

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Е. Н. Скуратенко
подпись
«____» _____ 2021 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 Прикладная информатика

Разработка шаблона мобильного приложения

Руководитель _____ зав. кафедрой, канд.техн. наук Е.Н. Скуратенко
подпись, дата

Выпускник _____ А. А. Семенович
подпись, дата

Консультанты
по разделам:

Экономический _____ Е. Н. Скуратенко
подпись, дата

Нормоконтролер _____ В. И. Кокова
подпись, дата

Абакан 2021

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Е. Н. Скуратенко
подпись
«_____» _____ 2021 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Студенту Семенович Артему Александровичу

Группа ХБ 17-03

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: Разработка шаблона мобильного приложения

Утверждена приказом по институту № 222 от 08.04.2021 г.

Руководитель ВКР: зав. кафедрой, канд.техн. наук Е.Н. Скуратенко, ХТИ – филиал СФУ

Исходные данные для ВКР: заказ ХТИ – филиала СФУ.

Перечень разделов ВКР:

1. Анализ предметной области. Выбор средств проектных решений.
2. Описание разработки шаблона.
3. Расчёт затрат и оценка экономической эффективности реализации шаблона.

Перечень графического материала: нет

Руководитель ВКР

подпись

Е.Н. Скуратенко

Задание принял к исполнению

подпись

А. А. Семенович

«08» апреля 2021 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка шаблона мобильного приложения» содержит 74 страницы текстового документа, 54 рисунка, 8 таблиц, 5 формул, 13 приложений, 20 использованных источников.

ШАБЛОН, МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, СТУДЕНТ, ХТИ – ФИЛИАЛ СФУ, ЭМУЛЯТОР, JAVA, FIREBASE, ANDROID STUDIO, ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ, КАПИТАЛЬНЫЕ ЗАТРАТЫ.

Цель выпускной квалификационной работы: разработка шаблона мобильного приложения, содержащего базовые функции.

Задачи:

- исследовать предметную область;
- смоделировать шаблон мобильного приложения;
- разработать шаблон мобильного приложения;
- проверить практическую значимость разработанного шаблона;
- обосновать экономическую эффективность проекта.

В результате были определены основные особенности шаблона, а также из списка самых распространенных элементов приложений были выбраны элементы, которые стали основным функционалом разрабатываемого шаблона. Далее был спроектирован и разработан шаблон мобильного приложения, после чего была проверена практическая значимость разработанного шаблона. Также была обоснована экономическая эффективность и рассчитаны риски проекта.

SUMMARY

The theme of the graduation thesis is «Development of Mobile Application Template». It contains 74 pages of a text document, 54 figures, 8 tables, 5 formulae, 13 appendices, 20 reference items.

TEMPLATE, MOBILE APPLICATION, STUDENT, KhTI - BRANCH OF SibFU, EMULATOR, JAVA, FIREBASE, ANDROID STUDIO, ECONOMIC EFFICIENCY, CAPITAL COSTS.

The purpose of the graduation thesis: development of a mobile application template containing basic functions.

Objectives:

- to investigate the subject area;
- to simulate a mobile application template;
- to develop a template for a mobile application;
- to prove practical significance of the developed template;
- to provide rationale for economic efficiency of the project.

Results: main features of the template have been identified; moreover, the elements being of main functionality of the template have been selected from the list of the most common application elements. The mobile application template has been designed and developed; practical significance of the developed template has been tested. The economic efficiency as well as the risks of the project have been calculated.

English language supervisor

signature, date

Chezybaeva N.V.

full name

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	8
1. Анализ сферы разработки мобильных приложений	10
1.1. Сфера разработки мобильных приложений	11
1.2 Классификация приложений и их составляющих	19
1.3 Выбор технологий и программных средств для разработки	24
1.4 Планирование инфраструктуры. Выбор оборудования	26
1.5 Практическая значимость шаблона	26
1.6 Выводы по аналитическому разделу	27
2 Разработка шаблона мобильного приложения	28
2.1 Требования к мобильному приложению	28
2.2 Установка системы «AndroidStudio»	29
2.3 Внедрение базы данных FirebaseRealtimeDatabase	32
2.4 Установка системы «AndroidEmulator»	35
2.5 Разработка интерфейса шаблона для мобильного приложения ...	37
2.6 Разработка новой версии шаблона приложения	47
2.6.1 Моделирование шаблона приложения	47
2.6.2 Проверка практической значимости разработанного шаблона.	59
2.7 Выводы по проектному разделу	62
3 Экономический раздел разработки шаблона для приложения	63
3.1 Капитальные затраты	63
3.2 Эксплуатационные затраты	67
3.3 Расчет реализации проекта	67
3.4 Определение доходов и экономической эффективности реализации проекта	69
3.5 Оценка рисков проекта	70
3.6 Выводы по экономическому разделу	71
ЗАКЛЮЧЕНИЕ	72

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	73
ПРИЛОЖЕНИЕ А – П.....	75 – 100

ВВЕДЕНИЕ

Во время прогрессивного роста сферы информационных технологий, большая часть современного общества окружает себя атрибутами из этой сферы.

У основной части населения планеты имеется в доме хотя бы один компьютер, ноутбук, смартфон и устройства с искусственным интеллектом (умная колонка, умный дом и т.д.).

На каждом девайсе установлено ПО, выполняющее какие-либо функции, свойственные данному устройству.

Что касается ПО для смартфонов, оно все больше и больше приковывает взгляды, у людей появляется желание придумать свой продукт, от чего рынок программ для смартфонов растет каждый год.

С ростом рынка ПО, растет и количество новых разработчиков. Но начинающие разработчики сталкиваются с трудностями в процессе обучения создания приложений.

Чаще всего, сложности возникают на первом этапе – программировании, этот этап является основой любого приложения, но так как эта сфера только в начале своего пути, теории в свободных источниках мало, поэтому если у разработчика возникают трудности с созданием продукта, то эти трудности, в большинстве случаев, приходится преодолевать самостоятельно.

Этот факт сокращает количество заинтересованных людей и, возможно, количество уникального ПО.

Для того чтобы снизить порог входа в индустрию разработки мобильных приложений, были созданы шаблоны с уже встроенными основными функциями приложения.

Целью выпускной квалификационной работы является разработка шаблона, содержащего основные функции мобильного приложения.

Задачи:

– исследовать предметную область;

- смоделировать шаблон мобильного приложения;
- разработать шаблон мобильного приложения;
- проверить практическую значимость разработанного шаблона;
- обосновать экономическую эффективность проекта.

1. Анализ сферы разработки мобильных приложений

В мире современных технологий гаджеты являются одним из самых необходимых инструментов общества. Их появление сильно упростило жизнь большинства людей.

Самыми распространенными устройствами на данный момент являются смартфоны, их функции значительно расширились с внедрением мобильных приложений, которые стали нормой применения в повседневной жизни.

Рынок мобильных приложений развивается также быстро, как и смартфоны. [1] Сейчас на рынке представлено колоссальное количество ПО для смартфонов, создаются приложения, благодаря которым, например, человек имеет возможность следить за своим здоровьем, питанием, передвижением и т.д. Статистика роста приложений изображена на рисунке 1.

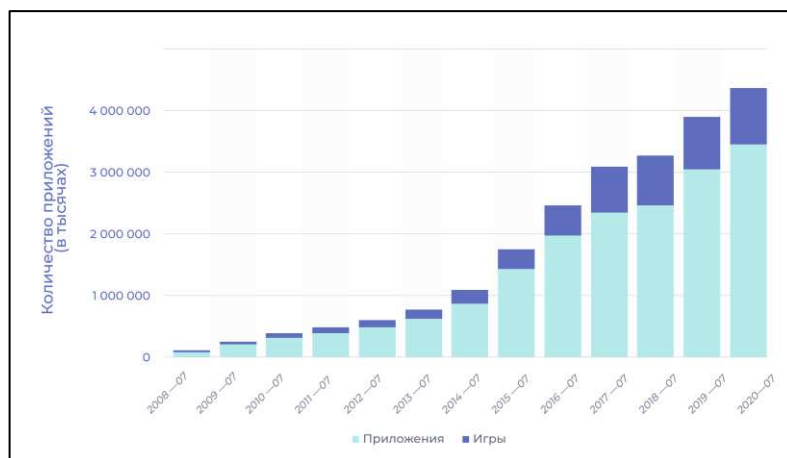


Рисунок 1 – Статистика роста приложений

В данный момент навык создания приложений – это один из самых востребованных навыков, многие люди, интересующиеся сферой IT-технологий, начинают свой путь с создания собственных приложений.

Некоторое время назад, для того чтобы создать самое простое приложение, начинающему разработчику приходилось начинать разработку с самого нуля, затрачивая на это огромное количество времени и сил.

Многие люди хотят получить результаты своей деятельности сразу, хотят быстро создать свое приложение, не вникая в структуру работы алгоритмов и графического оформления, т.к. многие сталкивались с трудностями на этапе изучения программирования, у них пропадал интерес к этому роду деятельности, тем самым индустрия разработки мобильных приложений теряла много потенциальных специалистов.

Чтобы привлечь больше целевой аудитории в данную сферу, опытные разработчики пришли к осознанию того, что нужно упростить данный процесс. Решением данной проблемы стали готовые шаблоны для приложений.

В шаблоны внедряются основные функции приложений, а также создается готовый интерфейс, чтобы начинающие разработчики могли сделать некоторые доработки, тем самым, за короткий промежуток времени получить готовое мобильное приложение.

1.1. Сфера разработки мобильных приложений

Как сказано выше, в настоящий момент разработчик мобильных приложений – это одна из самых распространенных профессий в сфере IT-технологий. Многие студенты, поступающие на специальности в этой сфере, хотят научиться создавать сайты, игры для ПК, мобильные приложения и т.д. В том числе и студенты специальности «Прикладная информатика» в Хакасском техническом институте – филиале СФУ.

Но, к сожалению, курсы разработки мобильных приложений не входят в перечень компетенций выпускника данного направления, а онлайн курсы данного направления очень дорогие. Рассмотрим несколько примеров.

Существует три основных платформы для обучения разработчиков мобильных приложений: GeekBrains [2], Skillbox[3], Яндекс.Практикум [4]. Информация о курсах указана на таблице 1.

Таблица 1 – Информация о курсах

Наименование	GeekBrains	Skillbox	Яндекс.Практикум
Результат обучения (навыки)	Навык разработки мобильных приложений для ОС Android	Навык разработки мобильных приложений для ОС Android	Навык разработки мобильных приложений для ОС Android
Стоимость	5833 р/мес	4692 р/мес	120 000р

Исходя из этих данных, можно сделать вывод, что среднестатистический студент или выпускник специальности «Прикладная информатика», не имеет возможности оплачивать данные курсы, даже с учетом рассрочки, самостоятельно.

Данный факт указывает на то, что у если студента или выпускника этой специальности появится желание обучиться профессии разработчика мобильных приложений, то обучаться он будет самостоятельно.

Одним из таких выпускников являюсь я, сфера разработки android-приложений заинтересовала меня на 4-м курсе обучения, многие аспекты профессии повлияли на мою заинтересованность, рассмотрим некоторые из них:

1. Заработная плата.

Заработная плата разработчика android-приложений по данным сайтов «hh.ru» и «ru.jobble.org» варьируется от 45 тысяч рублей и выше (таблица 2), все зависит от уровня разработчика и от его способностей.

Таблица 2 – Заработная плата разработчика мобильных приложений

	ru.jobble.org	Rabota.yandex.ru	hh.ru
Требования	1.Опыт разработки мобильных приложений. 2.Знания по React, Cordova, js	1.Знание Java / Kotlin, Android SDK, unit- тестирование.	1.Обладать опытом коммерческой разработки Android приложений на Java или Kotlin; 2.Знать гайдлайны Android;

Продолжение таблицы 2

	3.Примеры реализованных приложений	2.Знание архитектуры и принципов разработки Android приложений. 3.Опыт работы с мобильными БД (Room/SQLite) 4.Владение английским языком, достаточное для чтения документации (профессиональная терминология)	3.Владеть принципами объектно-ориентированного программирования; 4.Знать классические алгоритмы и структуры данных; 5.Аккуратно относиться к чужому коду; 6.Знать технический английский язык; 7.Уметь работать в команде.
Заработная плата	От 45 000р	От 75 000р	До 200 000р

2. Возможность удаленной работы.

Одним из плюсов вакансии является возможность работать удаленно, этот аспект безусловно является одним из главных преимуществ данной работы, он гарантирует возможность работать из любой точки мира и, что не менее важно, в комфортных условиях.

3. Взаимодействие с другими странами.

Профессия разработчика мобильных приложений объединяет десятки тысяч специалистов со всего земного шара, при хорошем уровне иностранного языка, появляются перспективы работать в иностранной компании, так же появляется возможность получения большего количества информации и навыков от представителей ведущих стран, в сфере IT-технологий.

4. Изучение английского языка.

Разработчики программируют на языках Java и Kotlin, большинство команд написаны при помощи английских слов, что способствует частичному

изучению английского и, в перспективе, подталкивает к полноценному изучению языка.

По данным App Annie, отрасль мобильных приложений создала колоссальные \$41,1 млрд валового годового дохода, ожидается рост этого показателя до \$50,9 млрд [5]. Согласно прогнозам Statista, в 2020 году валовой годовой доход превысит \$189 млрд. Несмотря на то, что данные разных исследователей немного отличаются, общий вывод таков: рынку далеко до насыщения. Прогнозы App Annie подтвердились в отчетах Forrester о том, что к концу 2016 года лишь 46% населения мира были владельцами смартфонов. Это свидетельствует о том, что широко обсуждаемая мобильная революция только начинается.

По статистике Forrester, отрасль демонстрирует огромный разрыв между ведущими компаниями, для которых мобильные устройства стали катализатором преобразования их бизнеса, и компаниями, которые относятся к мобильным устройствам просто как к еще одному направлению для развития. В начале 2016 года только 18% опрошенных компаний относились к первой категории. Этот показатель, как ожидается, преодолеет 25% уже в следующем году.

Потребители эволюционируют еще быстрее, чем бизнес. Сегодня мобильный интернет для многих пользователей стал насущной необходимостью.

Рынок мобильных приложений весьма разнообразен, чтобы понимать, какой вид приложений более востребован, их разделили на категории [6].

Выделяют пять основных категорий:

- контентные приложения;
- бизнес-приложения;
- игровые приложения;
- промо-приложения;
- социальные сети.

Кроме того, существует классификация приложений по типу монетизации:

- 1) Free – бесплатные;
- 2) Paid – платные;
- 3) Free+In-app purchase – бесплатные приложения с возможностью приобретения виртуальных благ, связанных с данным приложением;
- 4) Paid+ In-app purchase – комбинация платного приложения (оплата при скачивании) и возможности приобретения виртуальных благ, связанных с данным приложением;
- 5) Trial – платное приложение с бесплатной пробной версией.

Рассмотрим каждую категорию более детально.

1. Контентные приложения.

Приложения данного типа предоставляет пользователю доступ к различному контенту, например, пользователь получает возможность слушать музыку, смотреть видеоматериалы, фотоматериалы, читать цифровые книги и т.д.

Хорошим примером является приложение Flipboard, которое, с одной стороны, немного похоже на печатную газету, с другой стороны — имеет все элементы цифрового издания (рисунок 2).



Рисунок 2 – Приложение Flipboard

2. Бизнес-приложения.

Бизнес-приложения, как правило, направлены на выполнение узконаправленного потока работ. С помощью данной категории приложений пользователь может организовывать работу компании, повышать прибыль и разрабатывать бизнес-стратегии. В большинстве случаев данная деятельность осуществляется через игровую призму. Руководитель компании, при помощи данных приложений, имеет возможность получать необходимую информацию об организации.

Одним из таких приложений является Mailchimp, оно предоставляет возможность управления списками рассылки на ходу или получения доступа к своей аналитике, когда это необходимо, с помощью простого в использовании приложения (рисунок 3).

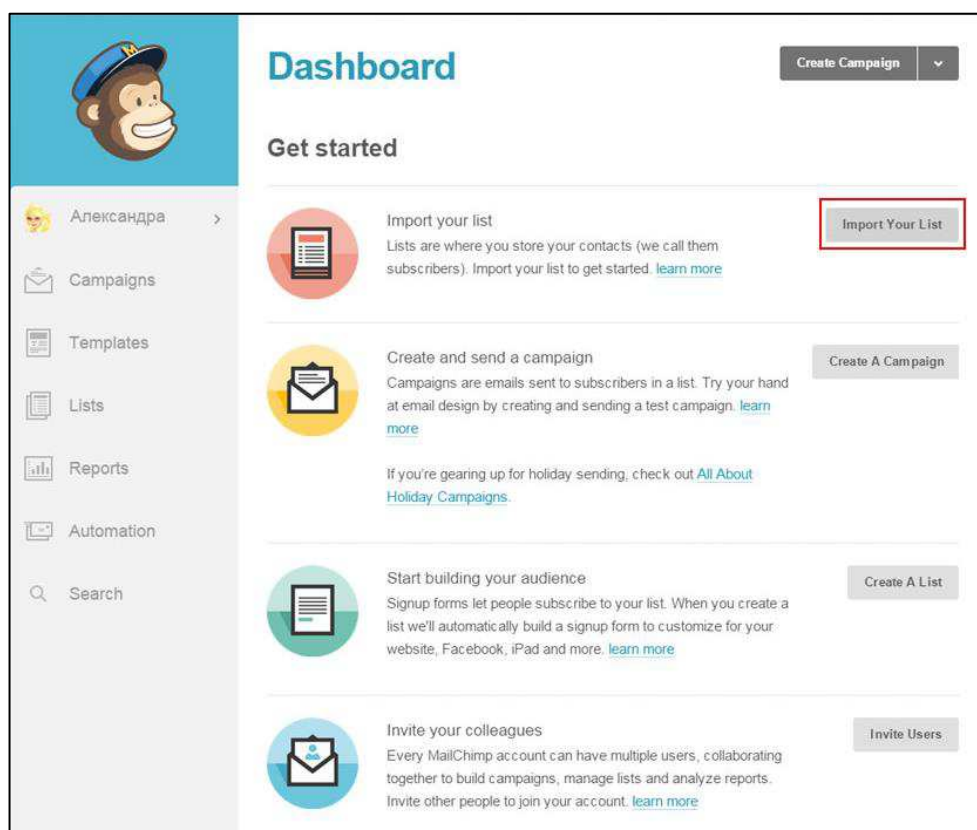


Рисунок 3 – Приложение Mailchimp

3. Игровые приложения.

Мобильные игры – самая большая категория приложений на рынке. Существуют сотни жанров игр, миллионы уже созданных и столько же находящихся в разработке. Данная категория постоянно набирает обороты, и большинство разработчиков выбирают именно это направление, ежедневно на площадках публикуются сотни игровых приложений, т.к. в данном сегменте находится наибольшее количество пользователей (рисунок 4).



Рисунок 4 – Игровые приложения

4. Промо-приложения.

Данные приложения могут создаваться, опираясь на любые темы. Главная их черта – это оригинальность и креативность. Они не несут за собой никакой особой смысловой нагрузки. Вся их суть сводится к тому, что отвлечь и развлечь потребителя.

Например, мобильное приложение от Coca-Cola (рисунок 5).

Помимо регистрации кодов и заработка баллов для обмена на приятные призы, это затягивающая игра.



Рисунок 5 – Мобильное приложение от Coca-Cola

5. Социальные сети.

Поскольку у данной категории на сегодняшний момент наибольшая востребованность, многим людям социальные сети нужны для общения, бизнеса и развлечения, поэтому в мобильных устройствах такая возможность также пользуется популярностью.

Например, приложение «ВКонтакте» (международное название: VK) — российская социальная сеть. «ВКонтакте» позволяет пользователям отправлять друг другу сообщения, создавать собственные страницы и сообщества, обмениваться изображениями, тегами, аудио- и видеозаписями, играть в игры. Приложение «ВКонтакте» изображено на рисунке 6.

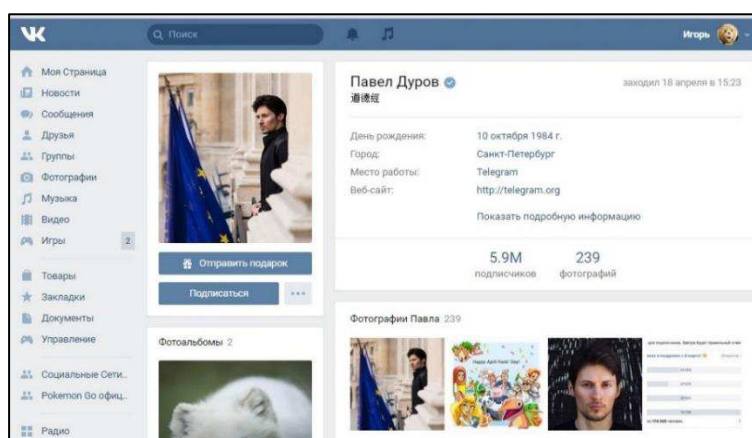


Рисунок 6 – Приложение «ВКонтакте»

1.2 Классификация приложений и их составляющих

Рассматривая основные категории приложений, стоит обратить внимание на их основные элементы. Основные категории и их элементы указаны на таблице 3.

Таблица 3 – Категории приложений

Название категории	Контентные приложения	Бизнес-приложения	Игровые приложения	Промо-приложения	Социальные сети
Основные функции	1.Предоставление контента по запросам пользователя 2.Рекомендации контента, частично подходящего пользователю 3.Распространение контента в сети	1.Контроль данных в компании 2.Рассылка информации сотрудникам 3.Учет деятельности и предприятия	Обучение и развлечение пользователя	Отвлечение пользователя от чего-либо	1.Передача и хранение текстовых и голосовых сообщений 2.Показ видео и аудио контента 3.Предоставление видеосвязи
Основные элементы	1.Базы данных 2.Видеоплееры 3.Аудиоплееры 4.Любой цифровой контент	1.Базы данных. 2.Мессенджер	1.Базы данных 2.Игровой контент 3.Аудио контент	1.Игровой контент. 2.Видео контент	1.Базы данных 2.Мессенджер 3.Аудио и видео контент

Также стоит обратить внимание на классификацию баз данных в мобильных приложениях [7].

Существует два типа баз данных интегрируемых в мобильные приложения:

1. Облачная база данных.

Под удаленной БД подразумевается база, находящаяся на сервере, все данные, которые с помощью приложения предоставляются в мобильном приложении, находятся в удаленном хранилище, тем самым, данные не заполняют память устройства [8].

Данный тип БД имеет преимущество перед встроенной в скорости обновления информации, база может пополняться, и в тот же момент пользователь будет иметь возможность данной информацией воспользоваться.

Недостатком этого типа является необходимость постоянного подключения к хранилищу.

2. Встроенная база данных.

Этот тип БД рассчитан на небольшой объем статичной информации, которой пользователь может воспользоваться. Данная база устанавливается на устройство вместе с приложением.

Преимуществом данного типа является возможность использования приложения без подключения к сети.

Недостатками данного типа являются более высокая загруженность памяти устройства, а также, при добавлении нового блока информации разработчиком, появляется необходимость загрузки обновления приложения.

Стоит отметить, что большинство приложений имеют оба этих типа БД, например, в большое количество программ встроена авторизация пользователя, это необходимо для сохранения прогресса пользователя приложения.

Помимо этого, необходимо выделить некоторые разновидности баз данных по контенту.

1. Текстовые базы данных.
2. Базы данных, содержащие фото и видео контент.
3. Базы данных, содержащие аудио контент.
4. Базы данных, содержащие смешанный контент.

Базы данных могут содержать сотни и тысячи записей. Часто бывает необходимо упорядочить записи, то есть расположить в определенной последовательности. Упорядочение записей называется сортировкой. Сортировка записей производится по какому-либо полю. Значения, содержащиеся в этом поле, располагаются в определенном порядке, который определяется типом поля: по алфавиту, если поле текстовое; по величине числа, если поле числовое; по дате, если тип поля - Дата/Время и так далее.

Сортировка записей может производиться либо по возрастанию, либо по убыванию значений поля. В процессе сортировки целостность записей сохраняется, то есть они переносятся из одного места таблицы в другой целиком. Сортировка записей базы данных - это их упорядочение по значениям одного из полей [9].

Фильтры позволяют отбирать записи, которые удовлетворяют заданным условиям. Условия отбора записей создаются с использованием операторов сравнения (=, >, < и так далее).

Простые фильтры содержат условие отбора записей только для одного поля.

Сложные фильтры содержат несколько условий для различных полей. В результате применения сложного фильтра будут отобраны только те записи, которые удовлетворяют всем условиям одновременно. Можно сказать, что условия в сложных фильтрах связаны между собой операцией логического умножения.

На таблице 4 отображены виды фильтрации и их принадлежность к видам баз данных.

Таблица 4 – Виды фильтрации и их принадлежность к видам баз данных

	Числовой	По тегам	По дате и времени	По совпадению запросов	По похожим запросам	По рейтинг у	По возраста нию/убы ванию
--	----------	-------------	-------------------------	------------------------------	---------------------------	--------------------	------------------------------------

Продолжение таблицы 4

Текстовые базы данных	+	+	+	+	+	-	+
Базы данных, содержащие фото и видео контент	-	+	+	+	+	+	+
Базы данных, содержащие аудио контент	-	-	+	+	+	+	-
Базы данных, содержащие смешанный контент	+	+	+	+	+	+	+

Рассмотрим виды фильтрации и их принадлежность к видам баз данных:

– Числовой

Данный фильтр распределяет информацию по значению числа в заданной разработчиком системе исчислений.

– Фильтр по тегам

В некоторых приложениях блокам контента присваивают теги, т.е. слова, по которым можно найти все подходящие по запросу блоки информации.

– Фильтр по дате и времени

Фильтрация по дате и времени является одной из самых частых сортировок по базам данных.

– Фильтр по совпадению запросов

Данный фильтр похож по алгоритму на фильтрацию по тегам, но поиск происходит, по ключевым словам, например, по названию, имени, фамилии и т.д.

– Фильтр по похожим запросам

Фильтрация по похожим запросам происходит на основе синонимов той информации, по которой пользователь запрашивает информацию, в большинстве случаев это происходит для подборки рекомендаций других блоков информации, схожих по запросу пользователя.

– Сортировка по рейтингу

В приложениях нередко используется система оценок, на основе чего производится составление рейтинга информации, например, в приложениях площадок по продаже товаров существует система рейтинга продавцов, который в свою очередь составляется на основе оценок и отзывов покупателей.

– Сортировка по убыванию или возрастанию значения

Данный вид сортировки используется, в основном, для отслеживания статистики какого-либо блока информации.

Обращаясь к данным таблицы 2, можно понять, что большинство видов баз данных так или иначе имеют несколько типов фильтрации, что говорит о многозадачности БД.

При создании собственного проекта, необходимо определить основной функционал приложения. Стоит изучить приложения разных классификаций, найти основные сходства и внедрить их в свой продукт.

По статистике, чаще всего внедряются такие компоненты:

1. Новостная лента.
2. Отправка и принятие сообщений.
3. Поиск и предоставление данных.
4. Аудио и видеоплеер.
5. Уведомления.

Исходя из того, что разработанный шаблон мобильного приложения является универсальным, в него были внедрены базовые функции приложений, такие как:

1. Добавление информации в базу данных.
2. Чтение информации из базы данных.

3. Разделение информации на категории.
4. Возможность перехода из приложения на требуемый сайт через прямую ссылку.
5. Настройка интерфейса, исходя из настроек девайса.

1.3 Выбор технологий и программных средств для разработки

Для реализации программы необходимы следующие программные продукты: операционную систему, для которой будет создаваться приложение, среду разработки для создания приложения, сервер для хранения данных и обработки запросов.

ПО создается на ОС Android, т.к. исследовательская фирма Consumer Intelligence Research Partners (CIRP) опубликовала отчет о состоянии рынка мобильных операционных систем по итогам первого квартала 2020 года.[10] В частности, аналитики подсчитали, сколько было активировано новых устройств под управлением iOS и Android, а также определили лояльность пользователей к каждой ОС

Сразу стоит отметить, что статистика была собрана по рынку США, поэтому в других регионах и во всём мире в целом цифры могут несколько отличаться. По данным CIRP, за первый квартал 2020 года 56% всех активированных новых смартфонов работали под управлением Android. Следовательно, доля iOS составила оставшиеся 44%.

Исходя из этого выбор сделан в пользу Android, больше пользователей – больше шансов у ПО стать популярным.

В качестве сетевого хранилища баз данных была выбрана сетевая база данных Firebase. Firebase служит базой данных, которая изменяется в реальном времени и хранит данные в JSON. Любые изменения в базе данных тут же синхронизируются между всеми клиентами, или девайсами, которые используют одну и ту же базу данных. Вместе с хранилищем, Firebase также предоставляет пользовательскую аутентификацию, и поэтому все данные

передаются через защищенное соединение SSL. Можно выбрать любую комбинацию e-mail и пароля для аутентификации, будь то Facebook, Twitter, GitHub, Google, или что-то другое. Так же FireBase предоставляет хранилище данных, в котором могут храниться всевозможные фотографии, видео или документы.

Для разработки приложения были рассмотрены две программы: Androidstudio и Eclipse.

Android Studio – программа, являющаяся средой разработки приложений для мобильной платформы Android [11].

Android Studio превосходит конкурента по нескольким параметрам, к которым можно отнести:

- гибкость среды разработки;
- большой набор функций;
- процесс разработки, который подстраивается под разработчика.

Во время создания приложений и утилит для операционной системы Android, пользователь программного обеспечения может наблюдать за изменениями в проекте, в режиме реального времени.

Android Studio – универсальная среда разработки, так как позволяет оптимизировать работу будущих приложения для работы не только на смартфонах, но и на планшете, портативных ПК, которые работают на основе рассматриваемой операционной системы [12].

В итоге выбор сделан в пользу Android Studio.

В этой среде разработки предоставлена возможность программировать на двух языках: Java и Kotlin.

Kotlin и Java — это в первую очередь взаимодополняющие языки. Безусловно, некоторые функции лучше реализовывать в Kotlin, а некоторые практичнее было бы использовать в Java, несмотря на это, оба языка полностью совместимы и способны компилироваться в байт-код. Kotlin в данном случае полностью использует JVM экосистему и библиотеку. Что позволяет предельно просто пользоваться Java методами и классами [13].

То, что они совместимы между собой, помогает при начале работы с Kotlin, позволяя периодически внедрять код Kotlin в программы, написанные на Java.

Исходя из того факта, что Kotlin является более лаконичным и простым языком, начинающим разработчикам стоит начинать свой путь с языком Java, т.к. программирование на этом языке будет более понятно с точки зрения структуры работы кода.

1.4 Планирование инфраструктуры. Выбор оборудования

Инфраструктура системы предполагает использование 1 клиента, а также облачную базу данных. Обмен данными будет происходить посредством сети Интернет, с помощью защищенных и зашифрованных протоколов защиты, которые предоставляет облачный сервис баз данных.

Для работы с пользовательским приложением требуется смартфон. Также требуется, чтобы операционная система устройства была актуальной (не ниже Android4.4Kitkat) [14].

Распространяться шаблон будет посредством маркетплейсов, созданных для продажи шаблонов, площадки, на которых каждый желающий сможет приобрести себе данный шаблон, и создать на его основе полноценное приложение.

Исходя из вышеперечисленного, можно сделать вывод, что проект будет создан с облачным хранилищем на операционной системе Android 4.4 Kitkat для того, чтобы большинство девайсов получили доступ к шаблону.

1.5 Практическая значимость шаблона

Для того чтобы проверить практическую значимость разработанного шаблона, было решено создать на его основе мобильное приложение

«Teacher`snotes», имеющее цель донесение необходимой информации от преподавателей до студентов направления «Прикладная информатика».

Идеей создания этого приложения стала актуальная проблема с дистанционным обучением. В любой момент ВУЗы могут снова перевести на дистанционную форму обучения, и преподавателям будет необходимо доводить информацию до студентов через интернет.

В большинстве случаев информация передается при помощи социальных сетей, но есть вероятность столкновения с проблемой отсутствия аккаунтов со стороны как студентов, так и преподавателей в данных социальных сетях. Также стоит отметить, что не каждый студент имеет желание добавлять в «друзья» своих преподавателей, а также некоторые из них хотят сохранить свою анонимность в сети.

Исходя из этих фактов, возникла идея создания приложения без регистрации и авторизации пользователей, в котором преподаватель сможет донести любую информацию для студентов.

1.6 Выводы по аналитическому разделу

Исходя из информации, полученной в аналитическом разделе, были определены основные особенности шаблона, а также из списка самых распространенных элементов приложений были выбраны элементы, которые стали основным функционалом разрабатываемого шаблона.

2 Разработка шаблона мобильного приложения

Перед тем как начать разработку приложения, необходимо составить план поэтапной разработки проекта.

План разработки приложения:

- 1.Определение основных требований к мобильному приложению.
- 2.Установка среды разработки «AndroidStudio».
- 3.Подключение базы данных «FirebaseRealtimedatabase» к проекту.
- 4.Установка вспомогательной системы «AndroidEmulator».
5. Разработка интерфейса шаблона приложения.
- 6.Создание мобильного приложения на основе шаблона для ХТИ – филиала СФУ.

2.1 Требования к мобильному приложению

Требования к системе зависят от назначения, вида и выполняемых системой функций.

Основными требованиями являются следующие:

- 1.Для создания мобильного приложения необходимо использовать языки системы Android.
- 2.Для эксплуатации данного приложения необходимо устройство с установленной системой Android.

Основным требованием к надежности системы является обеспечение целостности информации, находящейся в базе данных.

Также к надежности относятся следующие требования:

- все функции системы должны работать исправно;
- система не должна содержать ошибок, искажающих внешние спецификации, обнаруженных тестированием;
- база данных должна корректно записывать и выводить данные, при работе с системой.

Программный продукт выполняет следующие функции:

- сохранять в базу данных информацию, которую пользователь вносит, выбирая один из разделов в sub-меню «Новая запись»;

В sub-меню «Новая запись» пользователю необходимо выбрать один из трех разделов и внести информацию в базу данных, посредством приложения. Пользователь заполняет необходимые поля, после чего кнопкой «Добавить запись» вносит эту информацию в БД.

- выводить информацию, которая содержится в базе, а также осуществлять поиск необходимой информации через функцию поиска;

В sub-меню «Поиск», пользователю выводятся три категории, каждая из которых хранит в себе информацию, не имеющую отношения к другим категориям. После выбора нужной категории, система отображает всю информацию, содержащуюся в ней. Если БД содержит большое количество записей, что осложняет возможность поиска для пользователя, в данном окне внедрена функция поиска, при помощи которой, пользователь может осуществлять поиск необходимого блока информации через поисковую строку.

- переходить на официальный сайт ХТИ – филиала СФУ напрямую в приложении.

В sub-меню «Сайт ХТИ – филиала СФУ» создаётся окно, в котором через прямую ссылку на сайт института пользователь имеет прямой доступ к сайту, а также возможность получить информацию, которой нет в базе приложения.

2.2 Установка системы «AndroidStudio»

Для разработки шаблона мобильного приложения необходимо установить среду разработки, в которой создаются проекты приложений.

Для начала необходимо перейти на официальный сайт AndroidStudio «<https://developer.android.google.cn/studio?hl=es>» и нажать на кнопку «Download AndroidStudio» [15]. Официальный сайт AndroidStudio изображен на рисунке 7.

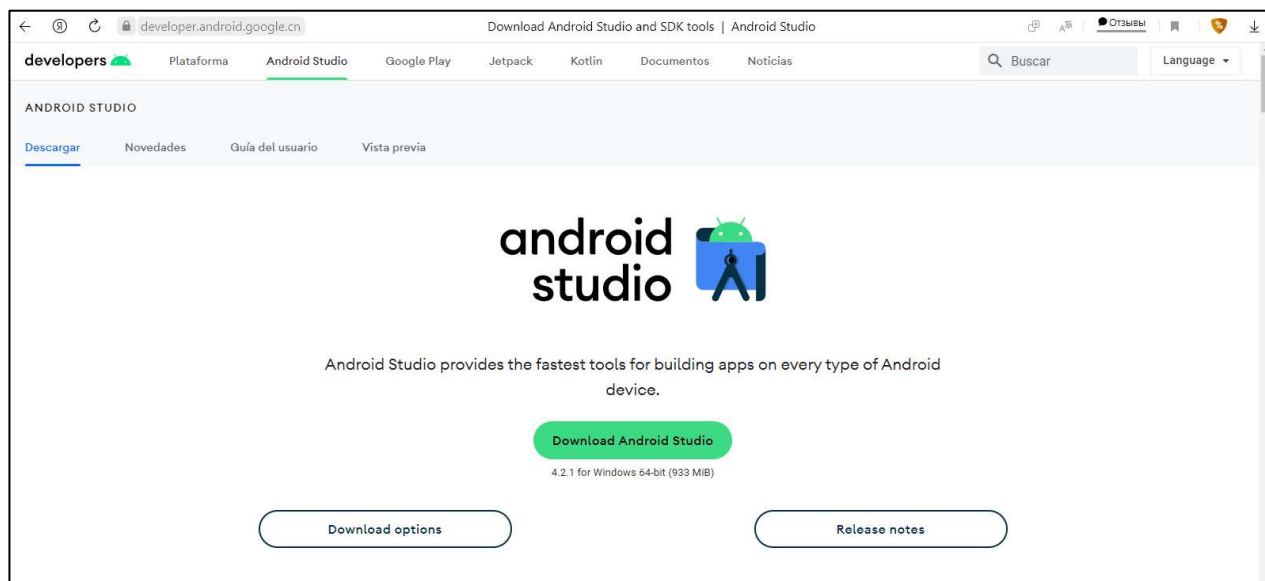


Рисунок 7 – Официальный сайт AndroidStudio

После данного действия на компьютер будет загружен установочный файл.

После установки AndroidStudio необходимо создать новый проект, в котором будет создаваться приложение. Окно создания нового проекта изображено на рисунке 8.

В окне NewProjectпредоставлены разные виды начального экрана приложения, если ни один из вариантов не подходит, на этот случай существует пустой бланк «EmptyActivity». Помимо этого, в окне создания проекта возможен выбор устройства, для которого создается приложение.

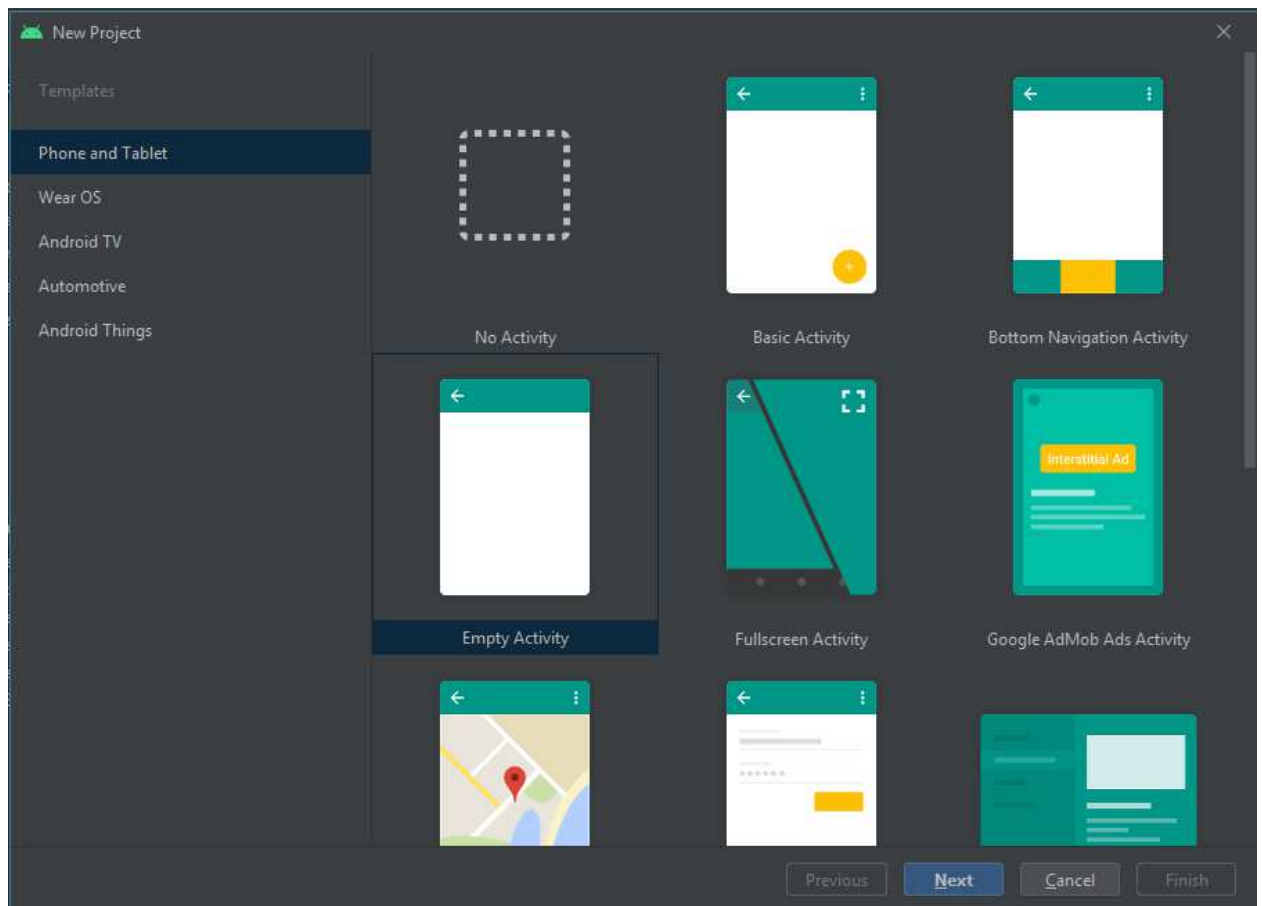


Рисунок 8 – Окно создания нового проекта

После выбора основного экрана приложения, система предлагает написать название проекта (рисунок 9).

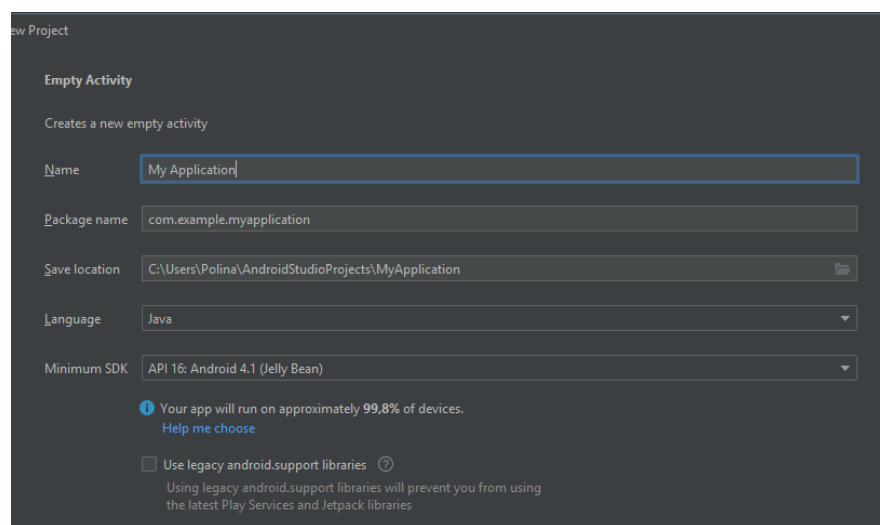


Рисунок 9 – Окно названия проекта

2.3 Внедрение базы данных Firebase Realtime Database

Для того чтобы создать базу данных, необходимо зайти на сайт <https://firebase.google.com> и нажать на кнопку «Getstarted» (рисунок 10).

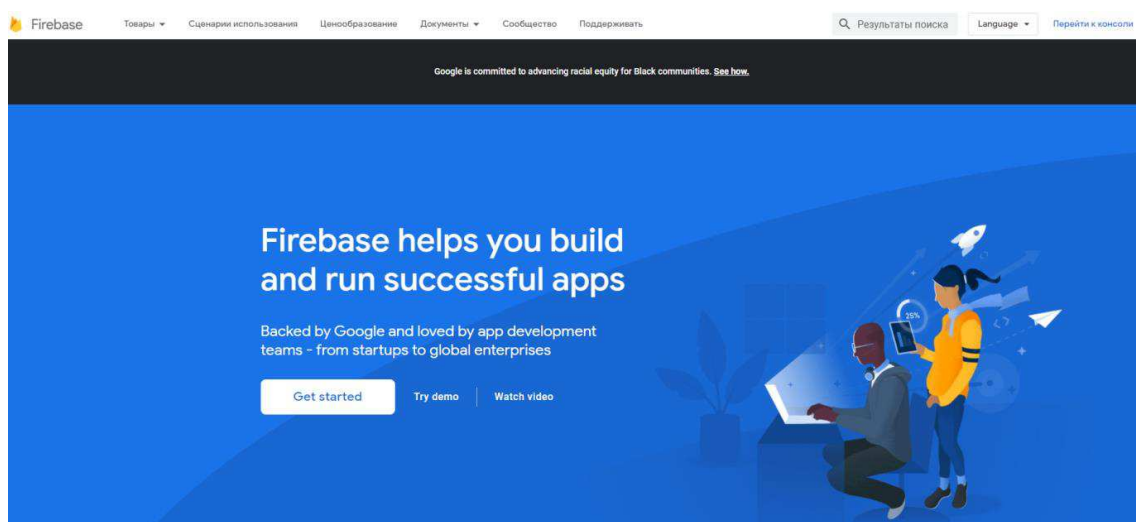


Рисунок 10 – Сайт Firebase

Следующим этапом необходимо войти в свой google-аккаунт (рисунок 11).

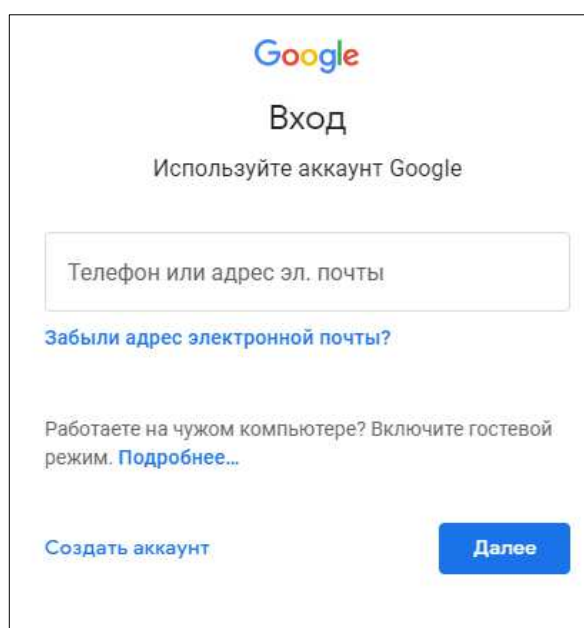


Рисунок 11 – Авторизация в Google

После авторизации в системе, на экран выводится окно, в котором необходимо ввести название своего проекта (рисунок 12).

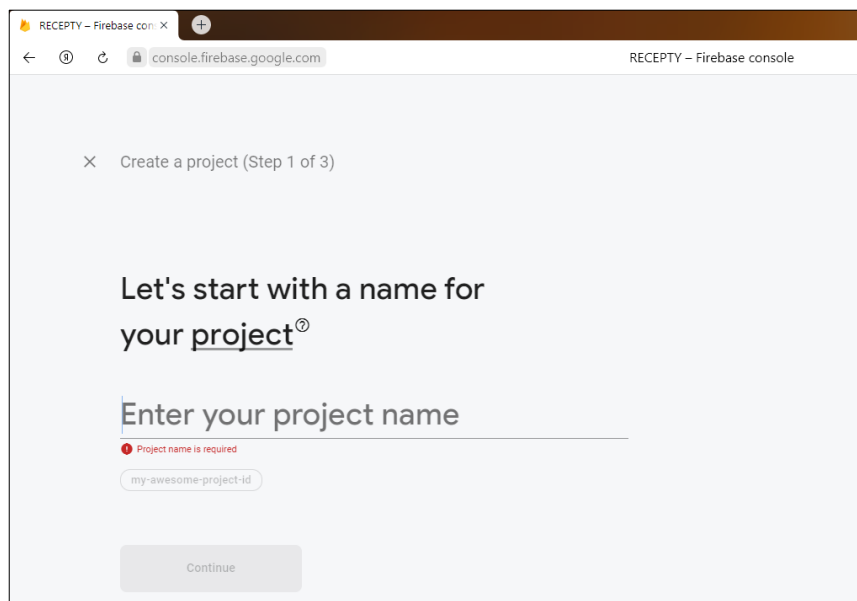


Рисунок 12 – Создание проекта firebase

Далее система предлагает подключить функцию GoogleAnalytics, для отслеживания статистики проекта (рисунок 13).

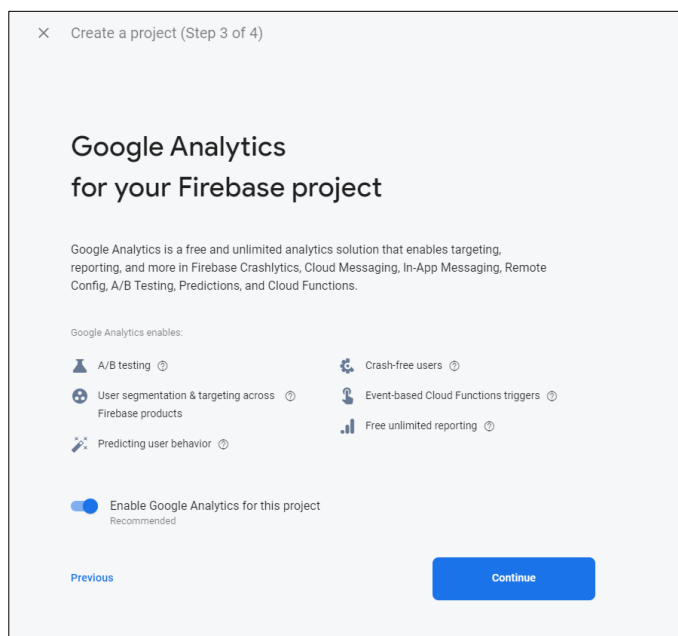


Рисунок 13 – Подключение GoogleAnalytics

После подключения функции GoogleAnalytics, система создает проект и предоставляет список функций, которые можно внедрить в свой проект.

Список функций представлен на рисунке 14.

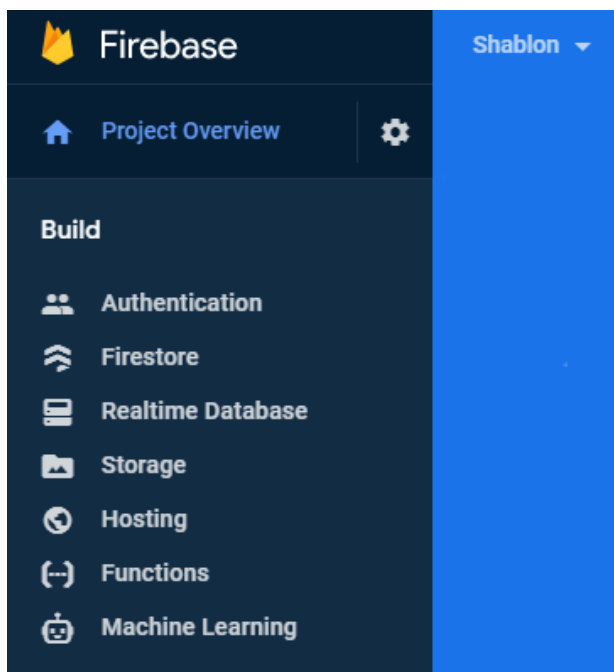


Рисунок 14 – Список функций проекта

Чтобы в проект внедрить облачную базу данных, нужно выбрать функцию «Realtime Database» и нажать на кнопку «Create Database» (рисунок 15).



Рисунок 15 – Создание базы данных

Система создает пустую базу данных, которая в дальнейшем будет заполнена при помощи шаблона, созданного в Androidstudio (рисунок 16).

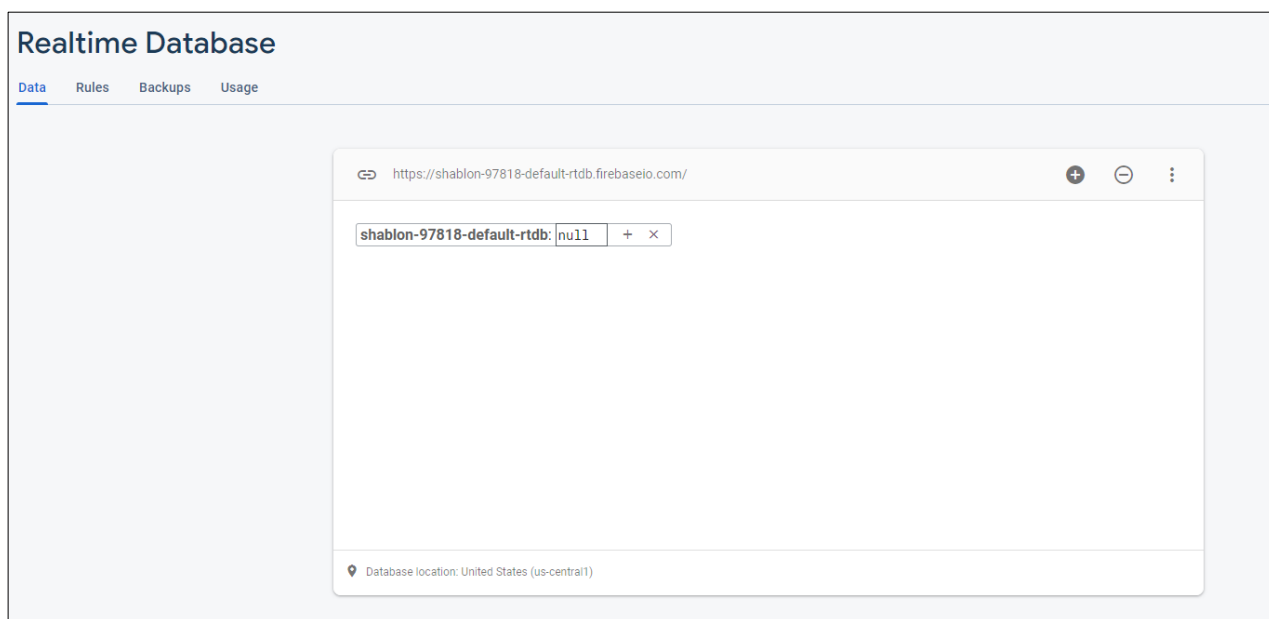


Рисунок 16 – Пустая база данных

После этого, в AndroidStudio база данных привязывается к проекту, после чего можно приступить установке системы «Android Emulator» и к разработке интерфейса шаблона для мобильного приложения.

2.4 Установка системы «AndroidEmulator»

Для установки «AndroidEmulator» была запущена программа «AndroidStudio». При запуске в открывшемся окне было раскрыто выпадающее меню вкладки «Configure» и выбран пункт меню «ADV Manager» (Рисунок 17).



Рисунок 17 – AVDManager

В открывшемся окне с помощью функции «CreateVirtualDevice» была выбрана модель телефона «Pixel 2» с размером 5 дюймов (рисунок 18).

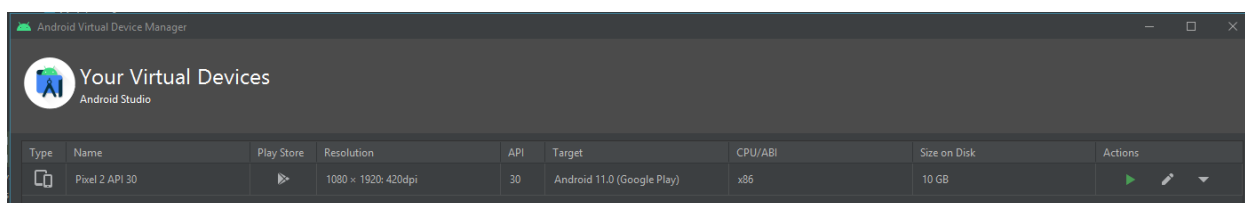


Рисунок 18 – Выбор модели «Pixel 2»

После создания эмуляции смартфона «Pixel 2», был запущен созданный «AndroidEmulator», который представлен на рисунке 19.



Рисунок 19 – AndroidEmulator

2.5 Разработка интерфейса шаблона для мобильного приложения

Основным процессом в создании приложения является разработка интерфейса. Этот процесс делится на два основных этапа:

а) Этап создания визуальной части проекта.

На этом этапе разработчик настраивает положение и размер элементов в приложении, выбирает цветовые палитры, темы, размер и шрифт текста. Также важно отметить, что существует несколько видов экрана, в среде разработки AndroidStudio они называются «Layout». Каждый из них отличается особенностями настройки и функциями [16]. Рассмотрим некоторые из них:

1) LinearLayout – этот Layout отображает элементы в зависимости от того, какую ориентацию задал разработчик. Ориентация бывает «vertical» и «horizontal», первая отображает элементы в виде столбца, а вторая в виде строки.

2) TableLayout – отображает элементы в виде таблицы.

3) RelativeLayout – этот Layout дает возможность настраивать положение элементов относительно друг друга, т.е. при изменении положения одного элемента, за ним тянутся и остальные.

4) AbsoluteLayout – для каждого элемента на этом Layout указывается позиция в системе координат (x,y).

5) ConstraintLayout – базовый и самый распространенный Layout, позволяющий привязать элементы к границам экрана, а также позволяет привязывать элементы друг к другу.

б) Этап программирования.

На этом этапе разработчик пишет код, который в свою очередь оживляет всю визуальную составляющую проекта.

Приступая к визуальной части проекта, стоит отметить, что приложение состоит из многих файлов, большинство из которых были созданы собственноручно, но при этом система предоставляет все удобства для этого, например, все файлы, которые содержатся в проекте, можно увидеть в навигационном поле (рисунок 20).

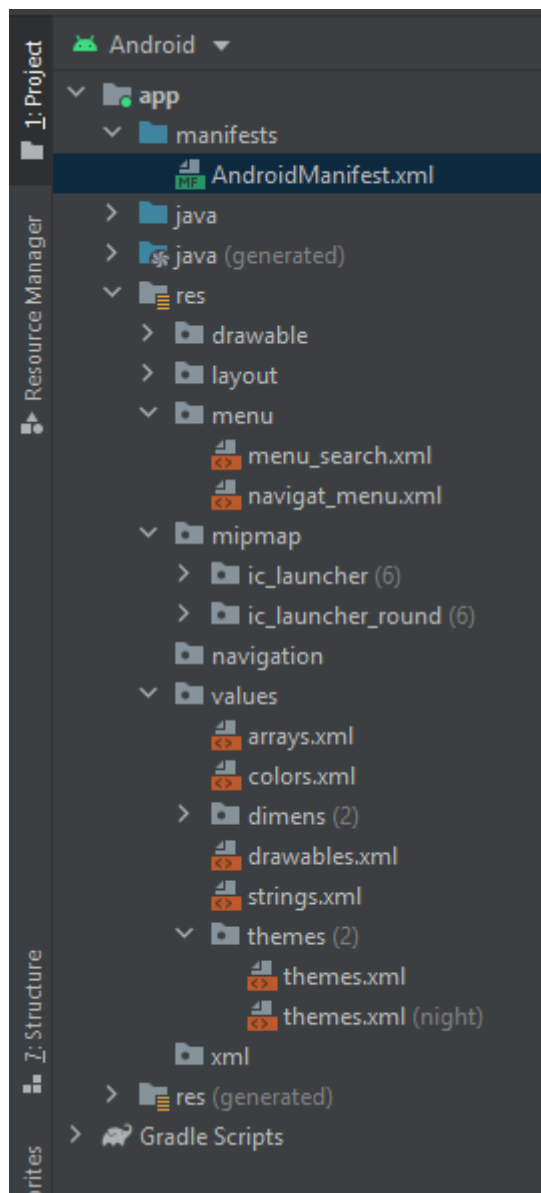


Рисунок 20 – Структура проекта

Первым шагом создания приложения был выбор главного экрана, так как это был первый опыт создания приложения, нужно было начать с пустого шаблона «EmptyActivity».

После выбора данного шаблона для главного экрана, AndroidStudio предлагает присвоить Activity какое-либо имя (рисунок 21).

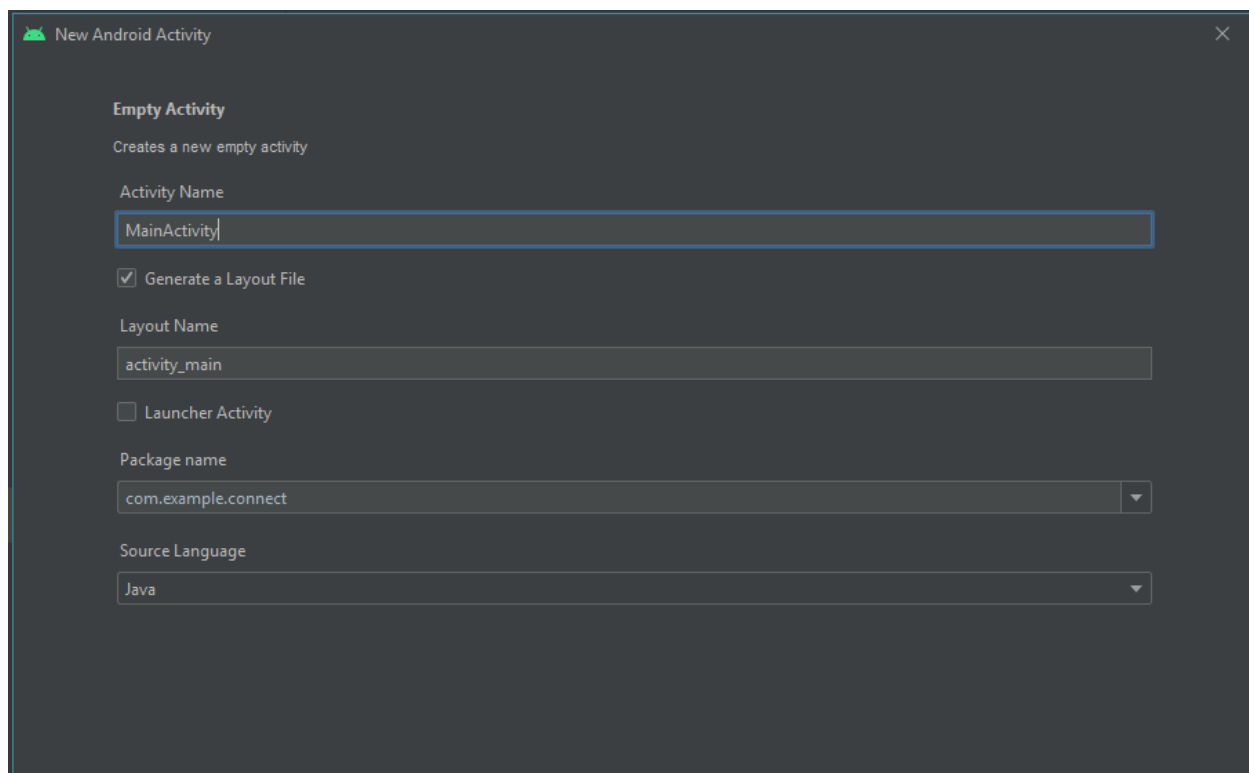


Рисунок 21 – Изменение имени Activity

По стандарту главный экран имеет название «MainActivity», было принято решение оставить все без изменений.

После создания MainActivity, система создает два файла: MainActivity.java и activity_main.xml.

Для разработки мобильного интерфейса в Android используется расширяемый язык разметки (Extensible Markup Language), сокращенно XML.

Язык разметки – искусственный язык, который использует набор аннотаций к тексту, предоставляющий инструкции относительно структуры текста или его отражение.

Стандарт XML определяет набор базовых лексических и синтаксических правил для построения языка описания информации путем применения простых тегов. Иными словами, предложенный стандарт определяет метаязык, на основе которого путем введения ограничений на структуру и содержание документов определяются специфические, предметно-ориентированные языки разметки данных.

При создании любого нового Activity создается два файла с расширением .java и .xml.

Xml файл отвечает за визуальную часть окна, а java за исполнение необходимых функций и циклов.

Работа с xml файлом проводится тремя возможными вариантами:

- в режиме «Code»;
- в режиме «Design»;
- в режиме «Split».

Режим «Code» подразумевает работу только с кодом (рисунок 22)



Рисунок 22 – Режим работы «Code»

Режим «Design» позволяет пользователю перетаскивать все возможные элементы на экран, при этом они автоматически прописываются в коде, этот режим, своего рода, конструктор. Режим «Design» показан на рисунке 23.

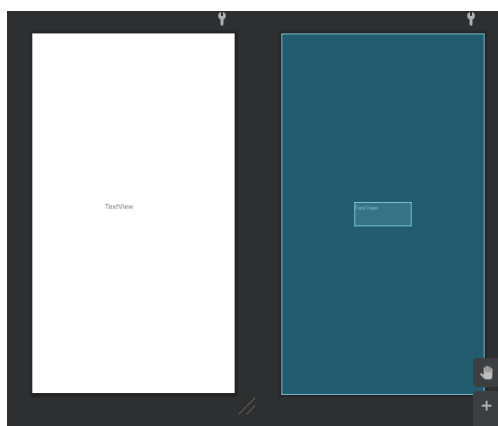


Рисунок 23 – Режим работы «Design»

Для работы с графическим конструктором использовались графические инструменты (рисунок 24).

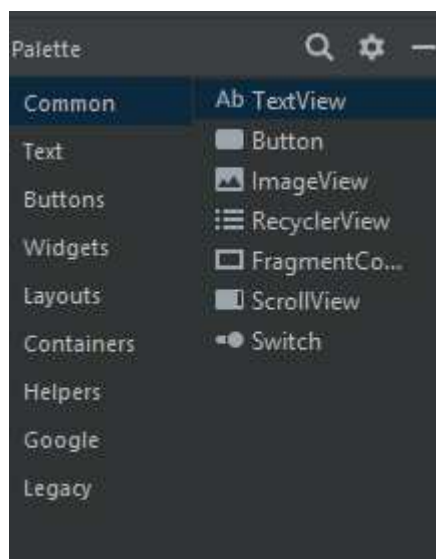


Рисунок 24 – Графические инструменты

Для создания главного экрана в приложении были использованы компоненты TextView и Button.

TextView предназначен для отображения текста без возможности редактирования его пользователем.

TextView – один из самых используемых компонентов. С его помощью пользователю удобнее ориентироваться в программе. Он служит для представления пользователю описательного текста, названий окон и тд. Для отображения текста в TextView в файле разметки используется атрибут `android:text=""`. Для редактирования компонента «TextView» используются следующие атрибуты:

- `android:id="@+id/textView5";`
- `android:layout_width="wrap_content";`
- `android:layout_height="wrap_content";`
- `android:gravity="center";`
- `android:text="___";`
- `android:textColor="@android:color/secondary_text_light_nodisable";`

– android:textSize="20sp".

Для любого графического элемента в приложении важны параметры `layoutwidth` и `layoutheight`, это параметры ширины и высоты элемента. Чаще всего в этих параметрах используются значения «`match_parent`» или «`wrap_content`», первый позволяет элементу занимать все доступное для него место, а второй адаптирует размер элемента в зависимости от его настроек, например, размера шрифта и т.д.

В этот `TextView` вписано название приложения «Connect».

Также стоит отметить атрибут `android:id="@+id/"`. Он предназначен для присвоения любому элементу уникального адреса, по которому можно целенаправленно изменять параметры конкретных элементов.

Помимо компонента `TextView` были добавлены 5 кнопок `Button` (рисунок 25):

- Поиск.
- Для групп.
- Информация по предметам.
- Новая информация.
- Настройки.

Каждая кнопка предназначена для перехода в другие `Activity`, в которых размещены необходимые для выполнения поставленных задач элементы экрана.

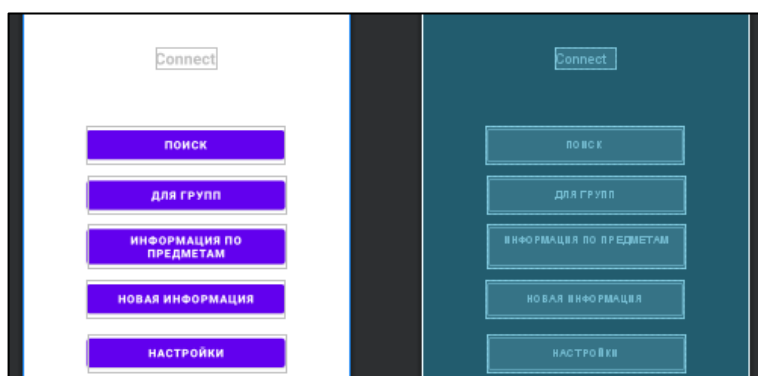


Рисунок 25 – Главный экран приложения «Connect»

Каждой кнопке присваивается атрибут `android:id="@+id/"`, благодаря которому появляется возможность работать с кнопкой в Java – файле.

Перед тем как прописывать кнопкам команды перехода на другие окна, необходимо эти окна создать. Все окна создаются по принципу создания окна `MainActivity`, создается `EmptyActivity`, в XML – файле которого прописывается интерфейс, а в Java – файле программный код, активирующий действия элементов окна.

На первоначальном этапе было создано 6 окон, но при работе с функцией вывода информации из базы данных в приложение появилась потребность в дополнительном окне.

По итогу в приложении было создано 7 окон, считая главное:

- `MainActivity`.
- `Activity search`.
- `Activity new ads`.
- `Activity gruppa`.
- `Activity predmety`.
- `Activity settings`.
- `ShowActivity`.

Прежде чем искать какую-либо информацию, необходимо заполнить заранее созданную базу данных.

Для этого были созданы окна `Activitynewads`, `Activity gruppa`, `Activity predmety`. Они абсолютно идентичны, единственное их отличие — это названия переменных, в которые записывается информация. Изображение экрана `Activitynewads` изображено на рисунке 26. Окна `Activitygruppa` и `Activitypredmety` визуально отличаются только заголовками. Код разделов категории «Новая запись» представлен в приложениях И – Л.

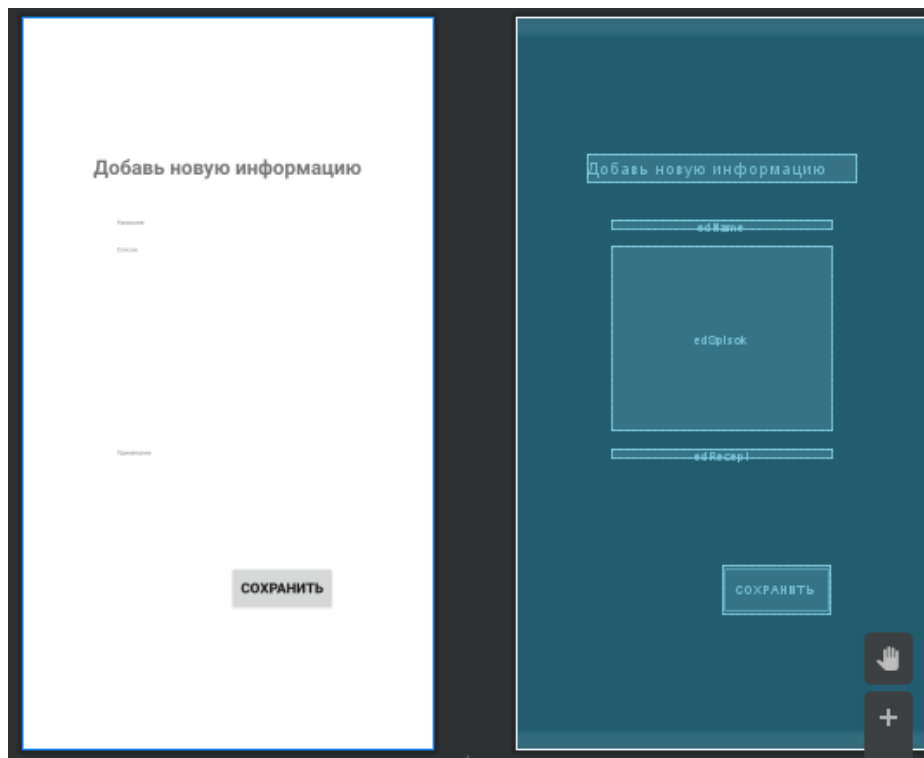


Рисунок 26– Изображение экрана Activitynewads

Чтобы внести данные, заполненные пользователем в этих окнах, в базу данных, необходимо нажать кнопку «Сохранить», после чего выполнится команда записи информации в базу. Код, направленный на действие кнопки, изображен на рисунке 27.

```
public void onClickAdd(View view) {

    String id = DataBaseRecepty.getKey();
    String name = edName.getText().toString();
    String recept = edRecept.getText().toString();
    String spisok = edSpisok.getText().toString();

    Recepty newRecept = new Recepty(id, name, spisok, recept);
    if (!TextUtils.isEmpty(name) && !TextUtils.isEmpty(spisok) && !TextUtils.isEmpty(recept)) {

        DataBaseRecepty.push().setValue(newRecept);
        Toast.makeText(context, this, text: "Запись добавлена", Toast.LENGTH_SHORT).show();
        this.finish();
    } else {
        Toast.makeText(context, this, text: "Не все поля заполнены", Toast.LENGTH_SHORT).show();
    }
}
```

Рисунок 27 – Код кнопки «Сохранить»

Следующим этапом разработки является создание функции поиска.

Activitysearchпредставляет из себя пустое окно с атрибутом ListView, это атрибут списка, в который выводится информация, которая содержится в базе данных. При помощи команды getDataFromDB(рисунок 28), данные из базы данных выгружаются в ListView.

```
private void getDataFromDB() {  
  
    ValueEventListener vListener = new ValueEventListener() {  
        @Override  
        public void onDataChange(@NotNull DataSnapshot dataSnapshot)  
  
            if (listData.size() > 0) listData.clear();  
            if (listTemp.size() > 0) listTemp.clear();  
            if (listTemp2.size() > 0) listTemp2.clear();  
            if (listTemp1.size() > 0) listTemp1.clear();  
            for (DataSnapshot ds : dataSnapshot.getChildren()) {
```

Рисунок 28 – Команда getDataFromDB

После вывода данных в виде списка, появилась информация о том, что при нажатии на любую строку, выбранный элемент можно выводить на отдельный экран. Для этого был создан экран ShowActivity и при помощи команды getIntentMain (рисунок 29), вся информация из данной строки выводится на экран.

```
private void getIntentMain()  
{  
    Intent i = getIntent();  
    if (i != null) {  
  
        tvName.setText(i.getStringExtra(Constant.RECEPT_NAME));  
        tvIngr.setText(i.getStringExtra(Constant.RECEPT_INGREDIENTY));  
        tvRecept.setText(i.getStringExtra(Constant.RECEPT_OPISANIE));
```

Рисунок 29 – Команда getIntentMain

Чтобы находить необходимую пользователю информацию, в окно Activitysearch был установлен виджет поисковой строки SearchView, код фильтрации через виджет SearchView показан на рисунке 30.

```
MenuItem menuItem = menu.findItem(R.id.search_view);
SearchView searchView = (SearchView) MenuItemCompat.getActionView(menuItem);

searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
    @Override
    public boolean onQueryTextSubmit(String query) { return false; }

    @Override
    public boolean onQueryTextChange(String newText) {
        adapter.getFilter().filter(newText);
        return false;
    }
});
return super.onCreateOptionsMenu(menu);
```

Рисунок 30 – Код виджета SearchView

После завершения этапов поиска и внесения новых данных появилась идея раздела «Настройки», ключевыми действиями в данном разделе являлись настройка темы (светлая/темная) при помощи виджета switch и изменение размера шрифта при помощи radiogroup.

Сначала была разработана визуальная часть окна, в нем были размещены переключатель (switch) и группа кнопок выбора (выбрать можно только один размер шрифта, несколько размеров одновременно выбирать в этой группе нельзя). Результат визуальной части окна изображен на рисунке 31.

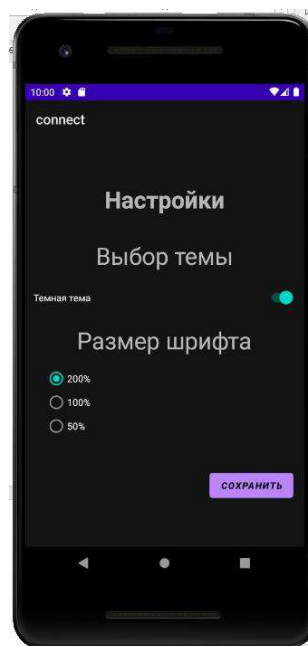


Рисунок 31 – Визуальная часть окна «Настройки»

После того как разработка шаблона была окончена, все команды прописаны, а интерфейс смоделирован, было принято решение пересмотреть визуальную часть, а вместе с этим и программную.

В силу того, что после разработки данного шаблона, появился небольшой опыт и багаж знаний, было решено разработать новый шаблон, но уже с корректировками, при этом проверить его практическую значимость.

2.6 Разработка новой версии шаблона приложения

Идеей для приложения стал вопрос о коммуникации преподавателей со студентами в сети. В 2021 году весьма актуальной темой стал дистанционный метод коммуникаций. Для частичного решения этой проблемы было создано приложение «Teacher`snotes». Целью этого приложения является оповещение студентов преподавателями о различных изменениях в расписании, списке предметов и т.д.

Функции, реализуемые приложением:

1. Добавление информации от лица преподавателя по категориям (общие объявления, записи для группы, информация о предметах, экзаменах и зачетах);
2. Поиск информации по вышеперечисленным категориям, как от лица преподавателя, так и от лица студентов;
3. Предоставление прямого доступа на сайт Хакасского технического института – филиала СФУ.

2.6.1 Моделирование шаблона приложения

Перед разработкой шаблона необходимо определить основные функции приложения.

На диаграмме IDEF0 необходимо указать основные функции шаблона, входящую, выходящую информацию, а также механизмы выполнения функций

и элементы управления системой. Диаграмма IDEF0 шаблона приложения показана на рисунках 32–33.

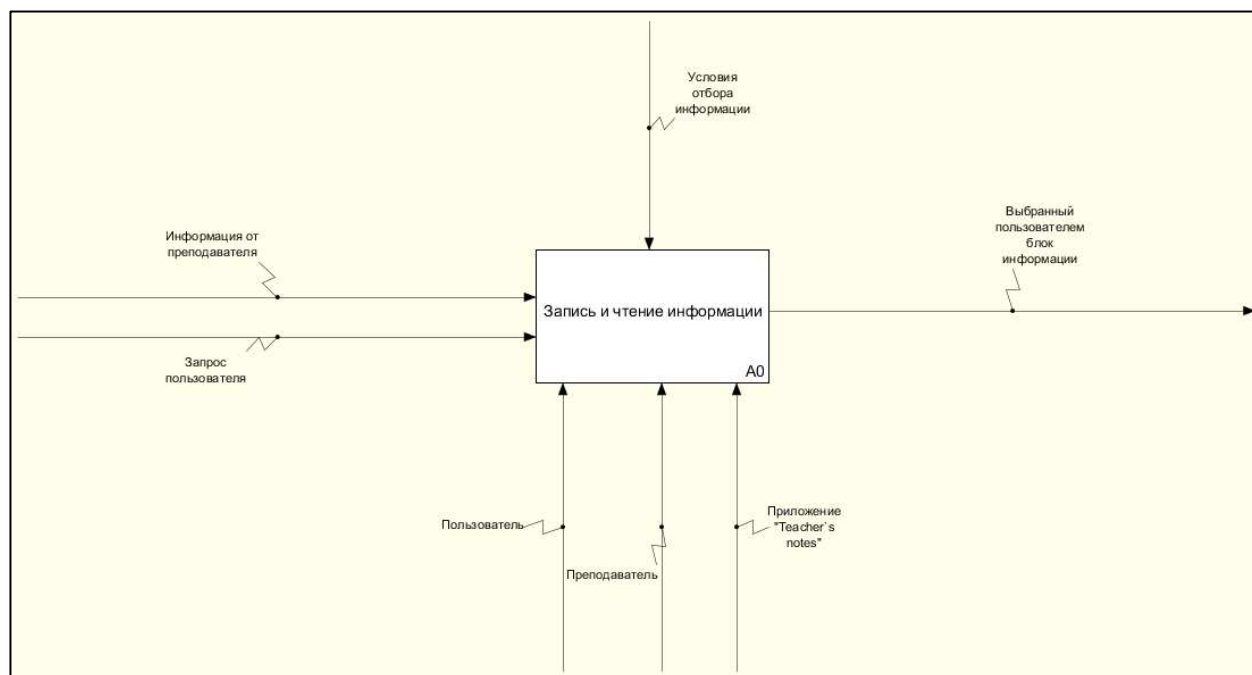


Рисунок 32 – Диаграмма IDEF0 шаблона приложения

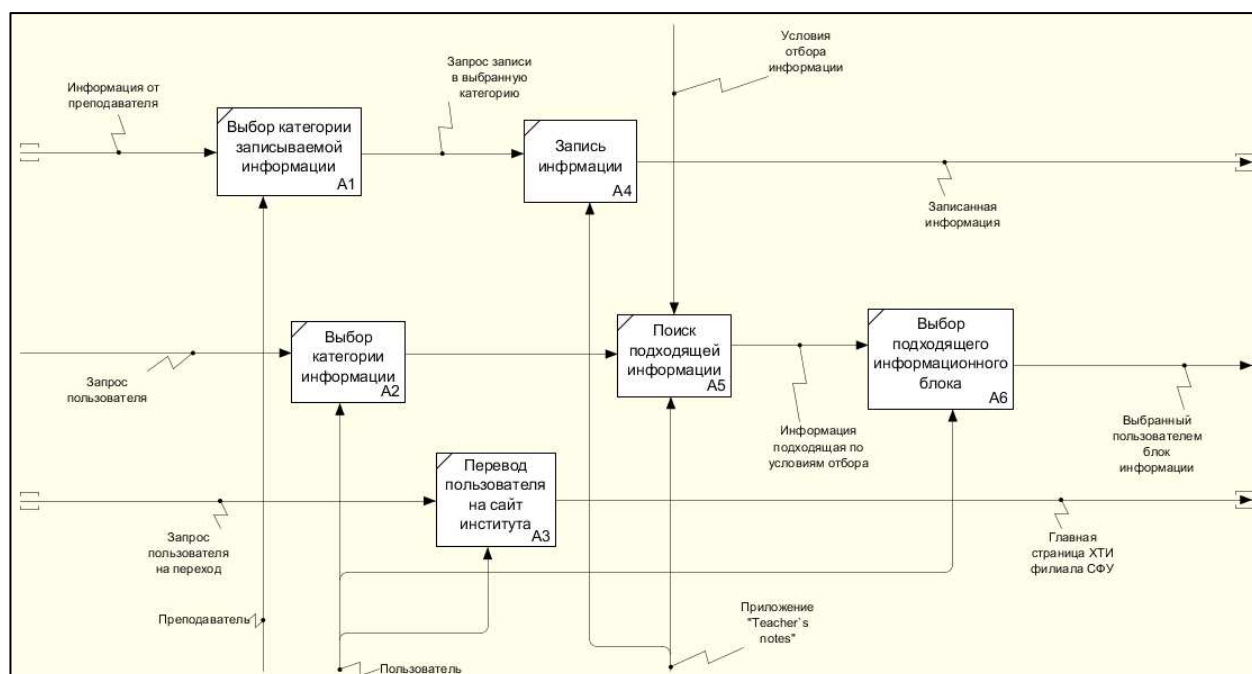


Рисунок 33 – Декомпозиция IDEF0 диаграммы шаблона приложения

Основными функциями данного шаблона являются: фильтрация информации в базе данных (фильтрация будет осуществлена при помощи совпадений значений запроса и информации в БД), запись информации в БД и чтение данных, которые ранее были внесены в базу.

После определения необходимых для шаблона функций, составляется диаграмма IDEF3.

Нотация IDEF3 — способ описания процессов с использованием структурированного метода, позволяющего эксперту в предметной области представить положение вещей как упорядоченную последовательность событий с одновременным описанием объектов, имеющих непосредственное отношение к процессу [17].

В отличие от большинства технологий моделирования бизнес-процессов, IDEF3 не имеет жестких синтаксических или семантических ограничений, делающих неудобным описание неполных или нецелостных систем. Кроме того, автор модели (системный аналитик) избавлен от необходимости смешивать свои собственные предположения о функционировании системы с экспертными утверждениями в целях заполнения пробелов в описании предметной области [18].

IDEF3 диаграмма шаблона приложения изображена на рисунке 34.

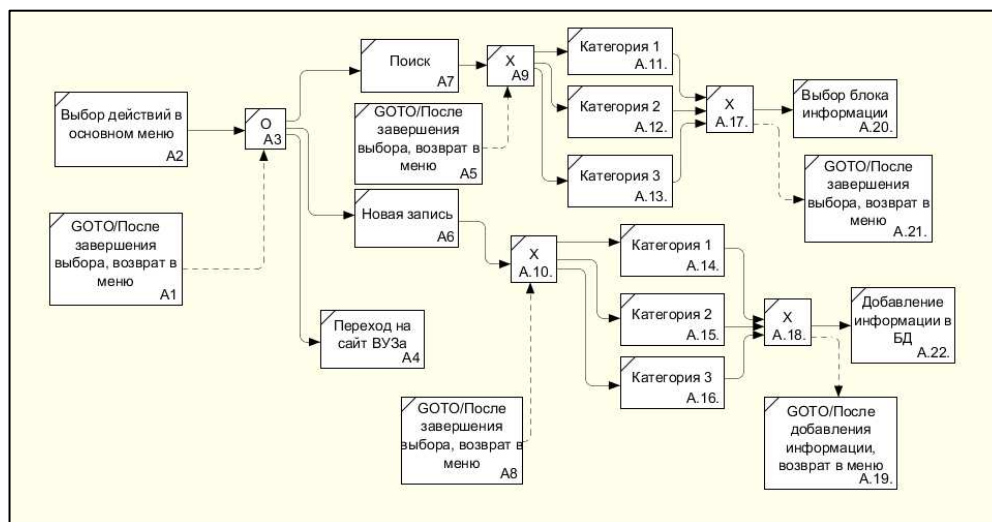


Рисунок 34 – IDEF3 диаграмма шаблона приложения

После определения функционала и потока данных разрабатываемого шаблона, можно приступить к разработке.

Разработка приложения началась с самого начала, первые изменения были внесены в главный экран MainActivity.

Вместо центральных кнопок главного меню было внедрено боковое меню NavigationDrawerActivity.

Особенностью это меню является то, что его можно вызвать из любого раздела, в каком бы вы не находились. Чтобы внедрить данное меню, был создан новый проект с названием «Teacher`snotes». После чего из предложенных системой шаблонов главного экрана был выбран NavigationDrawerActivity (рисунок 35).

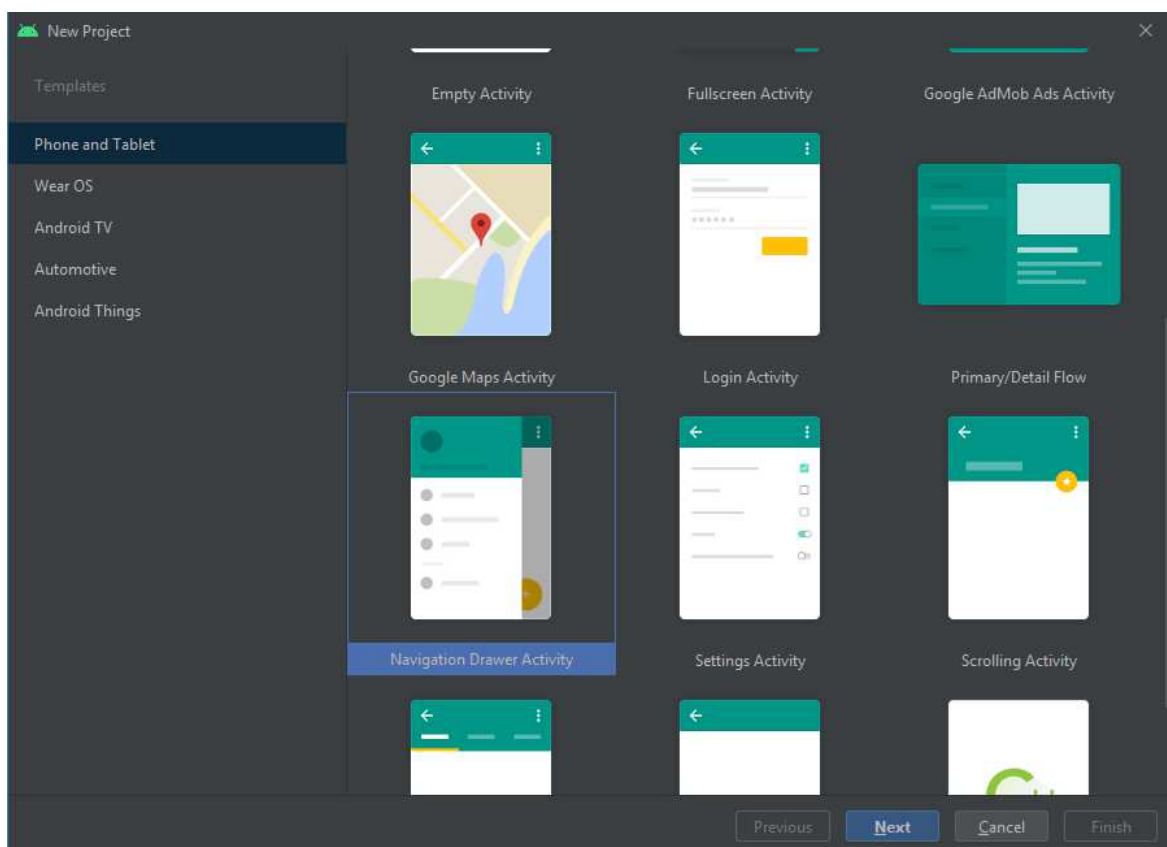


Рисунок 35 – Список шаблонов главного экрана

Если открыть файл `activity_main.xml` в режиме «Design», то можно увидеть, как будет выглядеть приложение с открытой шторкой.

За выдвигающую шторку отвечает элемент `NavigationView`, который входит последним в контейнере `DrawerLayout` и представляет собой навигационное меню. А перед меню находится вставка `include`, указывающая на разметку `app_bar_main.xml`.

Тег `NavigationView` содержит ссылку на собственную разметку в атрибуте `app:headerLayout`, который указывает на файл `nav_header_main.xml` (верхняя часть шторки), а также на меню в атрибуте `app:menu`, который ссылается на ресурс меню `menu/activity_main_drawer.xml`.

После создания главного экрана, были созданы разделы меню. Код созданных разделов указан на рисунке 36.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:tools="http://schemas.android.com/tools"
      tools:showIn="navigation_view">

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/nav_home"
            android:icon="@android:drawable/ic_menu_search"
            android:title="fragment1" />
        <item
            android:id="@+id/nav_gallery"
            android:icon="@android:drawable/ic_menu_edit"
            android:title="fragment2" />
        <item
            android:id="@+id/nav_slideshow"
            android:icon="@android:drawable/ic_menu_today"
            android:title="link fragment" />
    </group>
</menu>
```

Рисунок 36 – Код разделов меню

Следующим действием была разработка верхней части меню. Для этого в XML – файле `nav_header_main` была установлена иконка при помощи `ImageView`. Далее при помощи `TextView` были написаны заголовок и подзаголовок меню.

Результат разработки верхней части меню изображен на рисунке 37.



Рисунок 37 – Результат разработки верхней части меню

С помощью бокового меню пользователь имеет возможность переключаться между фрагментами приложения.

Фрагмент – это отдельный экран, в который помещаются различные графические объекты, как и в обычных окнах, но тут стоит обратить внимание, что функции, выполнение которых возможно в обычном `Activity`, частично не могут выполняться во фрагментах.

Первый фрагмент является фрагментом поиска и вывода информации, так как фрагмент имеет возможность лишь выводить информацию простым списком, было принято решение о структурированном отображении информации. Для структурированности было создано три кнопки поиска информации (рисунок 38).

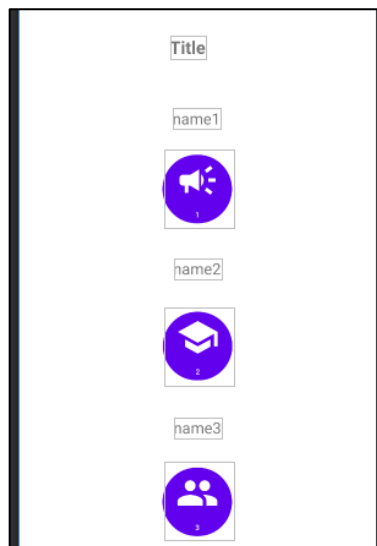


Рисунок 38 – Содержание первого фрагмента

При нажатии на одну из кнопок, вызывается процедура, открывающая простое Activity, в котором присутствует возможность структурировать записи. Код процедуры вызова Activity через кнопку изображен на рисунке 39.

```
btn1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent int1 = new Intent(getActivity(),searchadv.class);
        startActivity(int1);
    }
});

btn2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent int2 = new Intent(getActivity(),searchpredmet.class);
        startActivity(int2);
    }
});

btn3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent int3 = new Intent(getActivity(),searchgroup.class);
        startActivity(int3);
    }
});
```

Рисунок 39 – Код процедуры вызова Activity через кнопку

После нажатия на одну из кнопок, приложение открывает Activity, в котором создан обработчик RecyclerView.

RecyclerView — это группа представлений для отображения коллекций. Она обеспечивает более гибкую замену для старых групп представлений, таких как ListView и GridView.

Код созданного RecyclerView в Activity_searchadv изображен на рисунке 40.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".searchadv">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvAdv"
        android:layout_width="409dp"
        android:layout_height="729dp"
        android:layout_marginStart="1dp"
        android:layout_marginLeft="1dp"
        android:layout_marginTop="1dp"
        android:layout_marginEnd="1dp"
        android:layout_marginRight="1dp"
        android:layout_marginBottom="1dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Рисунок 40 – Созданный RecyclerView

Чтобы получить данные через RecyclerView, были созданы готовые шаблоны вывода информации, они называются item (рисунок 41).

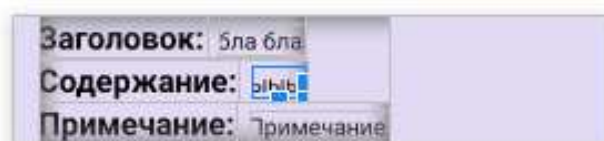


Рисунок 41 – Готовые шаблоны для вывода информации

Так как информация в базе данных разделена на три категории, то и шаблонов вывода информации было создано три.

Следующий раздел в меню – раздел «Новая запись». Данный раздел существует только в приложении преподавателя, в приложении студента он скрыт.

Раздел «Новая запись» открывает фрагмент с тремя кнопками, благодаря которым исполнено разделение информации по трем категориям: «Новое объявление», «Новый предмет», «Информация для группы» (рисунок 42).

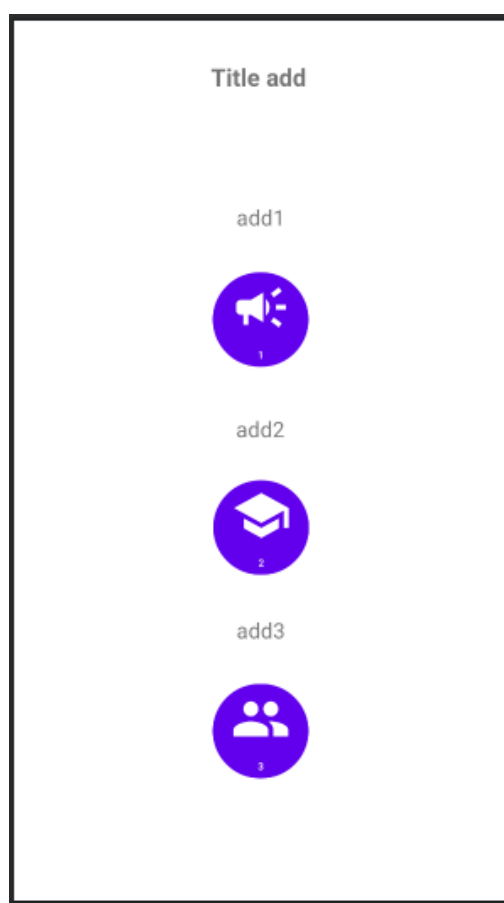


Рисунок 42 – Содержание второго фрагмента

В данном разделе кнопки действуют на открытие необходимого Activity, которые были созданы в шаблоне.

Код вызова Activity указан на рисунке 43.

```

b1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent in1 = new Intent(getActivity(), activity_new_ads.class);
        startActivity(in1);
    }
});

```

Рисунок 43 – Код вызова Activity

После заполнения базы данных через приложение, пользователь может просмотреть всю информацию, которая находится в базе данных, информация разделена на три категории, как ее добавление, так и чтение.

Результат чтения и вывода данных показан на рисунке 44.

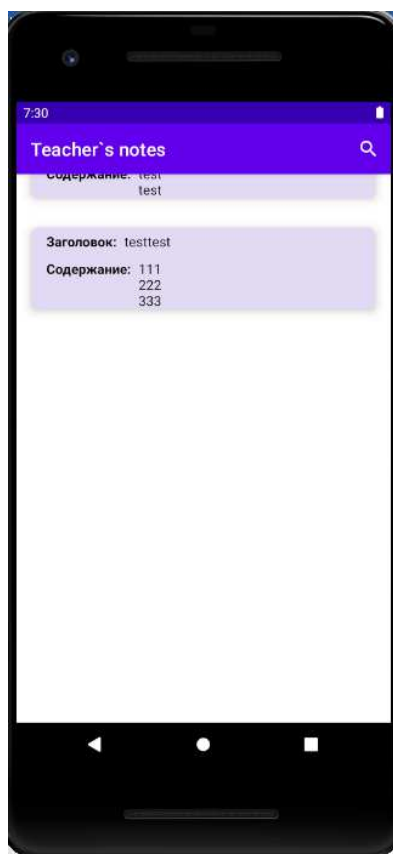


Рисунок 44 – Результат чтения и вывода данных

Третий раздел меню «Linkfragment» состоит из одного элемента – WebView.

WebView — это компонент платформы Android, который позволяет встраивать web-страницы в Android-приложения. По своей сути это внутренний

браузер, через WebView был прописан путь по прямой ссылке на сайт ХТИ, это было сделано для того, чтобы пользователь имел возможность оперативно зайти на сайт института, т.к. информация, находящаяся в базе данных приложения, в большинстве случаев будет «локальной», т.е. для небольшой аудитории, например, для группы студентов или подгруппы. Демонстрация работы WebView показана на рисунке 45.

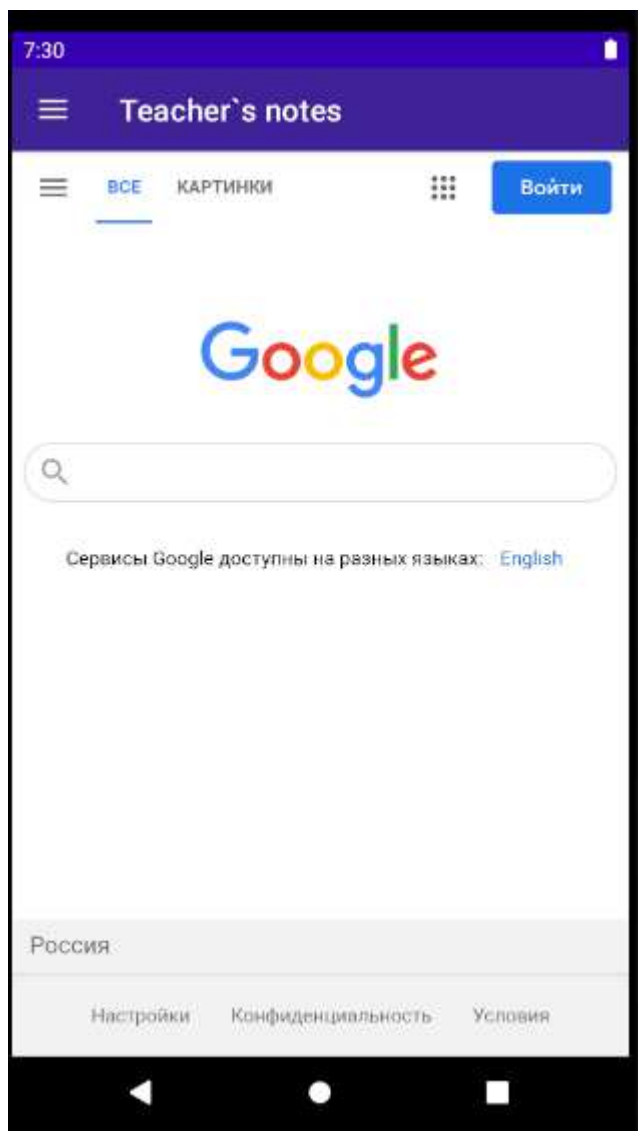


Рисунок 45 – Демонстрация работы WebView

Если сравнивать шаблон, созданный выше и приложение «Teacher'snotes», то можно заметить, что в приложении отсутствует раздел настроек.

Этот раздел был удален, т.к. параметры изменения размера шрифта были нецелесообразными, пользователь имеет возможность самостоятельно менять размер шрифта на устройстве, поэтому данная функция была бы лишней.

Что касается изменения светлой/темной темы, исходя из тех же соображений, была прописана процедура следования темы за темой устройства, т.е. как только система Android меняет тему, например, на темную, приложение повторяет это действие за устройством. Код автоматического переключения темы указан на рисунке 46.

```
super.onCreate(savedInstanceState);  
setContentView(R.layout.activity_main);  
  
//Темная тема настраивается вместе с системой  
getDelegate().setLocalNightMode(AppCompatDelegate.MODE_NIGHT_FOLLOW_SYSTEM);
```

Рисунок 46– Код автоматического переключения темы

Работа переключения темной и светлой темы изображены на рисунке 47.

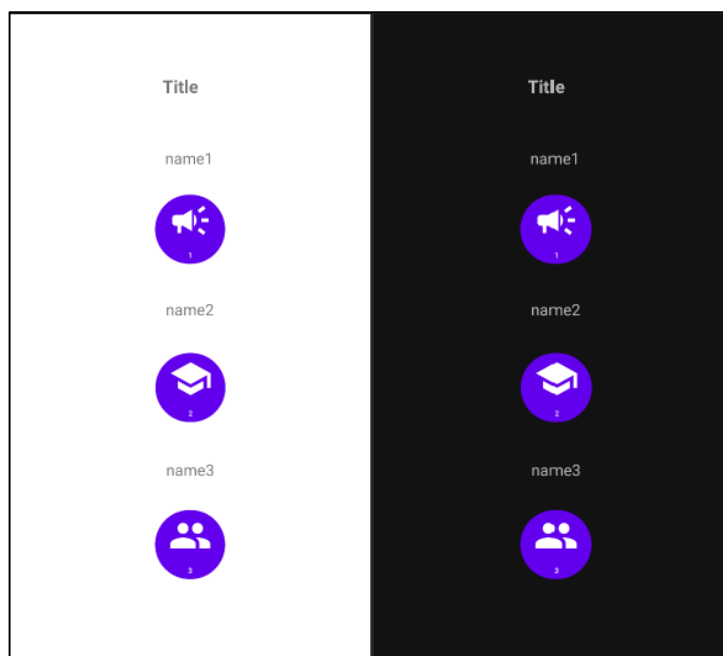


Рисунок 47 – Работа переключения темной и светлой темы

2.6.2 Проверка практической значимости разработанного шаблона

После того, как шаблон был усовершенствован, начался этап проверки практической значимости разработанного шаблона.

В первую очередь было изменено название приложения, а также названия пунктов главного меню (рисунок 48). Код главного меню приложения представлен в приложении А.

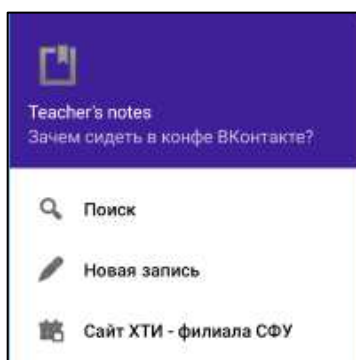


Рисунок 48 – Главное меню приложения

После этого были изменены разделы «Поиск» и «Новая запись». Измененные разделы показаны на рисунках 49-50. Код данных разделов представлен в приложениях Б – В.

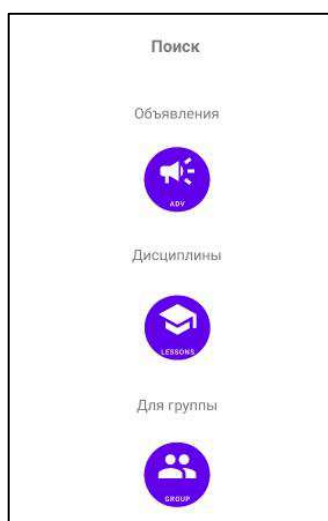


Рисунок 49 – Раздел «Поиск»

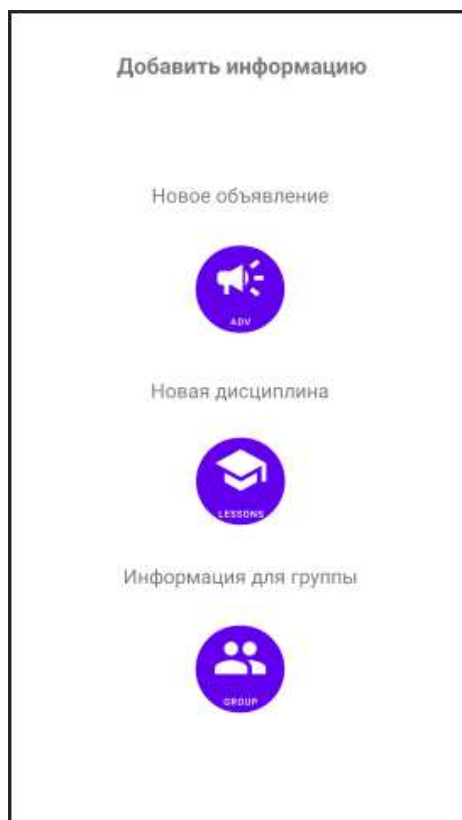


Рисунок 50 – Раздел «Новая запись»

Далее были изменены названия полей и кнопок в каждой категории, после чего в раздел меню «Сайт ХТИ – филиала СФУ» была введена прямая ссылка на сайт института. Код данного раздела указан на рисунке 51. Код третьего раздела меню приложения представлен в приложении Г.

```
public class Fragment3 extends Fragment {
    @Nullable
    @org.jetbrains.annotations.Nullable
    @Override
    public View onCreateView(@NonNull @NotNull LayoutInflater inflater, @Nullable @org.jetbrains.annotations.Nullable
        View view = inflater.inflate(R.layout.fragment3_layout, container, attachToRoot: false);
        WebView webView = (WebView)view.findViewById(R.id.raspisanieWeb);
        webView.getSettings().setJavaScriptEnabled(true);
        webView.setWebViewClient(new WebViewClient());
        webView.loadUrl("http://khti.sfu-kras.ru/");
        return view;
    }
}
```

Рисунок 51 – Код раздела «Сайт ХТИ – филиала СФУ»

Следующим этапом было необходимо заполнить базу данных информацией, необходимой для студентов (рисунок 52).

После чего было принято решение создать функцию, при помощи которой пользователь, выбирая любой блок информации, смог бы увидеть всю информацию этого блока в отдельном окне (рисунок 53). Код разделов категории «Поиск» указан в приложениях Д – Ж.

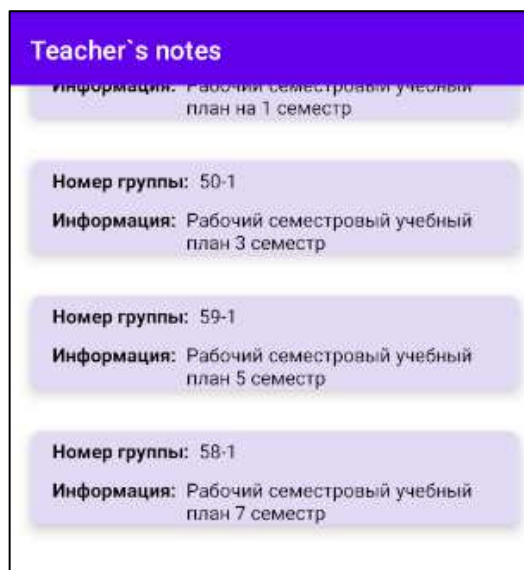


Рисунок 52 – Заполненная база данных

Код вывода списков информации разделов указан в приложениях М – П.

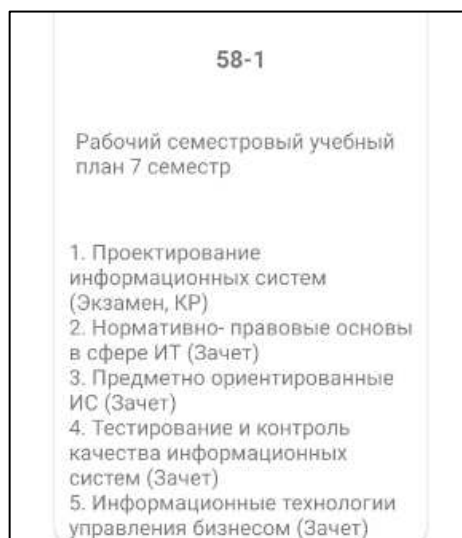


Рисунок 53 – Окно выбранного блока информации

2.7 Выводы по проектному разделу

В данном разделе выявлены основные требования к мобильному приложению в соответствии с техническим заданием.

Построены диаграммы IDEF3 и IDEF0 которые показывают, как работает система.

Описан процесс установки системы «AndroidEmulator», с помощью которого проверяется работоспособность программного продукта.

Разработан интерфейс первой версии шаблона для мобильного приложения. После того, как был накоплен необходимый опыт, была создана вторая версия шаблона приложения, отличающаяся как функционалом, так и визуальным оформлением.

Далее была проверена практическая значимость разработанного шаблона.

Во время разработки шаблона было большое количество трудностей, с которыми сталкивается большая часть начинающих разработчиков, например, одной из основных проблем была проблема недостатка информации на русском языке, приходилось читать много литературы и смотреть много видеоматериалов на английском языке, при этом, видеоматериалы часто озвучивались не носителями английского, из-за сильного акцента было крайне тяжело воспринимать информацию на слух.

Помимо этого, были трудности с внедрением боковой шторки меню в проект, было предпринято множество попыток внедрения, но по итогу все привело к тому, что пришлось создавать новый проект, внедрив боковое меню на первом этапе. Также, во время разработки шаблона, возникла сложность переноса созданных ранее функций в новую версию шаблона, так как в новой версии три основных экрана являются фрагментами, для того чтобы при переходе на какой-либо фрагмент открывалось необходимое окно, нужно было создать кнопки во фрагментах, после чего прописать через нажатие вызов необходимого окна.

3 Экономический раздел разработки шаблона для приложения

При разработке программного продукта очень важной является проблема оценки материальных затрат на успешное завершение проекта. Существует множество методов для выполнения такой оценки, среди которых можно выделить традиционную оценку затрат.

3.1 Капитальные затраты

Капитальные затраты – это затраты на информационную систему, носящие разовый характер, приносят прибыль. Данные затраты не утрачиваются, а воспроизводятся.

Затраты на разработку информационной системы вычисляются по формуле

$$K=K_{\text{пр}}+ K_{\text{тс}}+ K_{\text{лс}}+ K_{\text{ло}}+ K_{\text{ио}}+ K_{\text{об}}+ K_{\text{оэ}}, \quad (1)$$

где $K_{\text{пр}}$ – затраты на проектирование ИС;

$K_{\text{тс}}$ – затраты на технические средства управления;

$K_{\text{лс}}$ – затраты на создание линий связи локальных сетей;

$K_{\text{по}}$ – затраты на программные средства;

$K_{\text{ио}}$ – затраты на формирование информационной базы;

$K_{\text{об}}$ – затраты на обучение персонала;

$K_{\text{оэ}}$ – затраты на опытную эксплуатацию

Затраты на проектирование ИС рассчитываются по формуле

$$K_{\text{пр}}=K_{\text{зп}}+K_{\text{кипс}}+ K_{\text{свт}}+ K_{\text{проч}}, \quad (2)$$

где $K_{\text{зп}}$ – затраты на заработную плату специалистов;

$K_{\text{кипс}}$ – затраты на инструментальные программные средства проектирования;

$K_{\text{свт}}$ – затраты на средства вычислительной техники для проектирования;

$K_{\text{проч}}$ – прочие затраты на проектирование.

Для оценки экономической эффективности первоначально расцениваются затраты на создание программного продукта и на его внедрение.

Для создания шаблона приложения потребуется один программист и время его работы будет составлять один месяц, что достаточно для создания программного продукта. Заработная плата программиста соответствует минимальному размеру оплаты труда в Республике Хакасия. Кроме того, учтены северная надбавка, районный коэффициент и обязательные отчисления во внебюджетные фонды.

$$K_{\text{зп}} = 12\,792 * 1,6 * 1,302 = 26\,648 \text{ рублей.}$$

Оборудование, необходимое программисту для создания шаблона указано на таблице 5.

Таблица 5 – ПК программиста

Состав ПК:	Модель	Кол-во	Стоимость, руб.	Срок службы, лет
1.Видеокарта	MSI GeForce GTX 1050 Ti 4GT	1	17699	5
2.Процессор	AMD Ryzen 3 3100 OEM	1	9999	5
3.Оперативная память	Goodram Iridium [IR-X2666D464L16S/8G] 8 ГБ	2	6500	5
4.Жесткий диск	1 ТБ Жесткий диск WD Blue	1	3199	5
5.Блок питания	Deercool DN550	1	3799	5
6.Монитор	LG 24MK430H	1	8199	5

Продолжение таблицы 5

7.Клавиатура	A4Tech Bloody Q100	1	1299	5
8.Мышь	A4Tech N-770FX	1	899р	5
9.Материнская плата	ASRock A320M-DVS R4.0	1	3799	5
Сумма			55392	

Ставка амортизации ПК программиста за 5 лет: $55392/5 = 11078,4$ рубля

Ставка амортизации ПК за рабочий период: $11078,4/12 = 923,2$ рубля

$K_{\text{свт}} = 923,2$ рубля

Из программного обеспечения платным является только операционная система «Microsoft Windows 10 Домашняя» стоимостью 9199 рублей, активация является бессрочной, что позволяет ей пользоваться бесконечно, остальное ПО является бесплатным. ПО программиста указано в таблице 6.

Таблица 6 – ПО программиста

Название ПО	Стоимость	Срок подписки
Windows 10home	9199р	Не ограничено
Firebase database	Бесплатно	Не ограничено
Android studio	Бесплатно	Не ограничено

Из программного обеспечения платным является только операционная система «Microsoft Windows 10 home» стоимостью 9199 рублей, активация является бессрочной, что позволяет ей пользоваться бесконечно, остальное ПО является бесплатным. Амортизацию Windows будем считать по сроку службы ПК, то есть 5 лет.

Ставка амортизации программного обеспечения за год: $9199/5 = 1839,8$ рублей

Ставка амортизации ПО за месяц работы: $1839,8/12 = 153,3$ рубля

$K_{\text{кппс}} = 153,3$ рубля.

К прочим затратам можно отнести оплату интернета, электроэнергии и т.д.

Для расчета прочих затрат возьмем 3% от общей суммы всех затрат.

$K_{\text{проч}} = (26\,648 + 153 + 923) * 0,03 = 831,7\text{р.}$

Таблица 7 – Затраты на разработку

Затраты	Сумма затрат, руб.
Заработная плата программиста	26 648
Затраты на ПО	153
Затраты на средства вычислительной техники	923
Затраты на прочие расходы	831
Итого:	28 555

$K_{\text{пр}} = 28555$ рублей.

Шаблон размещается на маркетплейсах абсолютно бесплатно, следовательно, дополнительные затраты на внедрение и эксплуатацию отсутствуют.

Шаблон считается законченным после того, как программист завершит свою работу, помимо создания самого шаблона, программист, для наглядности работы алгоритмов шаблона, наполняет продукт небольшим количеством

информации, из этого следует, что нет необходимости пользоваться чьими-либо услугами, помимо услуг программиста.

$K_{\text{оз}} = 0$ рублей.

Остальные капитальные затраты отсутствуют, так как проект не требует каких-либо дополнительных программных средств, формирования информационной базы и обучения персонала и т.д.

$K = 28555$ рублей.

3.2 Эксплуатационные затраты

$$C = C_{\text{зп}} + C_{\text{ао}} + C_{\text{то}} + C_{\text{лс}} + C_{\text{ни}} + C_{\text{проч}}, \quad (3)$$

где $C_{\text{зп}}$ – зарплата персонала, работающего с информационной системой;

$C_{\text{ао}}$ – амортизационные отчисления;

$C_{\text{то}}$ – затрата на техническое обслуживание;

$C_{\text{лс}}$ – затраты на использование глобальных сетей;

$C_{\text{ни}}$ – затраты на носители информации;

$C_{\text{проч}}$ – прочие затраты.

После размещения шаблона на площадке, продукт становится автономным, никаких затрат на эксплуатацию не производится, размещение продукта на площадках также бесплатно, следовательно, $C = 0$.

3.3 Расчет реализации проекта

Для расчета затрат на реализацию проекта используется следующая формула TCO:

$$TCO = DE + IC_1 + IC_2, \quad (4)$$

где DE (direct expenses) – прямые расходы;

IC 1 (indirectcosts) – косвенные расходы первой группы;

IC 2 (indirectcosts) – косвенные расходы второй группы.

$$DE = DE\ 1 + DE\ 2 + DE\ 3 + DE\ 4 + DE\ 5 + DE\ 6 + DE\ 7 + DE\ 8, \quad (5)$$

где DE 1 – капитальные затраты;

DE 2 – расходы на управление информационными технологиями;

DE 3 – расходы на техническую поддержку автоматизированного обеспечения и программного обеспечения;

DE 4 – расходы на разработку прикладного программного обеспечения внутренними силами;

DE 5 – расходы на аутсорсинг;

DE 6 – командировочные расходы;

DE 7 – расходы на услуги связи;

DE 8 – другие группы расходов.

DE 1 = 28555 руб.;

DE 2 = 0 руб., т.к. никаких затрат на эксплуатацию не производится;

DE 3 = 0 руб., т.к. после размещения программы на площадку, тех. поддержка не нужна;

DE 4 = 0 руб., прикладное ПО не требуется;

DE 5 = 0 руб., при разработке не требовалась помощь сторонних специалистов;

DE 6 = 0 руб., отсутствие командировочных расходов;

DE 7 = 0 руб., интернет уже включен в капитальные затраты, проект не требует отдельных затрат на услуги связи;

DE 8 = 0 руб., прочие затраты отсутствуют;

DE = 28555 рублей.

TCO = 28555 + 0 + 0 = 28555 рублей.

3.4 Определение доходов и экономической эффективности реализации проекта

Стоимость готового шаблона мобильного приложения сильно зависит от его сложности, от разнообразия функционала, от языка, на котором он написан и т.д.

Базовые шаблоны имеют базовую стоимость 15-25 долларов, шаблоны, предусматривающие несколько видов приложений в одном, дороже, диапазон цен от 75 до 200 долларов, многое также зависит и от программиста, который самостоятельно оценивает свой продукт [19].

Цены шаблонов указаны на рисунке 54.

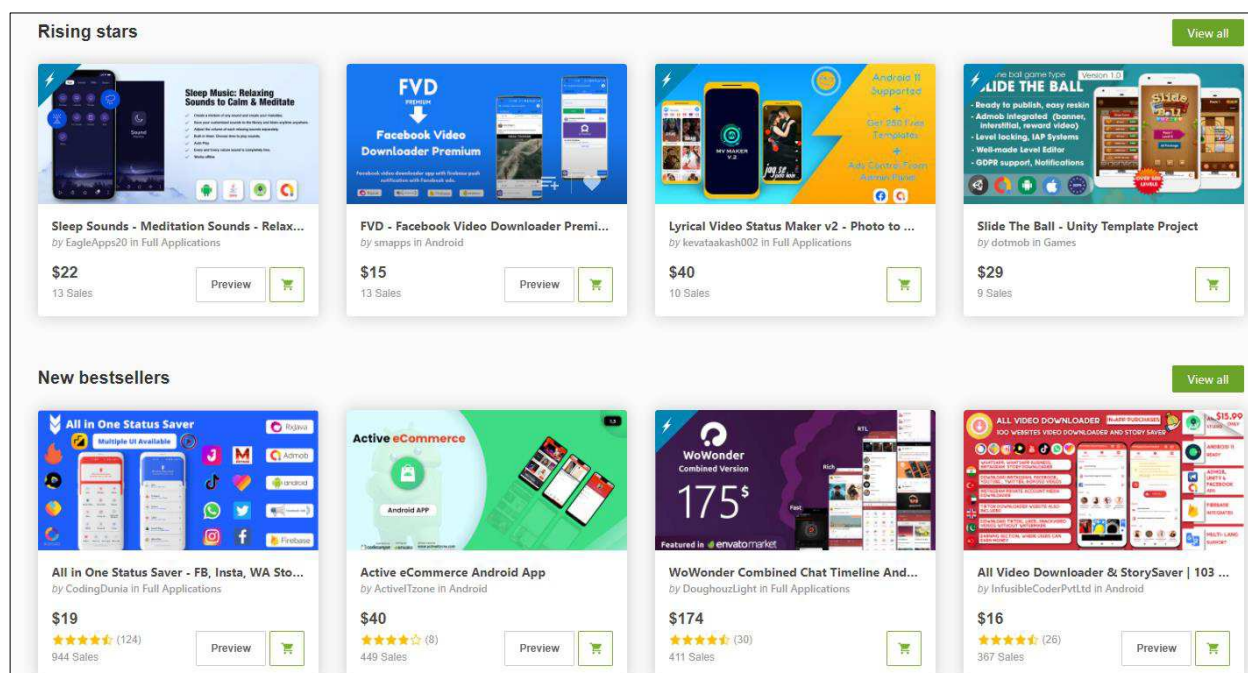


Рисунок 54 – Цены шаблонов

За шаблон, разработанный начинающим программистом, следует установить цену по низу рынка, т.к. примитивный функционал и фильтрация данных из базы являются базовыми элементами для приложения. Цена 10 долларов за созданный продукт, вполне допустима, среди конкурентов оно не

будет выделяться функционалом, но за счет низкой стоимости сможет привлекать к себе внимание.

Если учесть, что, в среднем, скачиваний у шаблонов подобной сложности около 40, примерно за 2 месяца, а за счет низкой цены это вполне может быть реализуемо, получив за 40 скачиваний в общей сложности около 400 долларов, разработка данного проекта окупится.

Исходя из вышесказанного, можно сделать вывод, что необходимо адекватно определить цену шаблона для того, чтобы количество скачиваний достигло цифры, необходимой для того, чтобы проект окупился.

3.5 Оценка рисков проекта

Для расчета рисков необходимо проанализировать создание системы поддержки принятия решений по двум основным группам рисков:

- Риски внедрения проекта.
- Эксплуатационные риски.

Риски проекта представлены в таблице 8.

Таблица 8 – Риски проекта

№	Группы рисков	Перечень рисков проекта	Уровень влияния риска на проект	Вероятность риска	Возможность предотвращения или снижения риска
2.	Риски внедрения проекта	Нехватка времени на внедрение системы	Высокий уровень	Низкая	Выбор оптимального количества времени
3.	Эксплуатационные риски	Недоработка системы	Высокий уровень	Средняя	Подробное техническое задание, грамотное планирование задач при разработке

Продолжение таблицы 8

		Ошибки в работе системы	Высокий уровень	Средняя	Тестирование системы перед внедрением
4.	Риски окупаемости	Недостаточное кол-во скачиваний	Средний уровень	Средняя	Доработка шаблона, размещение шаблона на нескольких площадках продаж.

3.6 Выводы по экономическому разделу

Исходя из всего перечисленного в данном разделе, стоит отметить, что помимо проектирования и создания приложения, крайне важно рассчитать расходы на ПО и оборудование, амортизацию, затраты из разряда оплаты сети Интернет, заработной платы персонала и т.д.

Необходимо рассчитать все риски и возможности их минимизации. Обоснованно указать стоимость на свой продукт, опираясь на данные рынка и многое другое.

Все это необходимо для того, чтобы не возникало вопросов о том, окупится ли проект, будет ли он актуален или нет.

Экономические расчеты нужно всегда брать во внимание, они важны настолько же, как и сам проект для разработчика.

ЗАКЛЮЧЕНИЕ

Была поставлена задача разработать шаблон мобильного приложения, а также проверить его практическую значимость.

Из средств разработки шаблона приложения была выбрана среда разработки AndroidStudio и язык программирования Java.

В качестве базы данных была выбрана RealtimedatabaseFirebase.

В ходе выполнения выпускной квалифицированной работы была достигнута поставленная цель и решены задачи:

- исследована предметная область;
- смоделирован шаблон мобильного приложения;
- разработан шаблон мобильного приложения;
- проверена практическая значимость разработанного шаблона;
- рассчитана экономическую эффективность проекта.

В результате было разработано несколько версий шаблона мобильного приложения, после чего была проверена практическая значимость разработанного шаблона.

Шаблон протестирован на разных устройствах системы Android, проблем с работоспособностью не выявлено.

Капитальные затраты составили 28555 рублей, эксплуатационные затраты равны нулю.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Рост рынка мобильных приложений [Электронный ресурс]. – Режим доступа: [https://www.tadviser.ru/index.php/Статья:Мобильные_приложения_\(мировой_рынок\)](https://www.tadviser.ru/index.php/Статья:Мобильные_приложения_(мировой_рынок)).
2. GeekBrains Android разработчик [Электронный ресурс]. – Режим доступа: https://gb.ru/geek_university/android?subtitles-mwv8=mobilnaya-razrabotka.
3. Разработчик мобильных приложений [Электронный ресурс]. – Режим доступа: <https://skillbox.ru/course/profession-mobdev/>.
4. Яндекс.Практикум [Электронный ресурс]. – Режим доступа: <https://praktikum.yandex.ru/java-developer/>.
5. Рейтинг мобильных приложений [Электронный ресурс]. – Режим доступа: <https://infourok.ru/proekt-po-informatike-rejting-mobilnyh-prilozhenij-4530944.html>.
6. Популярные виды мобильных приложений [Электронный ресурс]. – Режим доступа: <https://life.ru/p/1198714>.
7. Виды баз данных [Электронный ресурс]. – Режим доступа: <https://zametkinapolyah.ru/zametki-o-mysql/bazy-dannyx-vidy-i-tipy-baz-dannyx-struktura-relyacionnyx-baz-dannyx-proektirovanie-baz-dannyx-setevye-i-ierarxicheskie-bazy-dannyx.html>.
8. Облачные базы данных [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Облачные_базы_данных.
9. Виды фильтрации БД [Электронный ресурс]. – Режим доступа: https://studbooks.net/2338397/tehnika/bazy_dannyh_filtry.
10. Аналитика Android и iOS [Электронный ресурс]. – Режим доступа: <https://4pda.to/2020/04/11/370396/>.
11. Сравнительный анализ Eclipse и Android Studio [Электронный ресурс] – Режим доступа: <https://scienceforum.ru/2019/article/2018010389>.

12. Свободная энциклопедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Android_Studio.
13. Программирование на Андроид [Электронный ресурс]. – Режим доступа: <https://livetyping.com/ru/blog/na-chem-pishut-prilozhenija-pod-android>.
14. KotlinvsJava[Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/otus/blog/508060/>.
15. AndroidStudiownload[Электронный ресурс]. – Режим доступа: <https://developer.android.google.cn/studio?hl=es>.
16. LayoutsAndroidStudio[Электронный ресурс]. – Режим доступа: <https://startandroid.ru/ru/uroki/vse-uroki-spiskom/15-urok-6-vidy-layouts-kljuchevye-otlichija-i-svojstva.html>.
17. Основы IDEF3 [Электронный ресурс]. – Режим доступа: <https://www.cfin.ru/vernikov/idef/idef3.html>.
18. Нотация IDEF3 [Электронный ресурс]. – Режим доступа: <https://pcoding.ru/gost/idef3.pdf>.
19. Шаблоны мобильных приложений[Электронный ресурс]. – Режим доступа: <https://codecanyon.net/category/mobile/android>.
20. Выполнение и защита выпускной квалификационной работы по направлению 09.03.03 «Прикладная информатика» [Электронный ресурс] : метод. указания / сост. Е. Н. Скуратенко, В. И. Кокова, И. В. Янченко ; Сиб. федер. ун-т, ХТИ – филиал СФУ. – Электрон. текстовые, граф. дан. (0,71 МБ). – Абакан : ХТИ – филиал СФУ, 2017. – 1 файл. – Режим доступа: https://e.sfu-kras.ru/pluginfile.php/1368122/mod_resource/content/1/Met_1050.pdf.

ПРИЛОЖЕНИЕ А

Код главного меню приложения

```
Код activity main документ XML
<menu
xmlns:android="http://schemas.android.com/a
pk/res/android"
xmlns:tools="http://schemas.android.c
om/tools"
tools:showIn="navigation_view">
<group
android:checkableBehavior="single">
<item
android:id="@+id/nav_home"
android:icon="@android:drawable/ic_
menu_search"
android:title="fragment1" />
<item
android:id="@+id/nav_gallery"
android:icon="@android:drawable/ic_
menu_edit"
android:title="fragment2" />
<item
android:id="@+id/nav_slideshow"
android:icon="@android:drawable/ic_
menu_today"
android:title="link fragment" />
</group>
</menu>
```

Код activity main документ JAVA

```
public class MainActivity extends
AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {
private Toolbar toolbar;
private DrawerLayout drawer;
FragmentManager
fragmentManager;
FragmentManager
fragmentTransaction;
@RequiresApi(api =
Build.VERSION_CODES.JELLY_BEAN_MR1)
@Override
protected void onCreate(Bundle
savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
getDelegate().setLocalNightMode(AppComp
atDelegate.MODE_NIGHT_FOLLOW_SYS
TEM;
toolbar =
findViewById(R.id.toolbar);
setSupportActionBar(toolbar);
drawer =
findViewById(R.id.drawer_layout);
NavigationView navigationView
= findViewById(R.id.nav_view);
```

<pre> navigationView.setNavigationItemSelectedListener(new ActionBarDrawerToggle(this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close); drawer.addDrawerListener(toggle); toggle.syncState(); fragmentManager = getSupportFragmentManager(); fragmentTransaction = fragmentManager.beginTransaction(); fragmentTransaction.add(R.id.fragmentContainer, new Fragment1()); fragmentTransaction.commit(); // Passing each menu ID as a set of Ids because each // menu should be considered as top level destinations. Fragment2 fragment2 = new Fragment2(); FragmentManager manager = getSupportFragmentManager(); manager.beginTransaction().replace(R.id.fragmentContainer, fragment2).commit(); } @Override public boolean onOptionsItemSelected(@NonNull @NotNull MenuItem menuItem) { closeDrawer(); int id = menuItem.getItemId(); if (id == R.id.nav_home){ </pre>	<pre> fragmentManager = getSupportFragmentManager(); fragmentTransaction = fragmentManager.beginTransaction(); fragmentTransaction.replace(R.id.fragmentContainer, new Fragment1()); fragmentTransaction.commit(); } if (id == R.id.nav_gallery){ fragmentManager = getSupportFragmentManager(); fragmentTransaction = fragmentManager.beginTransaction(); fragmentTransaction.replace(R.id.fragmentContainer, new Fragment2()); fragmentTransaction.commit(); } if (id == R.id.nav_slideshow) { fragmentManager = getSupportFragmentManager(); fragmentTransaction = fragmentManager.beginTransaction(); fragmentTransaction.replace(R.id.fragmentContainer, new Fragment3()); fragmentTransaction.commit(); } return true; } private void closeDrawer(){ drawer.closeDrawer(GravityCompat.S TART); } } </pre>
--	--

ПРИЛОЖЕНИЕ Б

Код первого раздела меню

```
Кодfragment1ДокументXML
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">
<TextView
    android:id="@+id/textViewGroupS"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="100dp"
    android:layout_marginLeft="100dp"
    android:layout_marginEnd="100dp"
    android:layout_marginRight="100dp"
    android:layout_marginBottom="28dp"
    android:text="name3"
    android:textSize="18sp"
    app:layout_constraintBottom_toTopOf="@+id/buttonsgroup"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    ">
<TextView
    android:id="@+id/textViewPredmS"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="138dp"
    android:layout_marginLeft="138dp"
    android:layout_marginEnd="139dp"
    android:layout_marginRight="139dp"
    android:layout_marginBottom="32dp"
    android:text="name2"
    android:textSize="18sp"
    app:layout_constraintBottom_toTopOf="@+id/buttonspred"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    ">
<Button
    android:id="@+id/buttonsadv"
    android:layout_width="80dp"
    android:layout_height="90dp"
    android:layout_marginStart="166dp"
    android:layout_marginLeft="166dp"
    android:layout_marginTop="104dp"
    android:layout_marginEnd="166dp"
    android:layout_marginRight="166dp"
    android:drawableTop="@drawable/ic_adv"
    android:gravity="center"
    android:text="1"
    android:textSize="8.5dp"
    app:cornerRadius="50dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    ">
<Button
    android:id="@+id/buttonspred"
    android:layout_width="80dp"
```

```

        android:layout_height="90dp"
        android:layout_marginStart="166dp"
        android:layout_marginLeft="166dp"
        android:layout_marginEnd="166dp"
        android:layout_marginRight="166dp"
        android:layout_marginBottom="36dp"
        android:drawableTop="@drawable/ic_predm
"
        android:gravity="center"
        android:text="2"
        android:textSize="8.5dp"
        app:cornerRadius="50dp"
        app:layout_constraintBottom_toTopOf="@+i
d/textViewGroupS"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent
"
        />
<Button
        android:id="@+id/buttonsgroup"
        android:layout_width="80dp"
        android:layout_height="90dp"
        android:layout_marginStart="166dp"
        android:layout_marginLeft="166dp"
        android:layout_marginEnd="166dp"
        android:layout_marginRight="166dp"
        android:layout_marginBottom="84dp"
        android:drawableTop="@drawable/ic_group"
        android:gravity="center"
        android:text="3"
        android:textSize="8.5dp"
        app:cornerRadius="50dp"
        app:layout_constraintBottom_toBottomOf="p
arent"
        app:layout_constraintEnd_toEndOf="parent"

```

```

        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent
"
        />
<TextView
        android:id="@+id/textViewSearch"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="174dp"
        android:layout_marginLeft="174dp"
        android:layout_marginTop="68dp"
        android:layout_marginEnd="179dp"
        android:layout_marginRight="179dp"
        android:text="Title"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        />
<TextView
        android:id="@+id/textViewSearchAdv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="126dp"
        android:layout_marginLeft="126dp"
        android:layout_marginTop="56dp"
        android:layout_marginEnd="126dp"
        android:layout_marginRight="126dp"
        android:text="name1"
        android:textSize="18sp"
        app:layout_constraintTop_toBottomOf="@+i
d/textViewSearch"
        />
</androidx.constraintlayout.widget.Constraint
Layout>

```

Кодfragment1 документ JAVA

```

public class Fragment1 extends
Fragment {
    Button btn1,btn2,btn3;
    @Nullable
    @org.jetbrains.annotations.Nullable
    @Override
    public View
onCreateView(@NonNull @NotNull
LayoutInflater inflater, @Nullable
@org.jetbrains.annotations.Nullable
ViewGroup container, @Nullable
@org.jetbrains.annotations.Nullable Bundle
savedInstanceState) {
        View view =
inflater.inflate(R.layout.fragment1_layout,
container, false);
        btn1= (Button)
view.findViewById(R.id.buttonsadv);
        btn2= (Button)
view.findViewById(R.id.buttonspred);
        btn3= (Button)
view.findViewById(R.id.buttonsgroup);
        btn1.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent int1 = new
Intent(getActivity(),searchadv.class);
                startActivity(int1);
            }
        });
        btn2.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {

```

```

Intent int2 = new
Intent(getActivity(),searchpredmet.class);
                startActivity(int2);
            }
        });
        btn3.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent int3 = new
Intent(getActivity(),searchgroup.class);
                startActivity(int3);
            }
        });
        return view;
    }
}

```

ПРИЛОЖЕНИЕ В

Код второго раздела меню

```
Кодfragment2ДокументXML
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/secLayout"
    android:gravity="center"
    xmlns:android="http://schemas.android.com/a
    pk/res/android"
    <TextView
        android:id="@+id/textViewGroup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="100dp"
        android:layout_marginLeft="100dp"
        android:layout_marginEnd="100dp"
        android:layout_marginRight="100dp"
        android:layout_marginBottom="27dp"
        android:text="add3"
        android:textSize="18sp"
        app:layout_constraintBottom_toTopOf="@+i
        d/buttongroup"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        "
    <TextView
        android:id="@+id/textViewPredm"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="138dp"
        android:layout_marginLeft="138dp"
        android:layout_marginEnd="139dp"
        android:layout_marginRight="139dp"
        android:layout_marginBottom="25dp"
        android:text="add2"
        android:textSize="18sp"
        app:layout_constraintBottom_toTopOf="@+i
        d/buttonpred"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        "
    <Button
        android:id="@+id/buttonadv"
        android:layout_width="80dp"
        android:layout_height="90dp"
        android:layout_marginStart="166dp"
        android:layout_marginLeft="166dp"
        android:layout_marginTop="28dp"
        android:layout_marginEnd="166dp"
        android:layout_marginRight="166dp"
        android:drawableTop="@drawable/ic_adv"
        android:gravity="center"
        android:text="1"
        android:textSize="8.5dp"
        app:cornerRadius="50dp"
        app:layout_constraintEnd_toEndOf="paren"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        "
        app:layout_constraintTop_toBottomOf="@+i
```



```

d/TextViewAdv"
<Button
    android:id="@+id/buttonpred"
    android:layout_width="80dp"
    android:layout_height="90dp"
    android:layout_marginStart="166dp"
    android:layout_marginLeft="166dp"
    android:layout_marginEnd="166dp"
    android:layout_marginRight="166dp"
    android:layout_marginBottom="28dp"
    android:drawableTop="@drawable/ic_predm
"android:gravity="center"
    android:text="2"
    android:textSize="8.5dp"
    app:cornerRadius="50dp"
    app:layout_constraintBottom_toTopOf="@+i
d/TextViewGroup"
    app:layout_constraintEnd_toEndOf="paren"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="paret"
/>
<Button
    android:id="@+id/buttongroup"
    android:layout_width="80dp"
    android:layout_height="90dp"
    android:layout_marginStart="166dp"
    android:layout_marginLeft="166dp"
    android:layout_marginEnd="166dp"
    android:layout_marginRight="166dp"
    android:layout_marginBottom="96dp"
    android:drawableTop="@drawable/ic_grou"
    android:gravity="center"
    android:text="3"
    android:textSize="8.5dp"
    app:cornerRadius="50dp"

```

```

    app:layout_constraintBottom_toBottomOf="p
arent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent
"
/>
<TextView
    android:id="@+id/textViewAdd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="174dp"
    android:layout_marginLeft="174dp"
    android:layout_marginTop="33dp"
    android:layout_marginEnd="179dp"
    android:layout_marginRight="179dp"
    android:text="Title"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.479
"
    app:layout_constraintStart_toStartOf=
"parent"
    app:layout_constraintTop_toTopOf="pare/>
<TextView
    android:id="@+id/textViewAdv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="126dp"
    android:layout_marginLeft="126dp"
    android:layout_marginTop="91dp"
    android:layout_marginEnd="126dp"
    android:layout_marginRight="126dp"
    android:text="add1"
    android:textSize="18sp"

```

```

/>
</androidx.constraintlayout.widget.Constraint
Layout>

```

Кодfragment2 документ JAVA

```

public class Fragment2 extends
Fragment {
    Button b1,b2,b3;
    @Nullable
    @org.jetbrains.annotations.Nullable
    @Override
    public View onCreateView(@NonNull
    @NotNull LayoutInflater inflater, @Nullable
    @org.jetbrains.annotations.Nullable
    ViewGroup container, @Nullable
    @org.jetbrains.annotations.Nullable Bundle
    savedInstanceState) {
        View view =
inflater.inflate(R.layout.fragment2_layout,
container, false);
        b1=
view.findViewById(R.id.buttonadv);
        b2=
view.findViewById(R.id.buttonpred);
        b3=
view.findViewById(R.id.buttongroup);
        b1.setOnClickListener(new
View.OnClickListener() {
            @Override

```

```

        public void onClick(View v) {
            Intent in1 = new
Intent(getActivity(),activity_new_ads.class);
            startActivity(in1);
        }
    });
    b2.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent in2 = new
Intent(getActivity(),Predmety.class);
            startActivity(in2);
        }
    });
    b3.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent in3 = new
Intent(getActivity(),Gruppa.class);
            startActivity(in3);
        }
    });
    return view;
}
}

```

ПРИЛОЖЕНИЕ Г

Код третьего раздела меню

Кодfragment3 ДокументXML

```
<LinearLayout
xmlns:android="http://schemas.android.com/a
pk/res/android"
xmlns:app="http://schemas.android.com/apk/r
es-auto"
xmlns:tools="http://schemas.android.com/tool
s"
android:layout_width="match_parent"
android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".Fragment3">
<WebView
android:id="@+id/raspisanieWeb"
android:layout_width="match_parent"
android:layout_height="match_parent"></We
bView>
</LinearLayout>
```

Кодfragment3 документ JAVA

```
public class Fragment3 extends
Fragment
{
    @Nullable
```

```
@org.jetbrains.annotations.Nullable
@Override
public View onCreateView(@NonNull
@NotNull LayoutInflater inflater, @Nullable
@org.jetbrains.annotations.Nullable
 ViewGroup container, @Nullable
@org.jetbrains.annotations.Nullable Bundle
 savedInstanceState) {
    View view =
inflater.inflate(R.layout.fragment3_layout,
container, false);
    WebView webView =
(WebView)view.findViewById(R.id.raspisan
ieWeb);

webView.getSettings().setJavaScriptEnabled(
true);
    webView.setWebViewClient(new
WebViewClient());
    webView.loadUrl("http://google.com/");
    return view;
}
}
```

ПРИЛОЖЕНИЕ Д

Код категории «Поиск» раздела «Объявления»

```
КодsearchadvДокументXML
<?xml version="1.0" encoding="utf-
8"?>
<androidx.constraintlayout.widget.Constraint
Layout
xmlns:android="http://schemas.android.com/a
pk/res/android"
xmlns:app="http://schemas.android.com/apk/r
es-auto"
xmlns:tools="http://schemas.android.com/tool
s"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".searchadv">
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvAdv"
    android:layout_width="409dp"
    android:layout_height="729dp"
    android:layout_marginStart="1dp"
    android:layout_marginLeft="1dp"
    android:layout_marginTop="1dp"
    android:layout_marginEnd="1dp"
    android:layout_marginRight="1dp"
    android:layout_marginBottom="1dp"
    app:layout_constraintBottom_toBottomOf="p
arent"app:layout_constraintEnd_toEndOf="pa
rent"
    app:layout_constraintStart_toStartOf="parent
"
```

```
app:layout_constraintTop_toTopOf="parent"
/>
</androidx.constraintlayout.widget.Constraint
Layout>
```

```
Кодsearchadv документ JAVA
public class searchadv extends
AppCompatActivity {
    RecyclerView recyclerView;
    DatabaseReference databaseAdv;
    MyAdapter1 myAdapter1;
    ArrayList<AdvSearchData> list;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_searchadv)
        ;
        recyclerView =
        findViewById(R.id.rvAdv);
        databaseAdv =
        FirebaseDatabase.getInstance().getReference(
        "Adv");
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new
LinearLayoutManager(this));
        list = new ArrayList<>();
        myAdapter1 = new
MyAdapter1(this,list);
        recyclerView.setAdapter(myAdapter1);
```

```

databaseAdv.addValueEventListener(new
ValueEventListener() {
    @Override
    public void onDataChange(@NonNull
@NotNull dataSnapshot snapshot) {
        for (DataSnapshot dataSnapshot :
snapshot.getChildren()){
            AdvSearchData advSearchData =
dataSnapshot.getValue(AdvSearchData.class)
;
            list.add(advSearchData);
        }
myAdapter1.notifyDataSetChanged();
    }
    @Override
    public void onCancelled(@NonNull
@NotNull DatabaseError error) {
    }
});
}
@Override
public boolean
onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.menu_item,
menu);

```

```

MenuItem menuItem =
menu.findItem(R.id.search_action);
SearchView searchView =(SearchView)
menuItem.getActionView();

searchView.setOnQueryTextListener(new
SearchView.OnQueryTextListener() {
    @Override
    public boolean
onQueryTextSubmit(String query) {
        return false;
    }
    @Override
    public boolean
onQueryTextChange(String newText) {
myAdapter1.getFilter().filter(newText);
        return false;
    }
});
return
super.onCreateOptionsMenu(menu);
}
}

```

ПРИЛОЖЕНИЕ Е

Код категории «Поиск» раздела «Для группы»

```
КодsearchgroupДокументXML
<?xml version="1.0" encoding="utf-
8"?>
<androidx.constraintlayout.widget.Constraint
Layout
xmlns:android="http://schemas.android.com/a
pk/res/android"
xmlns:app="http://schemas.android.com/apk/r
es-auto"
xmlns:tools="http://schemas.android.com/tool
s"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".searchgroup">
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvGrup"
    android:layout_width="409dp"
    android:layout_height="729dp"
    android:layout_marginStart="1dp"
    android:layout_marginLeft="1dp"
    android:layout_marginTop="1dp"
    android:layout_marginEnd="1dp"
    android:layout_marginRight="1dp"
    android:layout_marginBottom="1dp"
    app:layout_constraintBottom_toBottomOf="p
arent"
    app:layout_constraintEnd_toEndOf="parent"

    app:layout_constraintStart_toStartOf="parent
"
```

```
app:layout_constraintTop_toTopOf="parent"
/>
</androidx.constraintlayout.widget.Constraint
Layout>
```

```
КодsearchgroupДокумент JAVA
public class searchgroup extends
AppCompatActivity {
    RecyclerView recyclerView1;
    DatabaseReference databaseGroup;
    MyAdapter3 myAdapter3;
    ArrayList<GroupSearchData> list;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_searchgrou
p);
        recyclerView1 =
findViewById(R.id.rvGrup);
        databaseGroup =
FirebaseDatabase.getInstance().getReference(
"Gruppa");
        recyclerView1.setHasFixedSize(true);
        recyclerView1.setLayoutManager(new
LinearLayoutManager(this));
        list = new ArrayList<>();
        myAdapter3 = new
MyAdapter3(this,list);
        recyclerView1.setAdapter(myAdapter3);
```

```

databaseGroup.addValueEventListener(new
ValueEventListener() {
    @Override
    public void onDataChange(@NonNull
@NotNull dataSnapshot snapshot) {
        for (DataSnapshot dataSnapshot :
snapshot.getChildren()){
            GroupSearchData
groupSearchData =
dataSnapshot.getValue(GroupSearchData.clas
s);

            list.add(groupSearchData);
        }
    }
});

myAdapter3.notifyDataSetChanged();
}

@Override
public void onCancelled(@NonNull
@NotNull DatabaseError error) {
}
}
}

```

ПРИЛОЖЕНИЕ Ж

Код категории «Поиск» раздела «Дисциплины»

```
КодsearchpredmetДокументXML
<?xml version="1.0" encoding="utf-
8"?>
<androidx.constraintlayout.widget.Constraint
Layout
xmlns:android="http://schemas.android.com/a
pk/res/android"
xmlns:app="http://schemas.android.com/apk/r
es-auto"
xmlns:tools="http://schemas.android.com/tool
s"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".searchpredmet">
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvPred"
    android:layout_width="409dp"
    android:layout_height="729dp"
    android:layout_marginStart="1dp"
    android:layout_marginLeft="1dp"
    android:layout_marginTop="1dp"
    android:layout_marginEnd="1dp"
    android:layout_marginRight="1dp"
    android:layout_marginBottom="1dp"
    app:layout_constraintBottom_toBottomOf="p
arent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent
"
```

```
app:layout_constraintTop_toTopOf="parent"
/>
</androidx.constraintlayout.widget.Constraint
Layout>
КодsearchpredmetДокументJAVA
public class searchpredmet extends
AppCompatActivity {
    RecyclerView recyclerView;
    DatabaseReference databasePred;
    MyAdapter2 myAdapter2;
    ArrayList<PredmSearchData> list;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_searchpred
met);
        recyclerView =
findViewById(R.id.rvPred);
        databasePred =
FirebaseDatabase.getInstance().getReference(
"Predmet");
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new
LinearLayoutManager(this));
        list = new ArrayList<>();
        myAdapter2 = new
MyAdapter2(this, list);
        recyclerView.setAdapter(myAdapter2);
```



```

databasePred.addValueEventListener(new
ValueEventListener()
    {
        @Override
        public void onDataChange(@NonNull
@NotNull dataSnapshot snapshot) {
            for (DataSnapshot dataSnapshot :
snapshot.getChildren()){
                PredmSearchData
predmSearchData =
dataSnapshot.getValue(PredmSearchData.clas
s);
                list.add(predmSearchData);
            }
        }
    }
);
myAdapter2.notifyDataSetChanged();
}
@Override
public void onCancelled(@NonNull
@NotNull DatabaseError error) {
}
});
}
}

```

ПРИЛОЖЕНИЕ И

Код категории «Новая запись» раздела «Объявления»

```
Кодnew_adsДокументXML
<LinearLayout
xmlns:android="http://schemas.android.com/a
pk/res/android"
xmlns:app="http://schemas.android.com/apk/r
es-auto"
xmlns:tools="http://schemas.android.com/tool
s"
    android:id="@+id/linearLayout2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    tools:context=".activity_new_ads">
<TextView
    android:id="@+id/tvAds"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="100px"
    android:gravity="center"
    android:text="Title"
    android:textSize="20dp"
    android:textStyle="bold" />
<EditText
    android:id="@+id/edZag"
    android:layout_width="220dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="50px"
    android:ems="10"
    android:hint="text1"
    android:importantForAutofill="no"
    android:inputType="textPersonName" />
<EditText
    android:id="@+id/edSoder"
    android:layout_width="220dp"
    android:layout_height="184dp"
    android:layout_marginBottom="50px"
    android:ems="10"
    android:gravity="start|top"
    android:hint="multiline          text"
    android:importantForAutofill="no"
    android:inputType="textMultiLine"
    android:lineSpacingExtra="8sp"
    android:lines="5"
    android:maxLines="150"
    android:minLines="5"
    android:scrollbars="vertical"
    android:singleLine="true" />
<EditText
    android:id="@+id/edPrim"
    android:layout_width="220dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="150px"
    android:ems="10"
    android:hint="text2          (optional)"
    android:importantForAutofill="no"
    android:inputType="textPersonName" />
<Button
    android:id="@+id/BTNAdd"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="55dp"
        android:layout_marginTop="50dp"
        android:onClick="onClickAdd"
        android:text="Button          add"
        android:textStyle="bold"          />
</LinearLayout>

```

Кодnew_adsДокумент JAVA

```

public class activity_new_ads extends
AppCompatActivity {
    private EditText edZag, edSoder, edPrim;
    private DatabaseReference DataBaseAds;
    private String ADV = "Adv";
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_new_ads);
        init();
    }
    public void init() {
        edZag = findViewById(R.id.edZag);
        edSoder = findViewById(R.id.edSoder);
        edPrim = findViewById(R.id.edPrim);
        DataBaseAds =
FirebaseDatabase.getInstance().getReference(

```

```

ADV);
    }
    public void onClickAdd(View view) {
        String id = DataBaseAds.getKey();
        String zagolovok =
edZag.getText().toString();
        String soderjanie =
edSoder.getText().toString();
        String primechanie =
edPrim.getText().toString();
        Adv newAdv = new Adv(id, zagolovok,
soderjanie, primechanie);
        if (!TextUtils.isEmpty(zagolovok) &&
!TextUtils.isEmpty(soderjanie)) {
            DataBaseAds.push().setValue(newAdv);
            Toast.makeText(this,
"Записьдобавлена",
Toast.LENGTH_SHORT).show();
            this.finish();
        } else {
            Toast.makeText(this,
"Невсеполязаполнены",
Toast.LENGTH_SHORT).show();
        }
    }
}

```

ПРИЛОЖЕНИЕ К

Код категории «Новая запись» раздела «Для группы»

```
КодгруппаДокументXML
<LinearLayout
xmlns:android="http://schemas.android.com/a
pk/res/android"
xmlns:app="http://schemas.android.com/apk/r
es-auto"
xmlns:tools="http://schemas.android.com/tool
s"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    tools:context=".Gruppa">
<TextView
    android:id="@+id/tvGrup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="100px"
    android:gravity="center"
    android:text="title"
    android:textSize="20dp"
    android:textStyle="bold" />
<EditText
    android:id="@+id/edNum"
    android:layout_width="220dp"

КодгруппаДокумент JAVA
public class Gruppa extends
AppCompatActivity {

private EditText edNum, edInf, edDop;
private DatabaseReference DataBaseGrup;
private String GRUPPA = "Gruppa";
@Override
protected void onCreate(Bundle
savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_gruppa);
    init();
}

public void init() {
    edNum = findViewById(R.id.edNum);
    edInf = findViewById(R.id.edInf);
    edDop = findViewById(R.id.edDop);
    DataBaseGrup =
FirebaseDatabase.getInstance().getReference(
GRUPPA);
}

public void onClickAddGr(View view) {
    String id = DataBaseGrup.getKey();
    String nomer_gruppy =
edNum.getText().toString();
    String inform =
edInf.getText().toString();
    String dopolnitelno =
edDop.getText().toString();
    GRUP newGrup = new GRUP(id,
nomer_gruppy, inform, dopolnitelno);
    if (!TextUtils.isEmpty(nomer_gruppy)
```

```

    && !TextUtils.isEmpty(inform)) {
        DataBaseGrup.push().setValue(new Grup);
        Toast.makeText(this,
            "Запись добавлена",
            Toast.LENGTH_SHORT).show();
        this.finish();
    } else {
        Toast.makeText(this,
            "Не все поля заполнены",
            Toast.LENGTH_SHORT).show();
    }
}

```

ПРИЛОЖЕНИЕ Л

Код категории «Новая запись» раздела «Дисциплины»

```
КодpredmetyДокументXML
<LinearLayout
xmlns:android="http://schemas.android.com/a
pk/res/android"
xmlns:app="http://schemas.android.com/apk/r
es-auto"
xmlns:tools="http://schemas.android.com/tool
s"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    tools:context=".Predmety">
<TextView
    android:id="@+id/tvPredm"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="100px"
    android:gravity="center"
    android:text="Title"
    android:textSize="20dp"
    android:textStyle="bold" />
<EditText
    android:id="@+id/edNameP"
    android:layout_width="220dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="50px"
    android:ems="10"
    android:hint="text1"
    android:importantForAutofill="no"
    android:inputType="textPersonName" />
<EditText
    android:id="@+id/edIP"
    android:layout_width="220dp"
    android:layout_height="184dp"
    android:layout_marginBottom="50px"
    android:ems="10"
    android:gravity="start|top"
    android:hint="multiline          text"
    android:importantForAutofill="no"
    android:inputType="textMultiLine"
    android:lineSpacingExtra="8sp"
    android:lines="5"
    android:maxLines="150"
    android:minLines="5"
    android:scrollbars="vertical"
    android:singleLine="true" />
<EditText
    android:id="@+id/edTeacher"
    android:layout_width="220dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="150px"
    android:ems="10"
    android:hint="text2"
    android:importantForAutofill="no"
    android:inputType="textPersonName" />
<Button
    android:id="@+id/BTNAddPredm"
    android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:layout_marginLeft="55dp"
        android:layout_marginTop="50dp"
        android:onClick="onClickAddGr"
        android:text="Button          add"
        android:textStyle="bold"          />
</LinearLayout>

```

```

КодpredmetryДокументJAVA
public class Predmetry extends
AppCompatActivity {
    private EditText    edNameP,    edIP,
edTeacher;
    private              DatabaseReference
DataBasePredmet;
    private String PREDMET = "Predmet";
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_predmetry);
        init();
    }
    public void init() {
        edNameP =
findViewById(R.id.edNameP);
        edIP = findViewById(R.id.edIP);
        edTeacher =
findViewById(R.id.edTeacher);
        DataBasePredmet =

```

```

FirebaseDatabase.getInstance().getReference(
PREDMET);
    }
    public void onClickAddGr(View view) {
        String id = DataBasePredmet.getKey();
        String nazvanie =
edNameP.getText().toString();
        String about_p =
edIP.getText().toString();
        String teacher =
edTeacher.getText().toString();
        PREDM newPredmet = new PREDM(id,
nazvanie, about_p, teacher);
        if (!TextUtils.isEmpty(nazvanie) &&
!TextUtils.isEmpty(about_p) &&
!TextUtils.isEmpty(teacher)) {
            DataBasePredmet.push().setValue(newPredm
et);
            Toast.makeText(this,
"Записьдобавлена",
Toast.LENGTH_SHORT).show();
            this.finish();
        } else {
            Toast.makeText(this,
"Невсеполязаполнены",
Toast.LENGTH_SHORT).show();
        }
    }
}

```

ПРИЛОЖЕНИЕ М

Код вывода списков информации из раздела «Объявления»

```
Кодview_adsДокументXML
<?xml version="1.0" encoding="utf-
8"?>
<androidx.constraintlayout.widget.Constraint
Layout
xmlns:android="http://schemas.android.com/a
pk/res/android"
xmlns:app="http://schemas.android.com/apk/r
es-auto"
xmlns:tools="http://schemas.android.com/tool
s"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ViewAds">
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="50dp"
    app:cardCornerRadius="20dp">
<androidx.constraintlayout.widget.Constraint
Layout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:scrollbarAlwaysDrawVerticalTrack=
"false">
<TextView
    android:id="@+id/ViewZag"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="127dp"
    android:layout_marginLeft="127dp"
    android:layout_marginTop="80dp"
    android:layout_marginEnd="127dp"
    android:layout_marginRight="127dp"
    android:gravity="center"
    android:text="TextView"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    "
    app:layout_constraintTop_toTopOf="parent"
/>
<TextView
    android:id="@+id/ViewSoder"
    android:layout_width="277dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="127dp"
    android:layout_marginLeft="127dp"
    android:layout_marginTop="43dp"
    android:layout_marginEnd="127dp"
    android:layout_marginRight="127dp"
    android:scrollbarAlwaysDrawHorizontalTrac
k="false"
    android:text="TextView"
    android:textSize="18sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    "
```



```

app:layout_constraintTop_toBottomOf="@+id/ViewZag"
<ScrollView
    android:layout_width="289dp"
    android:layout_height="416dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.521"
"
app:layout_constraintStart_toStartOf="parent"
"
app:layout_constraintTop_toBottomOf="@+id/ViewSoder"
app:layout_constraintVertical_bias="0.857">
<TextView
    android:id="@+id/viewPrim"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="TextView"
    android:textSize="18sp"
</ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.cardview.widget.CardView>
</androidx.constraintlayout.widget.ConstraintLayout>

```

```

Кодview_adsДокументJAVA
public class ViewAds extends
AppCompatActivity {
    TextView Zag,Soder,Prim;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_ads);
        Zag = findViewById(R.id.ViewZag);
        Soder = findViewById(R.id.ViewSoder);
        Prim = findViewById(R.id.viewPrim);
        String
zagolovok=getIntent().getStringExtra("zagolo
vok");
        String
soderjanie=getIntent().getStringExtra("soderj
anie");
        String
primechanie=getIntent().getStringExtra("prim
echanie");
        Zag.setText(zagolovok);
        Soder.setText(soderjanie);
        Prim.setText(primechanie);
    }
}

```

ПРИЛОЖЕНИЕ Н

Код вывода списков информации из раздела «Для группы»

```
Кодview_группаДокументXML
<?xml version="1.0" encoding="utf-
8"?>
<androidx.constraintlayout.widget.Constraint
Layout
xmlns:android="http://schemas.android.com/a
pk/res/android"
xmlns:app="http://schemas.android.com/apk/r
es-auto"
xmlns:tools="http://schemas.android.com/tool
s"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ViewГруппа">
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="50dp"
    app:cardCornerRadius="20dp">
<androidx.constraintlayout.widget.Constraint
Layout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:scrollbarAlwaysDrawVerticalTrack=
"false">
<TextView
    android:id="@+id/ViewNomer"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="127dp"
    android:layout_marginLeft="127dp"
    android:layout_marginTop="80dp"
    android:layout_marginEnd="127dp"
    android:layout_marginRight="127dp"
    android:gravity="center"
    android:scrollbarAlwaysDrawVerticalTrack=
"false"
    android:text="TextView"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
/>
<TextView
    android:id="@+id/ViewInfo"
    android:layout_width="277dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="127dp"
    android:layout_marginLeft="127dp"
    android:layout_marginTop="43dp"
    android:layout_marginEnd="127dp"
    android:layout_marginRight="127dp"
    android:text="TextView"
    android:textSize="18sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
```

```

"
app:layout_constraintTop_toBottomOf="@+id/ViewNomer"
<ScrollView
    android:layout_width="289dp"
    android:layout_height="402dp"
    android:layout_marginTop="48dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.521"
    "
    app:layout_constraintStart_toStartOf="parent"
    "
    app:layout_constraintTop_toBottomOf="@+id/ViewInfo">
<TextView
    android:id="@+id/viewDop"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="TextView"
    android:textSize="18sp"
</ScrollView>
</androidx.constraintlayout.widget.Constraint
Layout>
</androidx.cardview.widget.CardView>
</androidx.constraintlayout.widget.Constraint
Layout>

```

```

Кодview_группаДокументJAVA
public class ViewGruppa extends
AppCompatActivity {
    TextView Num,Inf,Dop;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_grup
pa);
        Num = findViewById(R.id.ViewNomer);
        Inf = findViewById(R.id.ViewInfo);
        Dop = findViewById(R.id.viewDop);
        String nomer_gruppy
=getIntent().getStringExtra("nomer_gruppy");
        String inform
=getIntent().getStringExtra("inform");
        String dopolnitelno
=getIntent().getStringExtra("dopolnitelno");
        Num.setText(nomer_gruppy);
        Inf.setText(inform);
        Dop.setText(dopolnitelno);
    }
}

```

ПРИЛОЖЕНИЕ П

Код вывода списков информации из раздела «Дисциплины»

```
Кодview_predmДокументXML
<?xml version="1.0" encoding="utf-
8"?>
<androidx.constraintlayout.widget.Constraint
Layout
xmlns:android="http://schemas.android.com/a
pk/res/android"
xmlns:app="http://schemas.android.com/apk/r
es-auto"
xmlns:tools="http://schemas.android.com/tool
s"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ViewPredm">
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="50dp"
    app:cardCornerRadius="20dp">
<androidx.constraintlayout.widget.Constraint
Layout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
<TextView
    android:id="@+id/ViewNazv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="127dp"
    android:layout_marginLeft="127dp"
    android:layout_marginTop="80dp"
    android:layout_marginEnd="127dp"
    android:layout_marginRight="127dp"
    android:gravity="center"
    android:text="TextView"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.481
"
    app:layout_constraintStart_toStartOf="parent
"
    app:layout_constraintTop_toTopOf="parent"
/>
<TextView
    android:id="@+id/viewTeacher"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="127dp"
    android:layout_marginLeft="127dp"
    android:layout_marginTop="44dp"
    android:layout_marginEnd="127dp"
    android:layout_marginRight="127dp"
    android:text="TextView"
    android:textSize="18sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.496
"
    app:layout_constraintStart_toStartOf="parent
"
```

```

app:layout_constraintTop_toBottomOf="@+id/ViewNazv"
<ScrollView
    android:layout_width="289dp"
    android:layout_height="414dp"
    android:layout_marginTop="36dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.521"
    "
    app:layout_constraintStart_toStartOf="parent"
    "
    app:layout_constraintTop_toBottomOf="@+id/viewTeacher">
<TextView
    android:id="@+id/ViewOpis"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="TextView"
    android:textSize="18sp"
</ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.cardview.widget.CardView>
</androidx.constraintlayout.widget.ConstraintLayout>

```

```

Кодview_predmДокументJAVA
public class ViewPredm extends
AppCompatActivity {
    TextView Naz,Opis,Teacher;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_pred

```

```

m);
    Naz = findViewById(R.id.ViewNazv);
    Opis = findViewById(R.id.ViewOpis);
    Teacher =
findViewById(R.id.viewTeacher);
    String nazvanie
= getIntent().getStringExtra("nazvanie");
    String opisanie
= getIntent().getStringExtra("about_p");
    String teacher
= getIntent().getStringExtra("teacher");
    Naz.setText(nazvanie);
    Opis.setText(opisanie);
    Teacher.setText(teacher);
}
}

```

Выпускная квалификационная работа выполнена мной самостоятельно.
Использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Отпечатано в одном экземпляре.

Библиография 20 наименований.

Один экземпляр сдан на кафедру.

« ____ » _____ 2021 г.

_____ Семенович Артем Александрович
подпись

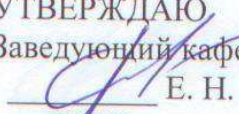
Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ

Заведующий кафедрой


Е. Н. Скуратенко

подпись

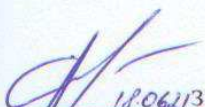
«18» » 06 2021 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 Прикладная информатика

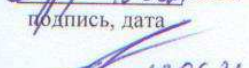
Разработка шаблона мобильного приложения

Руководитель


18.06.21
подпись, дата

зав. кафедрой, канд. техн. наук Е.Н. Скуратенко

Выпускник


18.06.21
подпись, дата

А. А. Семенович

Консультанты
по разделам:

Экономический


18.06.21
подпись, дата

Е. Н. Скуратенко

Нормоконтролер


18.06.21
подпись, дата

В. И. Кокова

Абакан 2021

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ

Заведующий кафедрой

Е.Н. Скуратенко

подпись

« 08 » апреля 2021 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Студенту Семенович Артему Александровичу

Группа ХБ 17-03

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: Разработка шаблона мобильного приложения

Утверждена приказом по институту № 222 от 08.04.2021 г.

Руководитель ВКР: зав. кафедрой, канд. техн. наук Е.Н. Скуратенко, ХТИ – филиал СФУ

Исходные данные для ВКР: заказ ХТИ – филиала СФУ.

Перечень разделов ВКР:

1. Анализ предметной области. Выбор средств проектных решений.
2. Описание разработки шаблона.
3. Расчёт затрат и оценка экономической эффективности реализации шаблона.

Перечень графического материала: нет

Руководитель ВКР


подпись

Е.Н. Скуратенко

Задание принял к исполнению


подпись

А. А. Семенович

«08» апреля 2021 г.