами, печатающими устройствами фотокопировальными машинами пр.;

В производственных и технологических помещениях различного назначения дополнительными источниками тепловыделений могут быть: нагретое производственное оборудование, горячие материалы, в том числе жидкости и различного рода полуфабрикаты, продукты сгорания и химических реакций.

Теплопоступления от людей зависят от интенсивности выполняемой работы и параметров окружающего воздуха. Тепло, выделяемое человеком, складывается из ощутимого (явного), то есть передаваемого в воздух помещения путем конвекции и лучеиспусканий, и скрытого тепла, затрачиваемого на испарение влаги с поверхности кожи и из легких.

Летом при 24 °С один мужчина выделяет явного тепла 67 Вт, а общего – 102 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого мужчины. Тогда выделение явного тепла в помещении составит:

$$Q_{л}^{\text{я}} = 67 \cdot 6 + 67 \cdot 3 = 603 \text{ Вт}$$

А выделение общего тепла:

$$Q_{л}^{\text{о}} = 102 \cdot 6 + 102 \cdot 3 \cdot 0,85 = 872,1 \text{ Вт}$$

Зимой при 20°С один мужчина выделяет явного тепла 82 Вт, а общего – 103 Вт. Тогда выделение явного тепла в помещении составит:

$$Q_{з}^{\text{я}} = 82 \cdot 6 + 82 \cdot 3 = 738 \text{ Вт}$$

А выделение общего тепла:

$$Q_{з}^{\text{о}} = 103 \cdot 6 + 103 \cdot 3 \cdot 0,85 = 880,65 \text{ Вт}$$

Теплопоступление от осветительных приборов, оргтехники и оборудования рассчитывается следующим образом. Теплопоступление от ламп определяется по формуле (5.4) [1]:

$$Q_{осе} = \eta \cdot N_{осе} \cdot F_{пол}, \text{ Вт} \quad (5.4)$$

где  $\eta$  – коэффициент перехода электрической энергии в тепловую (для люминесцентных ламп  $\eta=0.5-0.6$ );

$N_{\text{осв}}$  – установленная мощность ламп ( $N= 36 \text{ Вт/м}^2$ );

$F_{\text{пол}}$  – площадь пола:  $F_{\text{пол}} = 10 * 6 = 60 \text{ м}^2$

Тогда:

$$Q_{\text{осв}} = 0,5 * 36 * 60 = 1080 \text{ Вт}.$$

Тепло, выделяемое производственным оборудованием, определяется по формуле (5.5) [1]:

$$Q_{\text{об}} = N_{\text{уж}} \cdot K \quad (5.5)$$

$$Q_{\text{об}} = 1,8 * 11 * 0,95 = 18,81 \text{ кВт}$$

Теплопритоки, возникающие за счет находящейся оргтехники - это 30% мощности оборудования:

$$Q_{\text{об}} = 1,8 * 11 * 0,3 = 5,94 \text{ кВт}$$

#### 5.4 Расчет теплового баланса помещения

На основании выполненных расчетов составим баланс теплоступлений в помещении, который рассчитывается по формуле (5.6):

$$Q_{\text{взд}} = Q_{\text{д}} + Q_{\text{в}} + Q_{\text{мв}} + Q_{\text{из}} + Q_{\text{изб}} + Q_{\text{изб}} \quad (5.6)$$

Лето:

$$Q_{\text{изб}} = 110,8 + 603 + 1080 + 18810 + 5940 + 332,64 = 26\ 876,44 \text{ Дж}$$

Зима:

$$Q_{\text{изб}} = 110,8 + 738 + 1080 + 18810 + 5940 - 2192,4 = 24\ 486,4 \text{ Дж}$$

Так как тепловой баланс для лета больше зимнего теплового баланса, то рассчитаем теплонапряженность воздуха по формуле (5.7):

$$Q_{\text{н}} = \frac{Q_{\text{изблето}} * 860}{V_{\text{пом}}} \quad (5.7)$$

Тогда:

$$Q_{\text{н}} = \frac{26\ 876,44 * 860}{180} = 128,4 \text{ ккал/м}^3$$

При  $Q_{\text{н}} > 20 \text{ ккал/м}^3$ ,  $\Delta t = 8 \text{ }^\circ\text{C}$ ,

Определение количества воздуха, необходимое для поступления в помещение рассчитывается по формуле (5.8):

$$L = \frac{Q_{\text{изб}} * 860}{C * \Delta t * \gamma} \quad (5.8)$$

Тогда:

$$L = \frac{128,4 * 860}{0,24 * 8 * 1206} = 476,9 \text{ м}^3/\text{час}$$

где  $C=0,24 \text{ ккал/(кг}^\circ\text{C)}$  – теплоемкость воздуха,

$\gamma=1,206 \text{ кг/м}^3$  – удельная масса приточного воздуха.

## 5.5 Выбор кондиционера. Схема расположения

Обеспечение воздушного комфорта в жилых и производственных помещениях зависит от систем аспирации, вентиляции, отопления и кондиционирования воздуха.

Выполнив данные расчеты, была выбрана модель кондиционера Fujitsu ASY7USBCW с подходящими характеристиками, которые указаны в таблице (5.4). Кондиционер с воздушным охлаждением, состоящий из двух блоков: внутреннего блока (собственно кондиционера), в котором расположены компрессор, испаритель, вентилятор и автоматика; внешнего блока – выносного конденсатора или теплообменника. Воздух подается сверху непосредственно в помещение, а забирается через лицевую панель.

Таблица 5.4 – Основные технические характеристики

Модель	ASY7USBCW/AOY7USNC
Тип	настенная сплит-система
Дополнительные режимы	автоматический режим
Основные режимы	охлаждение/обогрев
Максимальный воздушный поток, м <sup>3</sup> /час	380
Мощность в режиме охлаждения, Вт	2200
Мощность в режиме обогрева, Вт	2300
Потребляемая мощность при охлаждении, Вт	830
Потребляемая мощность при обогреве, Вт	750
Пульт дистанционного управления	есть
Таймер включения/выключения	есть
Фаза	однофазный
Другие функции и особенности	дезодорирующий фильтр, функция запоминания настроек
Электропитание, В	220
Класс энергоэффективности (+/-)	D
Охлаждение	от +21С до +43С



Рисунок 5.1 – Модель кондиционера ASY7USBCW



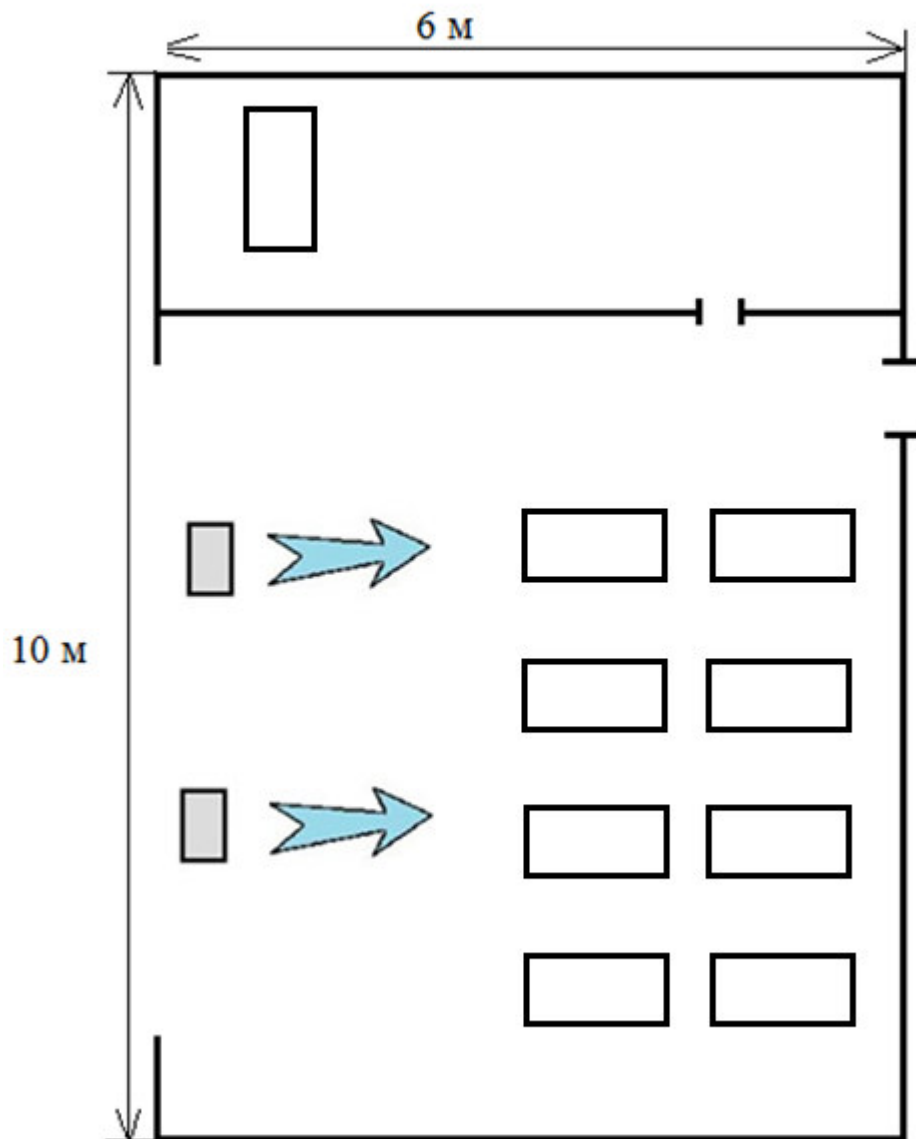


Рисунок 5.2 – Схема расположения кондиционера в производственном помещении

Вывод по разделу: Рассмотрев условия труда и имеющийся в данном кабинете модель кондиционера AC Electric07HN1\_16Y, который по расчетам не имеет достаточной мощности и имеется в малом количестве, я пришла к выводу, что в данный кабинет подходит кондиционер с верхней подачей воздуха модели FujitsuASY7USBCW (рисунок 5.1), который будет удовлетворять полученным требованиям по необходимому количеству воздуха. Из расчетов видно, что при достаточно маленьком пространстве и большом количестве человек и оборудования, количество избыточного тепла очень высоко, что предполагает установку достаточно мощной системы кондиционирования.

## Заключение

В данном дипломном проекте разрабатывалось мобильное приложение «Библиотека» в среде Android.. Основной задачей данного приложения является доступ студентам к необходимой литературе.

В первую очередь, мобильное приложение должно осуществить доступ простым пользователям к документам, имеющиеся в нем и в результате поиска находить нужную литературу. А также приложение имеет очень удобный и понятный интерфейс.

Во время проектирования были установлены основные требования к функциональности системы, предъявляемые пользователями. А затем на основе этих данных были построены: диаграмма последовательности, диаграмма размещения, взаимодействия, состояний, деятельности, компонентов и диаграмма прецедентов, которые отражают модель работоспособности системы и показывали возможности приложения.

В рамках дипломного проекта были достигнуты перечисленные ниже результаты. Было создано приложение, реализующее следующие функции:

- добавление меток по названию адреса
- регистрация пользователя по имени и паролю;
- информационная поддержка по выбранным объектам
- Изменение настроек и просмотр новостей.

Приложение было протестировано и показало свою работоспособность как на стандартных эмуляторах, взятых из SDK Android, так и на реальных устройствах на платформе Android (планшетном ПК и смартфоне).

## Список литературы

1. Медник З., Дорнин Л. Программирование под Android. Издательство Питер, 2012.
2. Амелин К. С., Граничин О. Н., Кияев В. И., Корявко А. В.. Введение в разработку приложений для мобильных платформ. Издательство ВВМ, 2011.
3. Дейтел П. Android для программистов: создаем приложения. Издательство Питер, 2012.
4. Голощапов А.Л. GoogleAndroid. Создание приложений для смартфонов и планшетных ПК. Издательство Питер 2012.
5. Мельникова О.М.: Смартфоны на Android. Издательство Эксмо, 2013.
6. Каймин В.А. Информатика: Учебное пособие: Изд. 2-е. Издательство РИОР, 2007.
7. Варакин М.В. Разработка мобильных приложений под Android. УЦ «Специалист» при МГТУ им. Н. Э. Баумана, 2012.
8. JohnWiley&Sons. RetoMeierProfessional Android 4 ApplicationDevelopment. Wrox, 2012.
9. MartinFowler. UML Distilled: A BriefGuidetotheStandardObjectModelingLanguage (ObjectTechnologySeries). AddisonWesley, 2003.
10. ПайлонД. UML 2 для программистов. ИздательствоПитер, 2012.
11. BillPhillips, BrianHardy. Android Programming: TheBigNerdRanchBig. NerdRanchGuides, 2013.
12. Котляров В.П. Основы тестирования программного обеспечения. Издательство Бином, 2009.
13. Абдимуратов Ж. С., Мананбаева С. Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат – Алматы: АУЭС, 2013. – 20 с.

## Приложение А

### Техническое задание

#### 1. Общие сведения

Техническое задание описывает особенности реализации мобильного приложения для платформы Android, взаимодействие с веб-сервисом системы, веб-клиентом и базой данных сервиса, а также возможности приложения по взаимодействию со сторонними сервисами.

Основная задача данного мобильного приложения поиск необходимых методических указаний, а также книг для студентов.

#### 2. Назначение и цели создания (развития) системы

Основной целью данной автоматизированной системы является разработка мобильного приложения «Библиотека», которое специально предназначено для поиска необходимой литературы для студентов, как методических указаний, так и книг разного жанра.

#### 3. Характеристика объектов автоматизации

Программная реализация описывалась в среде разработки Android Studio. Каждая из модулей имеет свой код, который отвечает за работу отдельных вкладок, ссылок и программ. Данные модули подробно описаны в основной части дипломного проекта.

#### 4. Требования к системе

В ходе всего проекта необходимо в результате получить программный продукт, который прошел тестирование, как и на платформе SDK Android Studio, так и на телефонах и планшетах, где операционная система Android.

- Приложение поддерживает возможность работы в портретной ориентации экрана;

- Локализация приложения предусматривает русскую, казахскую и английскую версии пользовательского интерфейса.

- Версия поддерживаемых Android устройств выше Android SDK 4.0., IOS выше 5.0

Мобильные приложения для Android должны реализовываться нативными средствами, либо на основе платформы, обеспечивающей компиляцию в нативные приложения для обеспечения максимальной производительности.

Документ представляет собой описание возможностей приложения в рамках функциональности отдельных экранов приложения.

Главным требованием к мобильному приложению – это соответствие дизайна мобильного приложения с основным сайтом университета.

## **5. Состав и содержание работ по созданию системы**

В составе всей автоматизированной системы имеется шесть модулей, каждый из которых отвечает за определенную работу всей программы. А именно:

- SplashActivity;
- MainActivity;
- LibraryFragment;
- NewsFragment;
- ProfileFragment;
- SettingsFragment;
- LoginActivity.

## **6. Источники разработки**

Источниками данной программной реализации является среда Android Studio, SDK Manager, сайт университета auret.kz.

## **7. Требования к документации**

В процессе работы над мобильным приложением программист-разработчик должен прокомментировать код программы (блоками), цветовая палитра приложения #2a579a.

Разработчик обеспечивает публикацию приложения для версии Android на GooglePlay на собственном аккаунте для дальнейшей технической поддержки.

## Приложение Б

### Листинг программы

#### Splash Activity

```
public class SplashActivity extends AppCompatActivity {

    WaveLoadingView mWaveLoadingView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        mWaveLoadingView = findViewById(R.id.waveLoadingView);
        mWaveLoadingView.setShapeType(WaveLoadingView.ShapeType.CIRCLE);
        mWaveLoadingView.setAmplitudeRatio(60);
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent = new Intent(SplashActivity.this, MainActivity.class);
                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);
                intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(intent);
            }
        }, 5000);
    }
}
```

#### MainActivity

```
public class MainActivity extends AppCompatActivity implements
LibraryFragment.OnFragmentInteractionListener {

    FragmentManager fragmentManager;
    FragmentTransaction fragmentTransaction;
    NavigationView navigationView;
    ImageView headerIV;
    TextView headerTV;
    Toolbar toolbar;
    DrawerLayout drawer;
    View headerLayout;
    Menu menuLayout;
    TextView toolbarTV;
    List<String> label;
    Boolean exit = false;
```

## Продолжение приложения Б

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    initialization();
    Paper.init(MainActivity.this);
    setSupportActionBar(toolbar);
    int i = getIntent().getIntExtra("intent", 0);
    ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
        this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
    drawer.addDrawerListener(toggle);
    toggle.syncState();
    label = new ArrayList<>();
    setLabel();

    Glide.with(this).load("https://pp.userapi.com/c854416/v854416043/347a0/UoBNTh
bOXaA.jpg").into(headerIV);
    headerTV.setText("Alua's Library");
    if (i == 1){
        openSettingsFragment();
        toolbarTV.setText(label.get(3));
    } else {
        openLibraryFragment();
        toolbarTV.setText(label.get(1));
    }
    navigationView.setNavigationItemSelectedListener(new
NavigationView.OnNavigationItemSelectedListener() {
        @Override
        public boolean onNavigationItemSelectedListener(@NonNull MenuItem menuItem)
        {
            int id = menuItem.getItemId();

            if (id == R.id.nav_profile) {
                openProfileFragment();
                toolbarTV.setText(label.get(0));
            } else if (id == R.id.nav_library) {
                openLibraryFragment();
                toolbarTV.setText(label.get(1));
            } else if (id == R.id.nav_news) {
                openNewsFragment();
                toolbarTV.setText(label.get(2));
            }
        }
    });
}
```

## Продолжение приложения Б

```
} else if (id == R.id.nav_settings) {
    openSettingsFragment();
    toolbarTV.setText(label.get(3));
} else if (id == R.id.nav_exit) {
    if (exit) {
        Paper.book().delete("UserLoggedIn");
        Intent intent = new Intent(MainActivity.this, SplashActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
    } else {
        exit = true;
        Toast.makeText(MainActivity.this,
            "Нажимая кнопку \"Выход\", вы перенаправитесь в окно
авторизации и удалите созданного вами пользователя",
            Toast.LENGTH_SHORT).show();
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                exit = false;
            }
        }, 2000);
    }
}
if (!exit) {
    drawer.closeDrawer(GravityCompat.START);
}
return true;
}
});
}

@Override
public void onBackPressed() {
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}

public void openLibraryFragment(){
```



## Продолжение приложения Б

```
fragmentManager = getSupportFragmentManager();
fragmentTransaction = fragmentManager.beginTransaction();
LibraryFragment libraryFragment = new LibraryFragment();
fragmentTransaction.replace(R.id.main_activity_frame,      libraryFragment,
LibraryFragment.TAG);
    fragmentTransaction.commit();
}
public void openProfileFragment(){
    fragmentManager = getSupportFragmentManager();
    fragmentTransaction = fragmentManager.beginTransaction();
    ProfileFragment profileFragment = new ProfileFragment();
    fragmentTransaction.replace(R.id.main_activity_frame,      profileFragment,
ProfileFragment.TAG);
    fragmentTransaction.commit();
}
public void openNewsFragment(){
    fragmentManager = getSupportFragmentManager();
    fragmentTransaction = fragmentManager.beginTransaction();
    NewsFragment newsFragment = new NewsFragment();
    fragmentTransaction.replace(R.id.main_activity_frame,      newsFragment,
NewsFragment.TAG);
    fragmentTransaction.commit();
}
public void openSettingsFragment(){
    fragmentManager = getSupportFragmentManager();
    fragmentTransaction = fragmentManager.beginTransaction();
    SettingsFragment settingsFragment = new SettingsFragment();
    fragmentTransaction.replace(R.id.main_activity_frame,      settingsFragment,
SettingsFragment.TAG);
    fragmentTransaction.commit();
}

@Override
public boolean onNavigationItemSelected(MenuItem item) {
    return false;
}

@Override
public void onFragmentInteraction(Uri uri) {}

public void initialization(){
    toolbar = findViewById(R.id.toolbar);
}
```

## Продолжение приложения Б

```
navigationView = findViewById(R.id.nav_view);
drawer = findViewById(R.id.drawer_layout);
headerLayout = navigationView.getHeaderView(0);
menuLayout = navigationView.getMenu();
headerIV = headerLayout.findViewById(R.id.header_IV);
headerTV = headerLayout.findViewById(R.id.header_text_view);
toolbarTV = findViewById(R.id.label_toolbar_main);
}
```

```
public void setLabel(){
    if (Paper.book().contains("language")) {
        if (Paper.book().read("language").equals("ru")) {
            label.add(0, getString(R.string.menu_profile_ru));
            label.add(1, getString(R.string.menu_library_ru));
            label.add(2, getString(R.string.menu_news_ru));
            label.add(3, getString(R.string.menu_settings_ru));
            label.add(4, getString(R.string.menu_exit_ru));
        } else if (Paper.book().read("language").equals("kz")) {
            label.add(0, getString(R.string.menu_profile_kz));
            label.add(1, getString(R.string.menu_library_kz));
            label.add(2, getString(R.string.menu_news_kz));
            label.add(3, getString(R.string.menu_settings_kz));
            label.add(4, getString(R.string.menu_exit_kz));
        } else {
            label.add(0, getString(R.string.menu_profile));
            label.add(1, getString(R.string.menu_library));
            label.add(2, getString(R.string.menu_news));
            label.add(3, getString(R.string.menu_settings));
            label.add(4, getString(R.string.menu_exit));
        }
    } else {
        label.add(0, getString(R.string.menu_profile));
        label.add(1, getString(R.string.menu_library));
        label.add(2, getString(R.string.menu_news));
        label.add(3, getString(R.string.menu_settings));
        label.add(4, getString(R.string.menu_exit));
    }
    menuLayout.getItem(0).setTitle(label.get(0));
    menuLayout.getItem(1).setTitle(label.get(1));
    menuLayout.getItem(2).setTitle(label.get(2));
    menuLayout.getItem(3).setTitle(label.get(3));
    menuLayout.getItem(4).setTitle(label.get(4));
}
```

```
}
}
```

### LibraryFragment

```
public class LibraryFragment extends Fragment {
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";
    public static final String TAG = "LibraryFragment";
    private String mParam1;
    private String mParam2;
    GridView gridView;
    GridViewAdapter gridViewAdapter;
    LibraryDbHelper mDbHelper;
    List<Book> books = new ArrayList<>();
    List<Book> booksToGridView = new ArrayList<>();
    Button loadDataButton;
    Spinner changeGenre;
    String genre;
    TextView libraryTV;
    private OnFragmentInteractionListener mListener;

    public LibraryFragment() {}

    public static LibraryFragment newInstance(String param1, String param2) {
        LibraryFragment fragment = new LibraryFragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
```

## Продолжение приложения Б

```
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    return inflater.inflate(R.layout.fragment_library, container, false);
}

public void onPressed(Uri uri) {
    if (mListener != null) {
        mListener.onFragmentInteraction(uri);
    }
}

@Override
public void onStart() {
    super.onStart();
    final FloatingActionButton fab = getView().findViewById(R.id.fab);
    gridView = getView().findViewById(R.id.library_list);
    loadDataButton = getView().findViewById(R.id.load_data_button);
    changeGenre = getView().findViewById(R.id.spinner_genre);
    libraryTV = getView().findViewById(R.id.libraryTV);
    mDbHelper = new LibraryDbHelper(getContext());
    Paper.init(getContext());
    setGenre();
    if (Paper.book().contains("books")){
        if (books.isEmpty()){
            getView().setVisibility(View.VISIBLE);
            loadDataButton.setVisibility(View.GONE);
            fab.show();
            displayDatabaseInfo();
        }
    } else {
        gridView.setVisibility(View.GONE);
        loadDataButton.setVisibility(View.VISIBLE);
        fab.hide();
    }
    loadDataButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            createBook();
            displayDatabaseInfo();
            gridView.setVisibility(View.VISIBLE);
            fab.show();
            loadDataButton.setVisibility(View.GONE);
        }
    });
}
```

## Продолжение приложения Б

```
Paper.book().write("books", 1);
setGenre();
}
});
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        deleteBook();
        books.clear();
        gridView.setVisibility(View.GONE);
        loadDataButton.setVisibility(View.VISIBLE);
        fab.hide();
        if (Paper.book().contains("books")){
            Paper.book().delete("books");
        }
    }
});
}

@Override
public void onAttach(Context context) {super.onAttach(context);}

@Override
public void onDetach() {
    super.onDetach();
    mListener = null;
}

public interface OnFragmentInteractionListener {
    boolean onNavigationItemSelected(MenuItem item);

    // TODO: Update argument type and name
    void onFragmentInteraction(Uri uri);
}

public class GridViewAdapter extends ArrayAdapter<Book>{

    public GridViewAdapter(@NonNull Context context, int resource,
List<Book> books) {
        super(context, resource, books);
    }
}
```

## Продолжение приложения Б

```
public GridViewAdapter(Context context, int resource) {
    super(context, resource);
}

@Override
public View getView(final int position, View convertView, ViewGroup
parent){
    View v = convertView;
    Book book = getItem(position);
    if(null == v){
        LayoutInflater inflater = (LayoutInflater)
getContext().getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        v = inflater.inflate(R.layout.books_item, null);
    }

    TextView name = v.findViewById(R.id.library_name);
    TextView author = v.findViewById(R.id.library_author);
    ImageView logo = v.findViewById(R.id.library_logo);

    name.setText(book.getName());
    author.setText(book.getAuthor());
    Glide.with(getContext()).load(book.getAvatar()).into(logo);
    v.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(getActivity(), LibraryItemActivity.class);
            if (booksToGridView.size()>1 && booksToGridView.size()<4){
                intent.putExtra("id", position+1);
            } else {
                intent.putExtra("id", position);
            }
            startActivity(intent);
        }
    });
    registerForContextMenu(v);
    return v;
}

private void createBook() {
    SQLiteDatabase db = mDbHelper.getWritableDatabase();
    ContentValues values1 = new ContentValues();
```

## Продолжение приложения Б

```
ContentValues values2 = new ContentValues();
ContentValues values3 = new ContentValues();
ContentValues values4 = new ContentValues();
values1.put(BookCreate.COLUMN_NAME, "Снежная королева");
values1.put(BookCreate.COLUMN_AUTHOR, "Ганс Христиан Андерсен");
values1.put(BookCreate.COLUMN_GENRE,
BookCreate.GENRE_FAIRYTALES);
values1.put(BookCreate.COLUMN_CONTENT,
getString(R.string.snow_queen_chapter1)          +
getString(R.string.snow_queen_chapter2));
values1.put(BookCreate.COLUMN_AVATAR,
"https://images.radario.ru/images/webeventposter/c4eb5489652e40dc97c0562e5ee0
358e.jpg");
values2.put(BookCreate.COLUMN_NAME, "Последний лист");
values2.put(BookCreate.COLUMN_AUTHOR, "О'Генри");
values2.put(BookCreate.COLUMN_GENRE,
BookCreate.GENRE_STORIES);
values2.put(BookCreate.COLUMN_CONTENT,
getString(R.string.the_last_leaf));
values2.put(BookCreate.COLUMN_AVATAR,
"https://files.adme.ru/files/news/part_87/878410/14422760-1-650-d85af6b851-
1484580106.jpg");
values3.put(BookCreate.COLUMN_NAME, "Белые слоны");
values3.put(BookCreate.COLUMN_AUTHOR, "Эрнест Хемингуэй");
values3.put(BookCreate.COLUMN_GENRE,
BookCreate.GENRE_STORIES);
values3.put(BookCreate.COLUMN_CONTENT,
getString(R.string.white_elephants));
values3.put(BookCreate.COLUMN_AVATAR,
"https://writology.com/userdata/writers/267/blog/55fa4826a6bc86cc544c36eb582d2
e208e7362a05dfceeb79e_pimgpsh_fullsize_distr.jpg");
values4.put(BookCreate.COLUMN_NAME, "Который из трех");
values4.put(BookCreate.COLUMN_AUTHOR, "Антон Павлович Чехов");
values4.put(BookCreate.COLUMN_GENRE,
BookCreate.GENRE_STORIES);
values4.put(BookCreate.COLUMN_CONTENT,
getString(R.string.which_of_the_three));
values4.put(BookCreate.COLUMN_AVATAR,
"https://i.livelib.ru/workpic/1000976590/200/29c2/A._P._Chehov__Kotoryj_iz_treh
.jpg");
```

## Продолжение приложения Б

```
List<ContentValues> values = new ArrayList<>();
values.add(0, values1);
values.add(1, values2);
values.add(2, values3);
values.add(3, values4);
db.insert(BookCreate.TABLE_NAME, null, values.get(0));
db.insert(BookCreate.TABLE_NAME, null, values.get(1));
db.insert(BookCreate.TABLE_NAME, null, values.get(2));
db.insert(BookCreate.TABLE_NAME, null, values.get(3));
}

public void deleteBook(){
    SQLiteDatabase db = mDbHelper.getWritableDatabase();
    db.delete(BookCreate.TABLE_NAME,null, null);
}

private void displayDatabaseInfo() {
    SQLiteDatabase db = mDbHelper.getReadableDatabase();

    String[] projection = {
        BookCreate._ID,
        BookCreate.COLUMN_NAME,
        BookCreate.COLUMN_AUTHOR,
        BookCreate.COLUMN_GENRE,
        BookCreate.COLUMN_CONTENT,
        BookCreate.COLUMN_AVATAR};

    // Делаем запрос
    Cursor cursor = db.query(
        BookCreate.TABLE_NAME, // таблица
        projection,           // столбцы
        null,                  // столбцы для условия WHERE
        null,                  // значения для условия WHERE
        null,                  // Don't group the rows
        null,                  // Don't filter by row groups
        null);                // порядок сортировки

    try {
        int idColumnIndex = cursor.getColumnIndex(BookCreate._ID);
        int nameColumnIndex = cursor.getColumnIndex(BookCreate.COLUMN_NAME);
    }
}
```



## Продолжение приложения Б

```

        int                authorColumnIndex                =
cursor.getColumnIndex(BookCreate.COLUMN_AUTHOR);
        int                contentColumnIndex                =
cursor.getColumnIndex(BookCreate.COLUMN_CONTENT);
        int                genreColumnIndex                  =
cursor.getColumnIndex(BookCreate.COLUMN_GENRE);
        int                avatarColumnIndex                  =
cursor.getColumnIndex(BookCreate.COLUMN_AVATAR);
        int i =0;

        while (cursor.moveToNext()) {
            int currentID = cursor.getInt(idColumnIndex);
            String currentName = cursor.getString(nameColumnIndex);
            String currentAuthor = cursor.getString(authorColumnIndex);
            String currentContent = cursor.getString(contentColumnIndex);
            String currentGenre = cursor.getString(genreColumnIndex);
            String currentAvatar = cursor.getString(avatarColumnIndex);
            books.add(i, new Book(currentName,
                currentAuthor, currentContent, currentGenre, currentAvatar));
            i++;
        }
    } finally {
        cursor.close();
    }
}

public void setGenre(){
    final                ArrayAdapter                genderSpinnerAdapter                =
ArrayAdapter.createFromResource(getContext(),
        R.array.array_genre_options, android.R.layout.simple_spinner_item);

    genderSpinnerAdapter.setDropDownViewResource(android.R.layout.simple_dropd
own_item_1line);

    changeGenre.setAdapter(genderSpinnerAdapter);
    changeGenre.setSelection(3);
    changeGenre.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
            String selection = (String) parent.getItemAtPosition(position);

```

## Продолжение приложения Б

```
if (!TextUtils.isEmpty(selection)) {
    if (selection.equals(getString(R.string.genre_ft))) {
        genre = "ft";
    } else if (selection.equals(getString(R.string.genre_s))) {
        genre = "s";
    } else if (selection.equals(getString(R.string.genre_mi))) {
        genre = "mi";
    } else {
        genre = "all";
    }
}
booksToGridView.clear();
gridViewAdapter = new GridViewAdapter(getContext(),
R.layout.books_item);
gridView.setAdapter(gridViewAdapter);
libraryTV.setVisibility(View.GONE);
if(!books.isEmpty()) {
    for (int i = 0; i < books.size(); i++) {
        if (genre != null) {
            if (genre.equals("ft")) {
                if (books.get(i).getGenre().equals("1")) {
                    booksToGridView.add(books.get(i));
                }
            } else if (genre.equals("s")) {
                if (books.get(i).getGenre().equals("2")) {
                    booksToGridView.add(books.get(i));
                }
            } else if (genre.equals("mi")) {
                if (books.get(i).getGenre().equals("3")) {
                    booksToGridView.add(books.get(i));
                }
            } else {
                booksToGridView.add(books.get(i));
            }
        }
    }
}
if (!booksToGridView.isEmpty()) {
    if (booksToGridView.size() < 2) {
        gridView.setNumColumns(1);
    } else {
        gridView.setNumColumns(2);
    }
}
```

## Продолжение приложения Б

```
    }
    gridViewAdapter = new GridViewAdapter(getActivity(),
R.layout.books_item, booksToGridView);
    gridView.setAdapter(gridViewAdapter);
} else {
    libraryTV.setText("No such files :(");
    libraryTV.setVisibility(View.VISIBLE);
}
}

@Override
public void onNothingSelected(AdapterView<?> parent) {

}
});
}
}
```

## NewsFragment

```
public class NewsFragment extends Fragment {
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";
    private String mParam1;
    private String mParam2;
    public static final String TAG = "NewsFragment";
    private OnFragmentInteractionListener mListener;
    WebView mWebView;
    public NewsFragment() {
        // Required empty public constructor
    }

    public static NewsFragment newInstance(String param1, String param2) {
        NewsFragment fragment = new NewsFragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
```

## Продолжение приложения Б

```
super.onCreate(savedInstanceState);
if (getArguments() != null) {
    mParam1 = getArguments().getString(ARG_PARAM1);
    mParam2 = getArguments().getString(ARG_PARAM2);
}
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_news, container, false);
}

public void onPressed(Uri uri) {
    if (mListener != null) {
        mListener.onFragmentInteraction(uri);
    }
}

@Override
public void onStart() {
    super.onStart();
    mWebView = getView().findViewById(R.id.webView);
    openWebView();
}

@Override
public void onAttach(Context context) {super.onAttach(context);}

@Override
public void onDetach() {
    super.onDetach();
    mListener = null;
}

public interface OnFragmentInteractionListener {
    // TODO: Update argument type and name
    void onFragmentInteraction(Uri uri);
}

public void openWebView(){
```

## Продолжение приложения Б

```
mWebView.getSettings().setJavaScriptEnabled(true);
mWebView.getSettings().setLoadWithOverviewMode(true);
mWebView.getSettings().setUseWideViewPort(true);
mWebView.loadUrl("https://aues.kz/?page_id=2");
mWebView.setWebViewClient(new MyWebViewClient());
}
}
```

### Profile Fragment

```
public class ProfileFragment extends Fragment {
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";
    private String mParam1;
    private String mParam2;
    public static final String TAG = "ProfileFragment";
    private OnFragmentInteractionListener mListener;
    ImageView userAvatar;
    TextView userName, projectName;

    public ProfileFragment() {
        // Required empty public constructor
    }

    public static ProfileFragment newInstance(String param1, String param2) {
        ProfileFragment fragment = new ProfileFragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
```

## Продолжение приложения Б

```
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_profile, container, false);
}

public void onPressed(Uri uri) {
    if (mListener != null) {
        mListener.onFragmentInteraction(uri);
    }
}

@Override
public void onStart() {
    super.onStart();
    userAvatar = getView().findViewById(R.id.userAvatar);
    userName = getView().findViewById(R.id.userName);
    projectName = getView().findViewById(R.id.projectName);

    Glide.with(this).load("https://pp.userapi.com/c854416/v854416043/347a0/UoBNTh
bOXaA.jpg").into(userAvatar);
    userName.setText("Alua");
    projectName.setText("Alua's Library");
}

@Override
public void onAttach(Context context) {super.onAttach(context);}

@Override
public void onDetach() {
    super.onDetach();
    mListener = null;
}

public interface OnFragmentInteractionListener {
    // TODO: Update argument type and name
    void onFragmentInteraction(Uri uri);
}
}
```

### **SettingsFragment**

```
public class SettingsFragment extends Fragment {
    public static final String TAG = "SettingsFragment";
```

## Продолжение приложения Б

```
private static final String ARG_PARAM1 = "param1";
private static final String ARG_PARAM2 = "param2";
private String mParam1;
private String mParam2;
private OnFragmentInteractionListener mListener;
Spinner changeLanguage;
TextView changeLanguageTV;
Button changeLanguageButton;
String language;
public SettingsFragment() {}

public static SettingsFragment newInstance(String param1, String param2) {
    SettingsFragment fragment = new SettingsFragment();
    Bundle args = new Bundle();
    args.putString(ARG_PARAM1, param1);
    args.putString(ARG_PARAM2, param2);
    fragment.setArguments(args);
    return fragment;
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (getArguments() != null) {
        mParam1 = getArguments().getString(ARG_PARAM1);
        mParam2 = getArguments().getString(ARG_PARAM2);
    }
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_settings, container, false);
}

public void onPressed(Uri uri) {
    if (mListener != null) {
        mListener.onFragmentInteraction(uri);
    }
}
```

## Продолжение приложения Б

```
@Override
public void onStart() {
    super.onStart();
    Paper.init(getActivity());
    changeLanguage = getView().findViewById(R.id.spinner_language);
    changeLanguageTV = getView().findViewById(R.id.change_language_TV);
    changeLanguageButton =
getView().findViewById(R.id.change_language_button);
    setLanguage();
    if (Paper.book().contains("language")) {
        if (Paper.book().read("language").equals("kz")) {

changeLanguageTV.setText(getString(R.string.change_language_button_kz));
        changeLanguageButton.setText(getString(R.string.accept_kz));
        } else if (Paper.book().read("language").equals("ru")) {

changeLanguageTV.setText(getString(R.string.change_language_button_ru));
        changeLanguageButton.setText(getString(R.string.accept_ru));
        } else {

changeLanguageTV.setText(getString(R.string.change_language_button_en));
        changeLanguageButton.setText(getString(R.string.accept_en));
        }
    }
}

public void setLanguage(){
    ArrayAdapter<String> genderSpinnerAdapter =
ArrayAdapter.createFromResource(getContext(),
        R.array.array_gender_options, android.R.layout.simple_spinner_item);

genderSpinnerAdapter.setDropDownViewResource(android.R.layout.simple_dropd
own_item_1line);

    changeLanguage.setAdapter(genderSpinnerAdapter);
    changeLanguage.setSelection(2);

    changeLanguage.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
```



## Продолжение приложения Б

```
public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
    String selection = (String) parent.getItemAtPosition(position);
    if (!TextUtils.isEmpty(selection)) {
        if (selection.equals(getString(R.string.language_kz))) {
            language = "kz";
        } else if (selection.equals(getString(R.string.language_ru))) {
            language = "ru";
        } else {
            language = "en";
        }
    }
}

@Override
public void onNothingSelected(AdapterView<?> parent) {
}

});
changeLanguageButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (language.equals("kz")) {
            Paper.book().write("language", "kz");
        } else if (language.equals("ru")) {
            Paper.book().write("language", "ru");
        } else {
            Paper.book().write("language", "en");
        }
        intentToMainMenu();
    }
});
}

@Override
public void onAttach(Context context) {super.onAttach(context);}

@Override
public void onDetach() {
    super.onDetach();
    mListener = null;
}
}
```

## Продолжение приложения Б

```
public interface OnFragmentInteractionListener {
    // TODO: Update argument type and name
    void onFragmentInteraction(Uri uri);
}
public void intentToMainMenu(){
    Intent intent = new Intent(getContext(), MainActivity.class);
    intent.putExtra("intent", 1);
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);
    intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
    startActivity(intent);
}
}
```

### LoginActivity

```
public class LoginActivity extends AppCompatActivity {

    EditText loginET, passwordET;
    Button loginButton;
    String login, password;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        loginET = findViewById(R.id.loginET);
        passwordET = findViewById(R.id.passwordET);
        loginButton = findViewById(R.id.login_button);
        login = "Alua";
        password = "password";
        Paper.init(LoginActivity.this);
        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (login.equals(loginET.getText().toString()) &&
password.equals(passwordET.getText().toString())){
                    Paper.book().write("UserLoggedIn", login);
                    Intent intent = new Intent(LoginActivity.this, MainActivity.class);
                    intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
                    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);
                    startActivity(intent);
                }
            }
        });
    }
}
```

## Продолжение приложения Б

```
    } else {
        if(Paper.book().read("language").equals("kz")){
            Toast.makeText(LoginActivity.this,
getString(R.string.wrong_data_kz),Toast.LENGTH_SHORT).show();
        }
        if(Paper.book().read("language").equals("ru")){
            Toast.makeText(LoginActivity.this,
getString(R.string.wrong_data_ru),Toast.LENGTH_SHORT).show();
        }
        else {
            Toast.makeText(LoginActivity.this,
getString(R.string.wrong_data_en),Toast.LENGTH_SHORT).show();
        }
    }
}
});
}
}
```

### **Authorization Activity**

```
public class AuthorizationActivity extends AppCompatActivity {

    EditText loginET, passwordET;
    Button createButton, cancelButton;
    User user;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_authorization);
        loginET = findViewById(R.id.loginET);
        passwordET = findViewById(R.id.passwordET);
        createButton = findViewById(R.id.create_button);
        cancelButton = findViewById(R.id.cancel_button);
        Paper.init(this);
        fillStrings();
        cancelButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AuthorizationActivity.super.onBackPressed();
            }
        });
    }
}
```

## Продолжение приложения Б

```

createButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        user = new User(loginET.getText().toString(),
passwordET.getText().toString());
        Paper.book().write("UserLoggedIn", user);
        Intent intent = new Intent(AuthorizationActivity.this, MainActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);
        startActivity(intent);
    }
});
}

public void fillStrings(){
    if (!loginET.getText().toString().isEmpty() &&
!passwordET.getText().toString().isEmpty()) {
        if (Paper.book().contains("language")) {
            if (Paper.book().read("language").equals("kz")) {
                createButton.setText(getString(R.string.sign_up_kz));
                cancelButton.setText(getString(R.string.cancel_kz));
            } else if (Paper.book().read("language").equals("ru")) {
                createButton.setText(getString(R.string.sign_up_ru));
                cancelButton.setText(getString(R.string.cancel_ru));
            } else {
                createButton.setText(getString(R.string.sign_up_en));
                cancelButton.setText(getString(R.string.cancel_en));
            }
        } else {
            createButton.setText(getString(R.string.sign_up_en));
            cancelButton.setText(getString(R.string.cancel_en));
        }
    } else {
        createButton.setText(getString(R.string.sign_up_en));
        cancelButton.setText(getString(R.string.cancel_en));
    }
}
}
}

```