

Министерство науки и высшего образования РФ  
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  
«\_\_\_\_\_» \_\_\_\_\_ 2022 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.03 Прикладная информатика

Разработка мобильного приложения «Детская безопасность» для ЦУР  
Республики Хакасия

Руководитель \_\_\_\_\_ ст. преподаватель В. И. Кокова  
подпись, дата

Выпускник \_\_\_\_\_ А. В. Белицкий  
подпись, дата

Консультанты  
по разделам:

Экономический \_\_\_\_\_ Е. Н. Скуратенко  
подпись, дата

Нормоконтролер \_\_\_\_\_ В. И. Кокова  
подпись, дата

Абакан 2022

Министерство науки и высшего образования РФ  
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  
«\_\_\_\_\_» \_\_\_\_\_ 2022 г.

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

Абакан 2022

Студенту Белицкому Андрею Валерьевичу

Группа ХБ 18-03

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: Разработка мобильного приложения «Детская безопасность» для ЦУР Республики Хакасия

Утверждена приказом по институту № 208 от 14.04.2022 г.

Руководитель ВКР: В. И. Кокова, ст. преподаватель, ХТИ – филиал СФУ

Исходные данные для ВКР: заказ ЦУР Республики Хакасия

Перечень разделов ВКР:

1. Анализ предметной области.
2. Разработка мобильного приложения «Детская безопасность».
3. Оценка затрат на реализацию мобильного приложения «Детская безопасность».

Перечень графического материала: нет

Руководитель ВКР

\_\_\_\_\_

подпись

В. И. Кокова

Задание принял к исполнению

\_\_\_\_\_

подпись

А. В. Белицкий

«14» апреля 2022 г.

## РЕФЕРАТ

Выпускная квалификационная работа по теме: «Разработка мобильного приложения «Детская безопасность» для ЦУР Республики Хакасия» содержит 76 страниц текстового документа, 52 иллюстрации, 11 использованных источников, 7 формул и 8 таблиц.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, ПОЛЬЗОВАТЕЛЬ, ANDROID, ДЕТСКАЯ БЕЗОПАСНОСТЬ, ЦЕНТР УПРАВЛЕНИЯ РЕГИОНОМ, PUSH-УВЕДОМЛЕНИЕ, SDK, ANDROID STUDIO.

Цель выпускной квалификационной работы: разработать мобильное приложение для помощи детям, попавшим в трудную ситуацию.

Задачи:

1. охарактеризовать основную деятельность Центра управления регионом Республики Хакасия;
2. обосновать выбор инструментальных средств разработки;
3. разработать мобильное приложение «Детская безопасность»;
4. оценить затраты на реализацию мобильного приложения;
5. проанализировать возможные риски и пути их решения.

Первый раздел посвящен вопросам анализа предметной области и исследования Центра управления регионом Республики Хакасия (ЦУР РХ) и выбору программных средств для разработки приложения. Второй раздел содержит описание этапов разработки мобильного приложения. В третьем разделе проведен расчет капитальных и эксплуатационных затрат, описана социальная польза мобильного приложения и риски проекта.

На основе анализа предметной области выбраны средства разработки: ANDROID STUDIO, JAVA.

На основе расчетов капитальных и эксплуатационных затрат, определена стоимость мобильного приложения: ТСО = 89 918 рублей.

## SUMMARY

The theme of the graduation thesis is «Development of Mobile Application ‘Children’s Safety’ for Regional Management Center of Republic of Khakassia». It contains 76 pages of a text document, 52 figures, 11 references, 7 formulae, 8 tables.

MOBILE APP, USER, ANDROID, CHILD SAFETY, REGIONAL CONTROL REGIONAL MANAGEMENT CENTER, PUSH-NOTIFICATION, SDK, ANDROID STUDIO.

The purpose of the thesis: to develop a mobile application to help children in difficult situations.

Objectives:

1. to characterize the framework of Regional Management Center of Republic of Khakassia;
2. to support the choice of development tools;
3. to develop a mobile application “Children’s safety”;
4. to estimate economic costs of implementing a mobile application;
5. to analyze possible risks and ways to solve them.

The first section is devoted to analysis of the subject area of the Regional Management Center of Republic of Khakassia and the choice of software tools for developing the application as well. The second section contains a description of stages of mobile application development. The third section deals with the calculation of capital and operating costs, social benefits of the mobile application and risks of the project described.

The following development tools have been selected: ANDROID STUDIO, JAVA.

The capital and operating costs have been calculated; the cost of the mobile application has been determined: 89,918 rubles.

## СОДЕРЖАНИЕ

Введение.....	8
1 Анализ предметной области .....	9
1.1 Описание основной деятельности Центра управления регионом Республики Хакасия.....	9
1.2 Обоснование актуальности разработки программного продукта....	13
1.3 Обзор аналогов мобильного приложения «Детская безопасность»	14
1.4 Постановка цели и задач разработки .....	17
1.5 Выбор средств разработки .....	17
1.5.1 Выбор среды разработки .....	17
1.5.2 Выбор языка программирования.....	22
1.6 Модель взаимодействия пользователя с мобильным приложением «Детская безопасность».....	26
1.7 Выбор архитектуры мобильного приложения.....	28
1.8 Вывод по разделу «Анализ предметной области» .....	29
2 Разработка мобильного приложения «Детская безопасность» .....	30
2.1 Создание и настройка проекта в Android Studio.....	30
2.2 Подключение мобильного приложения к Firebase.....	33
2.3 Разработка кнопок мобильного приложения .....	38
2.4 Выводы по разделу «Разработка мобильного приложения «Детская безопасность» .....	59
3 Оценка затрат на реализацию мобильного приложения «Детская безопасность» .....	59
3.1 Капитальные затраты.....	59

3.2 Расчет эксплуатационных затрат .....	66
3.3 Расчет совокупной стоимости владения мобильным приложением	69
3.4 Риски реализации проекта .....	69
3.5 Мероприятия по снижению рисков.....	71
3.6 Выводы по разделу «Оценка затрат на реализацию мобильного приложения «Детская безопасность».....	71
Заключение .....	73
Список использованных источников .....	74

## ВВЕДЕНИЕ

Совсем недавно человечество не могло представить такую цифровизацию мира. В сегодняшние дни у каждого человека под рукой гаджет с доступом к всемирной паутине Интернет, где можно найти практически любую информацию. Поэтому такой возможностью можно воспользоваться, а скорее даже нужно. Сейчас почти у каждого ребенка есть смартфон с доступом в Интернет, этим можно воспользоваться для повышения безопасности детей в экстренных ситуациях.

Мобильное приложение позволит предоставить быстрый доступ к инструкции, что делать в определенной ситуации, а кнопки быстрого доступа позволят связаться с экстренными службами или службами психологической помощи.

Цель выпускной квалификационной работы: разработка мобильного приложения «Детская безопасность» для помощи детям, попавшим в трудную ситуацию.

Для достижения цели необходимо решить следующие задачи:

1. охарактеризовать основную деятельность Центра управления регионом Республики Хакасия;
2. обосновать выбор инструментальных средств разработки;
3. разработать мобильное приложение «Детская безопасность»;
4. оценить затраты на реализацию мобильного приложения;
5. проанализировать возможные риски и пути их решения.



## **1 Анализ предметной области**

### **1.1 Описание основной деятельности Центра управления регионом Республики Хакасия**

Главная задача ЦУР – информирование граждан и обслуживание их запросов. Для информирования граждан о важных федеральных, региональных и местных новостях силами ЦУР созданы/визуально оформлены и переданы для работы 36 официальных групп органов местного самоуправления во всех социальных сетях.

Для того, чтобы реализовать национальные проекты в сфере технологий были созданы ЦУРЫ.

ЦУР является организацией, которая помогает в решении социальных проблем, курируемых Правительством Республики Хакасия, с помощью цифровизации. В связи с этим разрабатывается совместный проект с МЧС «Детская безопасность». Для данной цели уже созданы telegram бот и viber бот.

Основными задачами Центра управления регионом Республики Хакасии являются:

1) координация работы по мониторингу и обработке обращений и сообщений, направленных в исполнительные органы государственной власти Республики Хакасия, органы местного самоуправления муниципальных образований Республики Хакасия (далее – органы местного самоуправления) или должностным лицам указанных органов в письменной форме или в форме электронного документа;

2) мониторинг и контроль исполнения управленческих решений;

3) мониторинг исполнения планов и программ, а также достижения установленных ими значений целевых показателей;

4) обеспечение сбора, обработки и анализа информации, характеризующей общественно-политическую, социально-экономическую и

информационную обстановку в Республике Хакасия и необходимой для планирования, подготовки и принятия управленческих решений;

5) обеспечение информационной поддержки принятия управленческих решений, проведение совещаний в очном и дистанционных режимах;

6) обеспечение необходимого уровня защиты информации, обрабатываемой в ЦУР.

В соответствии с задачами ЦУР реализуются следующие функции и полномочия:

1) формирование оценки работы исполнительных органов государственной власти Республики Хакасия и органов местного самоуправления при обработке обращений и сообщений;

2) выработка рекомендаций для определения приоритетных направлений в работе исполнительных органов государственной власти Республики Хакасия, органов местного самоуправления, республиканских государственных учреждений и муниципальных учреждений в Республике Хакасия, организаций, созданных с участием Республики Хакасия или муниципальным участием (далее – учреждения и организации);

3) проведение аналитических исследований и опросов по вопросам обработки обращений и сообщений;

4) выявление и анализ лучших практик, их применение для оперативного решения проблем, обнаруженных в ходе анализа поступивших обращений и сообщений, мониторинга и контроля исполнения решений, во взаимодействии исполнительных органов государственной власти Республики Хакасия и органов местного самоуправления;

5) обеспечение оперативного доступа к информационному фонду ЦУР и информационная поддержка пользователей ЦУР в целях принятия и исполнения управленческих решений;

6) оценка эффективности взаимодействия исполнительных органов государственной власти Республики Хакасия, а также контроль в

установленном законодательством порядке принятых ими в процессе работы управленческих решений;

7) мониторинг качества предоставления исполнительными органами государственной власти Республики Хакасия и органами местного самоуправления государственных и муниципальных услуг;

8) мониторинг социально-экономического развития, общественно-политической ситуации, комплексной безопасности Республики Хакасия;

9) разработка специализированных материалов с последующим адресным донесением до целевой аудитории;

10) выстраивание взаимодействия с региональными средствами массовой информации, средствами электронной коммуникации и лидерами общественного мнения;

11) проведение образовательных семинаров для профильных специалистов исполнительных органов государственной власти Республики Хакасия и органов местного самоуправления;

12) выявление конфликтных ситуаций и ошибок при коммуникации исполнительных органов государственной власти Республики Хакасия и органов местного самоуправления с жителями Республики Хакасия;

13) формирование рекомендаций по онлайн-взаимодействию исполнительных органов государственной власти Республики Хакасия и органов местного самоуправления с жителями Республики Хакасия, предложений по разработке соответствующих сервисов;

14) координация формирования и сопровождения системы официальных страниц исполнительных органов государственной власти Республики Хакасия и органов местного самоуправления в электронных средствах массовой коммуникации на территории Республики Хакасия;

15) обеспечение удобной и наглядной визуализации и представления Главе Республики Хакасия – Председателю Правительства Республики Хакасия, Правительству Республики Хакасия и пользователям ЦУР результатов деятельности ЦУР;

- 16) документирование и видеозапись мероприятий, проводимых в ЦУР;
- 17) обеспечение подготовки информационных, информационно-справочных и информационно-аналитических материалов к мероприятиям, проводимым в ЦУР.

Структура ЦУР Республики Хакасия представлена в таблице 1.

Таблица 1 – Структура ЦУР Республики Хакасия

№ п/п	Наименование органа / должность	Роль
1	2	3
1	Аппарат Главы Республики Хакасия – Председателя Правительства Республики Хакасия	координатор
2	Государственный комитет цифрового развития и связи Республики Хакасия	технический куратор
3	Руководитель обособленного подразделения Центра компетенций ЦУР в Республике Хакасия (по согласованию)	руководитель
4	Руководители/заместители руководителей исполнительных органов государственной власти Республики Хакасия по тематикам: жилищно-коммунальное хозяйство, твердые коммунальные отходы, энергетика, образование, здравоохранение, дороги, транспорт, социальная защита	ответственные за отраслевые блоки по тематикам
5	Руководитель пресс-службы Главы Республики Хакасия – Председателя Правительства Республики Хакасия и Правительства Республики Хакасия	ответственный за работу с масс-медиа
6	Сотрудники исполнительных органов государственной власти/подведомственных учреждений Республики Хакасия по тематикам: жилищно-коммунальное хозяйство, твердые коммунальные отходы, энергетика, образование, здравоохранение, дороги, транспорт, социальная защита	руководители отраслевых блоков Центра управления регионом Республики Хакасия по тематикам

## Окончание таблицы 1

7	Сотрудники обособленного подразделения Центра компетенций ЦУР в Республике Хакасия (по согласованию)	руководители направлений Центра управления регионом Республики Хакасия
---	--	--

### 1.2 Обоснование актуальности разработки программного продукта

Актуальность проблемы с детской безопасностью и психологическими проблемами стоит остро в России. По уровню самоубийств среди детей и подростков Россия расположена на одном из первых мест в мире – средний показатель самоубийств среди подростков и детей более чем в 3 раза превышает средний показатель в мире.

С начала 1990-х годов коэффициент самоубийств среди подростков почти удвоился. Достигнув самого высокого показателя в 2002 году, он несколько снизился в 2004 году до 20,4 на 100 тысяч человек. Юноши гораздо чаще совершают самоубийства, чем девушки (32,8 на 100 тысяч человек против 7,6 – в 2004 году) (Росстат, 2004 год).

Если коэффициент самоубийств среди девушек оставался почти стабильным, то среди юношей этот показатель, начиная с 1989 года, удвоился.

Республика Хакасия имеет один из самых высоких коэффициентов самоубийств подростков в России – 49,1 на 100 тысяч подростков.

В текущей ситуации необходимо разрабатывать методы и инструменты для решения вопросов, приведенных в тексте выше. Одним из возможных таких инструментов может являться смартфон, так как он находится рядом практически у всех подростков. Государство беспокоится, чтобы интернет был безопасен. Проект направлен на создание безопасной среды.

ЦУР является совместным проектом с правительством Республики Хакасия и в рамках информационного проекта «Детская безопасность» занимается разработкой данных инструментов [5].

### 1.3 Обзор аналогов мобильного приложения «Детская безопасность»

Всего рассмотрим 2 самых популярных идентичных приложений – «SOS Alert» и «Безопасность – Помощь – SOS». В целом, функционал приложений похож, но есть некоторые отличия. Разберем приложение «SOS Alert». При запуске приложения появляются запросы на подтверждение использования SMS сообщений и местоположения. После подтверждения появляется инструкция, как пользоваться приложением. Инструкция выполнена на английском языке. Инструкция изображена на рисунке 1.

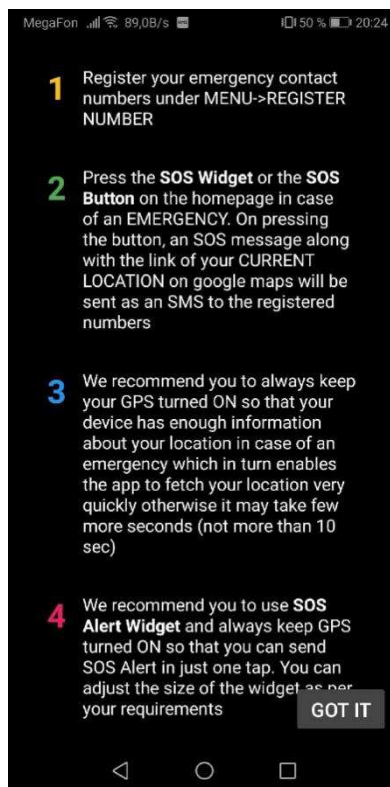


Рисунок 1 – Инструкция приложения «SOS Alert»

На главном меню красная кнопка. Основной функционал приложения – это отправка SMS сообщений с просьбой о помощи и местоположением пользователя зарегистрированным номерам в приложении. В приложении присутствует отдельное окно для записи номеров, на которые будет отправлено SMS сообщение.

Следующее приложение «Безопасность – Помощь – SOS» работает аналогичным образом, но оно на русском языке. При нажатии на кнопку, изображенную на рисунке 2, отправляется SMS сообщение зарегистрированным номерам с просьбой о помощи и местоположением.

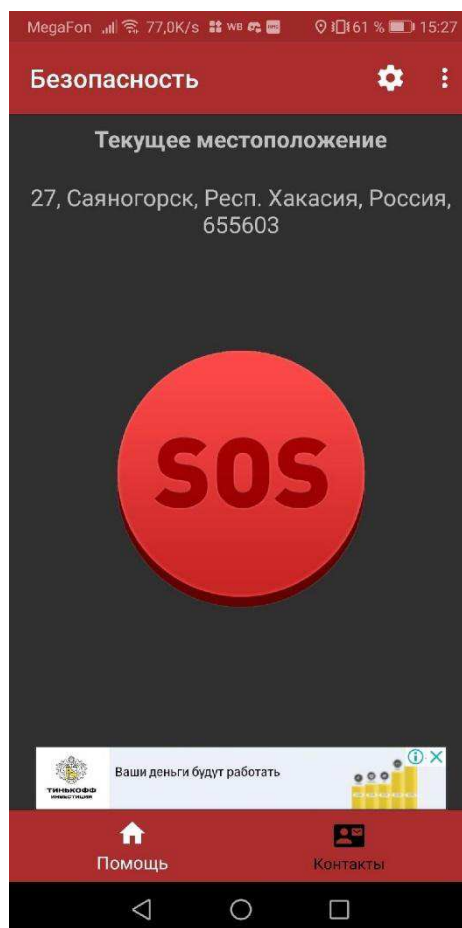


Рисунок 2 – Главное меню приложения

Во вкладке «Контакты» можно регистрировать номера, на которые отправляется сообщение. Для регистрации контакта необходимо будет ввести имя контакта и номер телефона.

Проанализировав аналоги, выделим 2 важные вещи:

- 1) Не все приложения разработаны на русском языке.
- 2) Приложения направлены на звонок зарегистрированным контактам и не имеют инструкций и плана действий при ЧС.

С 2018 года в России стали формироваться Центры управления регионами или сокращенно ЦУР. Эта государственная структура создана для того, чтобы оперативно реагировать на жалобы россиян не только по прямому обращению в органы власти, но и на информацию в социальных сетях, газетах и иных общедоступных источниках.

По решению президента Центр управления регионом должен быть в каждом субъекте России. Вся оперативная информация по обстановке в регионе стекается сюда и распределяется по отраслям: здравоохранение, образование, предпринимательство и т.д.

Важной составляющей центра является платформа обратной связи. В единую базу поступают обращения граждан через горячие линии, социальные сети, мессенджеры, официальные инстанции. ЦУР обрабатывает их и направляет в уполномоченные органы для оперативного решения поставленной проблемы

Центр как принимает жалобы по конкретным вопросам, так и анализирует информацию из различных публичных источников по актуальным для граждан вопросам. Исходя из практики работы ЦУР, жители обращаются по таким темам, как:

- некачественный ремонт дорог;
- установка мусорных контейнеров;
- школьное питание;
- низкие темпы строительства социальных объектов;
- качество и доступность медицинской помощи;
- вакцинация от коронавируса;
- проблемы с социальными пособиями.

После принятия жалобы сотрудники дают промежуточный ответ о направлении информации в ответственный орган, далее сообщают о принятом решении. Весь массив поступившей информации анализируется и предоставляется региональным властям в виде аналитического отчета.



Еще одним направлением деятельности является информирование. Так, в ряде регионов запускаются системы автоматического информирования граждан по насущным вопросам. Например, в Ленинградской области жители могут получить информацию о детских пособиях, в Оренбургской области чат-бот дает ответы на вопросы о бесплатной газификации, а во Владимирской области бот помогает решать вопрос с отсутствием сертификата о вакцинации от коронавируса на госуслугах.

## **1.4 Постановка цели и задач разработки**

Основанием для разработки послужила необходимость в создании мобильного приложения, которое поможет снизить количество несчастных случаев.

Целью данного проекта служит разработка мобильного приложения для помощи детям, попавшим в трудную ситуацию.

Задачами для достижения поставленной цели являются:

1. охарактеризовать основную деятельность Центра управления регионом Республики Хакасия;
2. обосновать выбор инструментальных средств разработки;
3. разработать мобильное приложение «Детская безопасность»;
4. оценить затраты на реализацию мобильного приложения;
5. проанализировать возможные риски и пути их решения.

## **1.5 Выбор средств разработки**

### **1.5.1 Выбор среды разработки**

Для разработки мобильного приложения потребуется среда разработки. На сегодняшний момент существует большой выбор IDE для мобильных приложений на Android и IOS. Разберем популярные IDE на сегодняшний день.

IDE – среда разработки, текстовый редактор, направленный на написание на одном или нескольких языках программирования. Программа выполняет несколько задач — упрощает работу с кодом и подготавливает рабочее окружение.

Программа понимает структуру языка программирования и выполняет следующие задачи:

— Подсказывает возможные ошибки, например, если забыли закрыть скобку. Программа сразу напомнит об этом и не придется тратить время на запуск, проверку сообщений об ошибках и правку. Это можно сравнить с подчеркиванием ошибок в Word.

— Когда используется какая-либо функция, показывает документацию для нее.

— Помогает быстро переименовать переменную. Если переменная использована в нескольких местах, все можно править в один клик.

— По мере написания кода ищет в нем ошибки и указывает на них.

В реальной жизни разработчик пишет меньше половины кода – остальное генерирует или подсказывает IDE. На рисунке 3 изображен интерфейс IDE Eclipse.

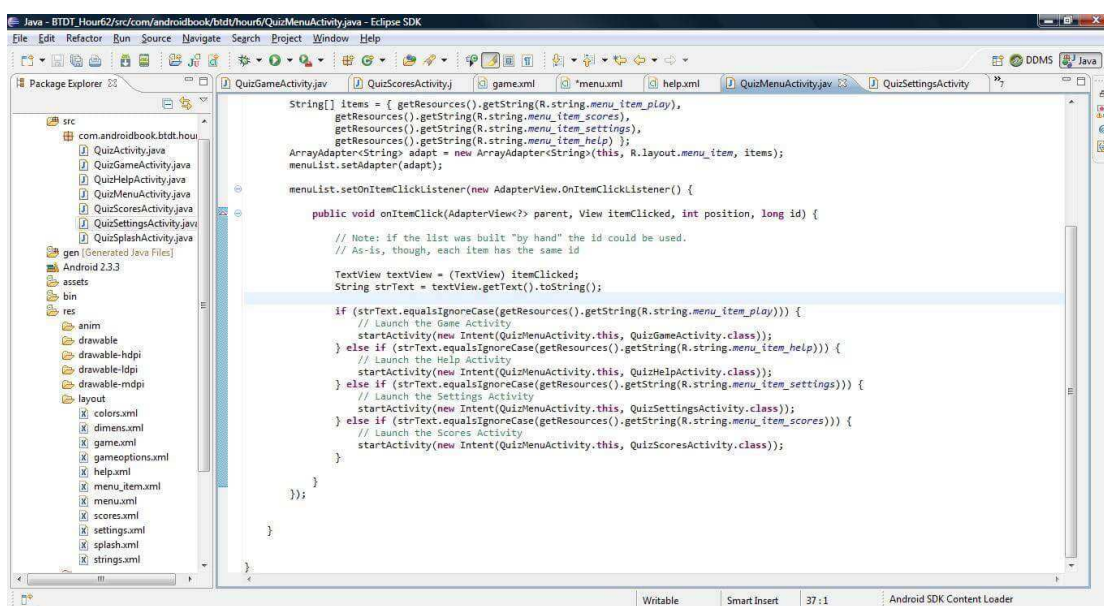


Рисунок 3 – Интерфейс IDE Eclipse

Это бесплатная IDE, разработанная некоммерческой компанией Eclipse Foundation. Эта программа является базой, которая регулирует процессы создания приложений. Преимущества Eclipse:

- интерфейс переведен на русский язык (документация прилагается);
- отлично «гоняет» на компьютерах с низкой производительностью;
- имеет дополнительные функции для серверной работы и анализа базы данных;
- может подключаться к модулям;
- может работать в групповом режиме, когда проект создают несколько человек одновременно.

До появления Android Studio, в качестве основного инструмента для разработки Android-приложений, разработчики использовали Eclipse. Этот IDE поддерживает несколько различных языков программирования, в том числе и Java с Android SDK. В отличие от Android Studio, Eclipse не предлагает встроенную поддержку и требует более тщательной настройки. На данный момент Google отключила официальную поддержку IDE Eclipse, в связи с чем нет оснований ее использовать.

Рассмотрим следующий вариант IDE – IDE IntelliJ IDEA. На рисунке 4 изображен интерфейс IDE IntelliJ IDEA.

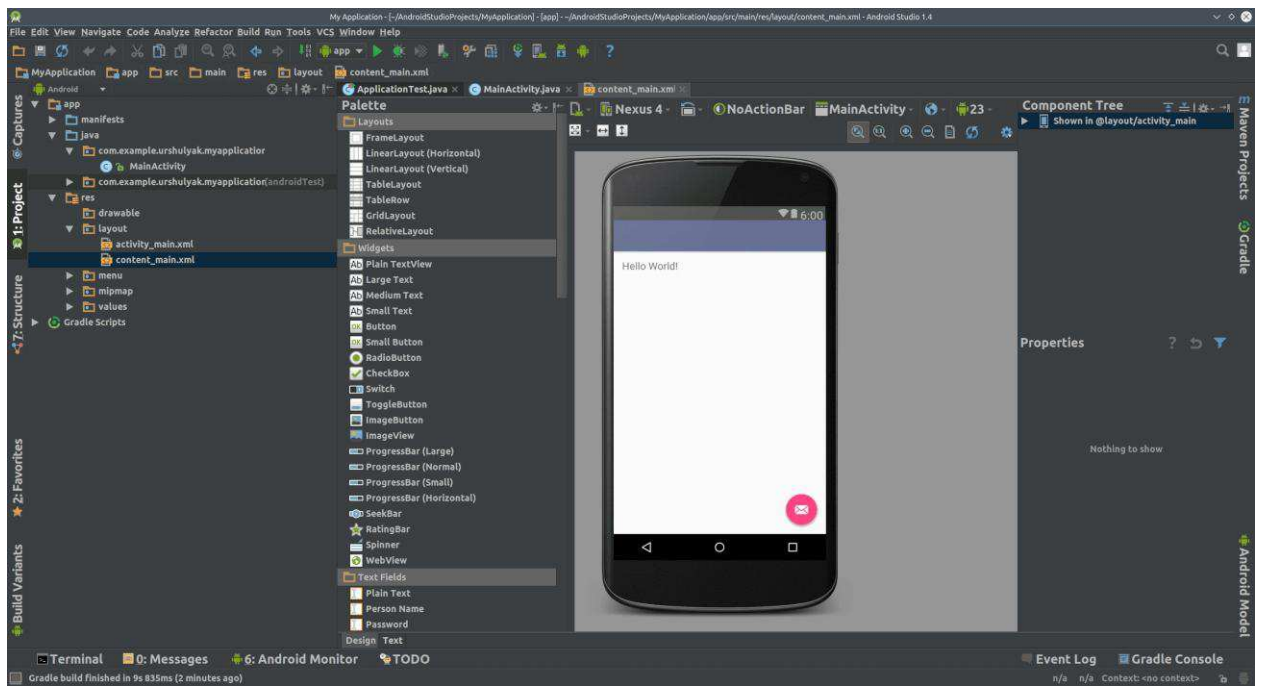


Рисунок 4 – Интерфейс программы IntelliJ Idea

Данную программу разработала компания JetBrains. Подобно Эклипс, эта среда позволяет создавать приложения и программы на нескольких языках программирования. Также IntelliJ IDEA не перегружает ПК. Достоинства IntelliJ IDEA:

- более оперативная отладка значений;
- предусмотрен автозаполнитель методов;
- есть рефакторинг (переработка кода);
- интерфейс более понятный и лаконичный;
- подходит для тех, кто программирует на Java.

Существует бесплатная редакция IDE IntelliJ Idea, на базе открытого исходного кода, – IntelliJ IDEA Community Edition. Данную редакцию можно использовать бесплатно и в личных целях и для коммерческой разработки.

Есть несколько причин, почему IntelliJ IDEA не подходит:

- Отсутствие современных плагинов. Хотя и Android Studio основан на IntelliJ IDEA, в ней отсутствуют необходимые плагины для мобильной разработки, к примеру такие плагины как Android APK Support.

Рассмотрим далее интегрированную среду разработки Android Studio, основанную на программном обеспечении IntelliJ IDEA.

Android Studio – это специализированный инструмент для разработки мобильных приложений для платформы Android. В дополнение к базовым функциям IDE Android Studio предоставляет функции, облегчающие взаимодействие с мобильными функциями, такими как push-уведомления или интеграция с камерами. Предприятия, использующие Android Studio, могут легко создавать и публиковать приложения для Android.

Android Studio создана для разработки приложений для Android, поэтому в ней есть отличные инструменты для этой платформы, но нет специальных функций для других платформ. Это делает Android Studio плохим выбором для компаний, которые хотят разрабатывать для других платформ. Android Studio все еще может быть хорошим дополнением к другой среде разработки для предприятий, которые, помимо прочего, разрабатывают приложения для Android. Интерфейс Android Studio изображен на рисунке 5.

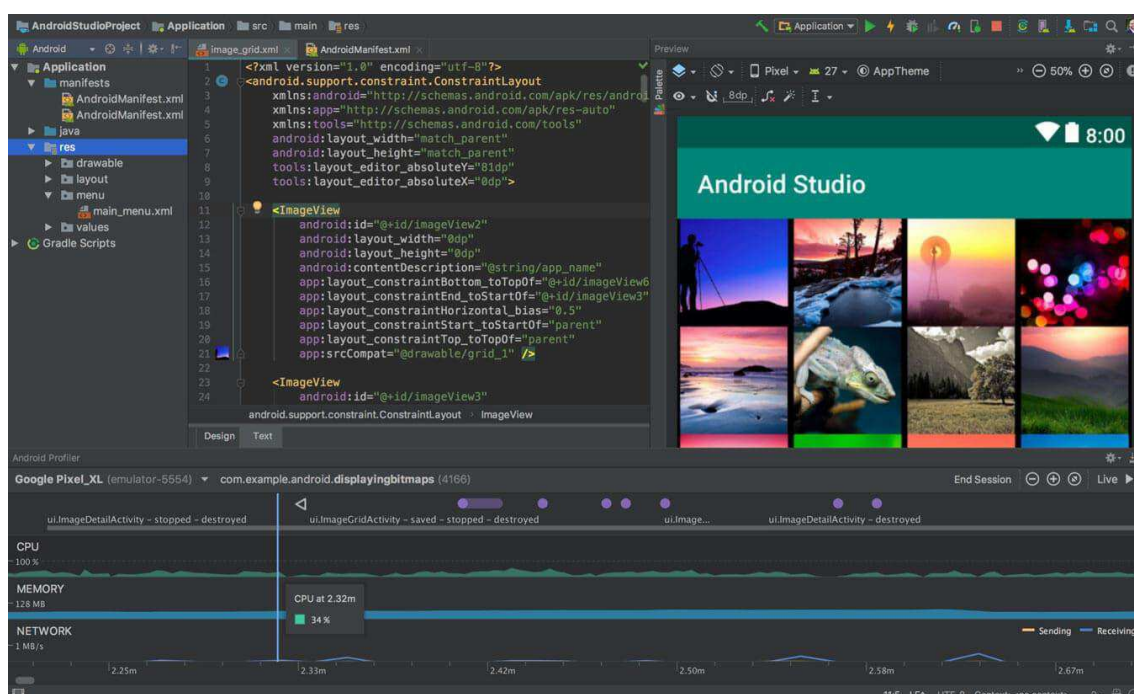


Рисунок 5 – Интерфейс программы Android Studio

В таблице 2 приведены параметры для сравнения средств Android разработки.

Таблица 2 – Сравнение инструментов разработки

<b>Особенность</b>	<b>Android Studio</b>	<b>IntelliJ IDEA</b>	<b>Eclipse</b>
Потребление ресурсов ПК	4	4	3
Кроссплатформенность	1	5	5
Популярность	5	3	3
Расширяемость	4	4	4
Обновления для Android разработки	5	2	2

Таким образом, для разработки мобильного приложения была выбрана среда разработки Android Studio. Для разработки мобильных приложений в данной среде разработки уже все присутствует, она бесплатна и активно поддерживается как Google, так и большим сообществом разработчиков. Остальные среды разработки ориентированы на другие языки и обновления для Android разработки выходят гораздо реже [1].

### **1.5.2 Выбор языка программирования**

Для разработки мобильных приложений на базе Android наиболее популярны такие языки как Java и Kotlin.

Java – язык программирования общего назначения. Относится к объектно-ориентированным языкам программирования, к языкам с сильной типизацией.

По данным компании Oracle, программы на Java запускаются на 3 млрд девайсов. Тем не менее Java широко используется и входит в число самых востребованных языков, это не вызывает сомнения.

Большинство крупных компаний так или иначе используют Java. Очень много серверных приложений для корпораций написаны на этом языке.

Язык программирования Java популярен благодаря нескольким важным достоинствам:

— Java – высокоуровневый язык, поэтому его легко использовать для чтения, а также для написания и обслуживания. Синтаксис похож на C++, но значительно проще.

— Программирование на Java соответствует стандартам, предъявляемым к корпоративным вычислительным системам. Благодаря множеству библиотек разработчики могут реализовать широкий функционал, необходимый организации.

— Объектно-ориентированная платформа дает возможность обозначить типы данных, их структурные особенности и функционал, который к ним применяется. Управляя структурами данных, можно создать связи между объектами. ООП дает возможность использовать объекты для других приложений повторно, не показывая данные, к которым не положен доступ для предотвращения ошибок, грамотно структурирует программы, облегчая процесс усовершенствования устаревших кодов.

— Функции Java устраняют основные уязвимости. Благодаря наличию Security Manager с политикой безопасности, позволяющей настроить правила доступа к базам, удастся избежать уязвимостей.

— Независимость от платформ позволяет писать код один раз, потом применяя его везде. Java обладает кроссплатформенными возможностями, позволяющими компилировать байт-коды, чтобы запустить приложение на любой другой платформе, работающей с Java Virtual Machine (JVM).

— Поддерживает распределенное программирование, обеспечивающее комфортную работу в удаленном режиме на нескольких компьютерах.

— Функция автоматического управления памятью способствует эффективной работе процессора, производительность которого не уменьшается под воздействием софта для очистки памяти.

— Платформа поддерживает многопоточность. При одновременном запуске потоков они используют одинаковые области памяти. Между потоками, функционирование которых не зависит друг от друга, можно оперативно переключаться, что особенно важно в анимационных приложениях.

При множестве плюсов язык программирования Java имеет некоторые минусы:

— Плата за коммерческое использование. Организации, использующие Java Standard Edition 8, при получении обновлений и устранении ошибок с 2019 года должны вносить оплату.

— Не предусмотрен нативный дизайн. Чтобы разрабатывать программы с нативным дизайном, на Android предусмотрен инструмент Android Studio. Но для разработки интерфейса пользователя на ПК необходимый инструментариум отсутствует, поэтому разбираясь, в чем недостатки Java, следует упомянуть несоответствия интерфейсов.

— Сложные коды. Многословные коды упрощают изучение языка. Но длинные и сложные предложения часто вызывают затруднения при чтении и просмотре кодов [2].

Kotlin – молодой объектно-ориентированный язык программирования, разработанный компанией JetBrains в 2011 г.

Главное преимущество Kotlin – функциональное программирование. То есть разработчик не задаёт последовательность команд, а описывает взаимодействие между командами и подпрограммами. Это значит, что программист задаёт правила, а код сам решает, как их использовать. Такой подход избавил Kotlin от основных недостатков C++ и Java – с ним работать проще, удобнее и быстрее.



Kotlin отличается лаконичным, понятным и удобным синтаксисом – это одинаково ценят новички и опытные разработчики. Большое количество кода в Kotlin программисты игнорируют — язык сам добавляет недостающие символы.

Чаще всего Kotlin применяют для android-разработки. Но он работает везде, где работает Java.

Преимущества программирования на Kotlin:

— Совместимость с Java. Kotlin используют в проектах, уже реализованных на Java.

— Чистота. В Kotlin нет лишних функций, избыточных модулей, он компактен и лаконичен. По сравнению с другими языками, код в Kotlin короче на 40 % – это снижает количество ошибок.

— Безопасность. На этапе компиляции предусмотрена автоматическая проверка кода. В Kotlin есть функция Null Safety, которая помогает избежать ошибок неопределённости в коде – самых распространённых и самых трудно выявляемых ошибок.

— Простота синтаксиса. Этот язык задумывался, в том числе, как лёгкий для изучения. Авторы уверяют, что перейти с любого другого языка на Kotlin можно за неделю.

Недостатки программирования на Kotlin:

— Низкая скорость компиляции. В среднем компиляция на Kotlin дольше, чем на Java на 30-40 %. Это связано с тем, что компиляция в виртуальной машине Java Virtual Machine предназначена для Java, поэтому с родным языком она работает быстрее.

— Мало дополнений. В силу молодости языка у него намного меньше библиотек и фреймворков, чем у Java. Но это вопрос времени, команда разработки Kotlin продолжает работать над продуктом и новые возможности языка не за горами.

— Малочисленное сообщество. Официальный релиз языка состоялся в 2017 году и пока разработчики при обнаружении багов вынуждены обращаться в техподдержку JetBrains, добавлять свою проблему в очередь и ждать её решения [3].

## **1.6 Модель взаимодействия пользователя с мобильным приложением «Детская безопасность»**

Диаграмма DFD наглядно отображает перемещение или движение потока данных в пределах процесса или системы. Для изображения входных и выходных данных, точек хранения информации и путей ее передвижения между источниками и пунктами доставки в таких диаграммах применяются стандартные фигуры, такие как прямоугольники и круги, а также стрелки и краткие текстовые метки. Диаграммы DFD варьируются от простейших набросков процессов (включая нарисованные вручную) до подробных многоуровневых схем с глубоким анализом способов обработки данных. Диаграммы DFD применяются для анализа существующих и моделирования новых систем. В лучших традициях визуализации данных диаграммы DFD часто наглядно «рассказывают» о процессах, которые сложно объяснить словами, и позволяют эффективно донести информацию и до «физиков», и до «лириков», то есть до всех участников организации — от разработчиков до генеральных директоров. Вот почему диаграммы DFD не утратили популярности за долгие годы существования [4].

На рисунке 6 изображена диаграмма DFD для мобильного приложения.

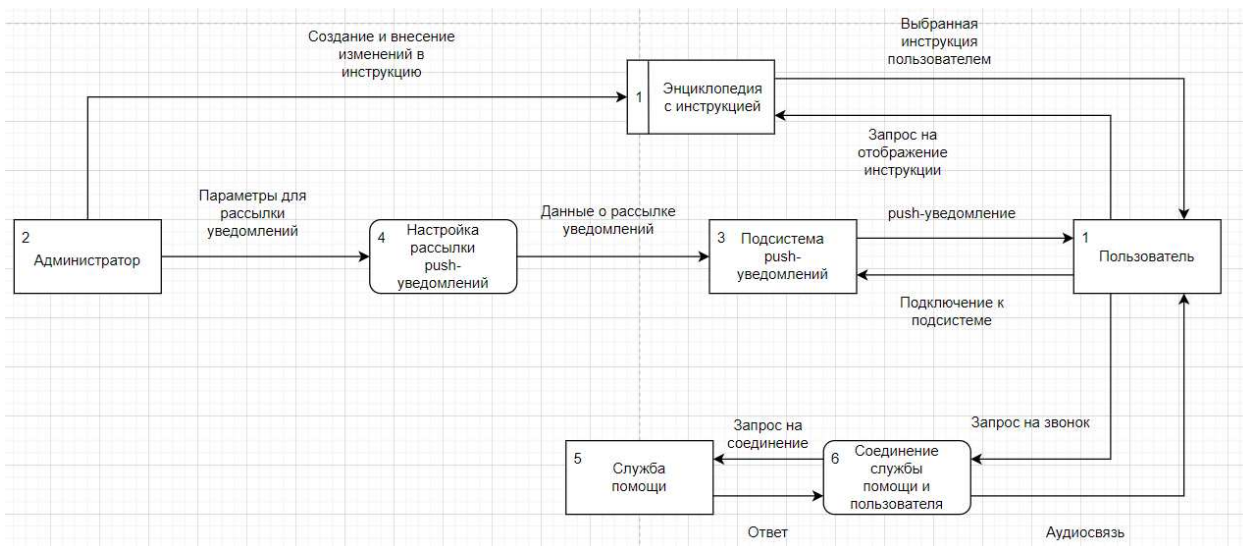


Рисунок 6 – Диаграмма DFD

В данной диаграмме отображается взаимодействие пользователя с мобильным приложением. Разберем DFD диаграмму по частям.

В DFD диаграмме присутствуют следующие сущности:

а) Пользователь.

Пользователь будет выбирать инструкцию в приложении и получит выбранную инструкцию на дисплее в виде текста или кнопок. Пользователь сможет получать push-уведомления на смартфон, для этого ему надо подключиться к подсистеме. Подключение к подсистеме автоматическое. Пользователь сможет отправлять запросы на звонки, после чего будет происходить попытка соединения со специальными службами.

б) Администратор.

Администратор необходим для управления энциклопедией и подсистемой push-уведомлений. Администратор сможет добавлять, менять или удалять данные из накопителя данных «Энциклопедия с инструкцией». Четвертый блок отвечает за управление рассылкой push-уведомлений. Администратор обязательно должен настроить заголовок и текст уведомления, на какое приложение будет отправлено и время отправки.

в) Служба помощи.

В приложении будут представлены телефоны разных служб, и пользователь сможет получить к ним быстрый доступ с помощью интерфейса приложения.

### 1.7 Выбор архитектуры мобильного приложения

Проанализировав предметную область можно предположить, что для данной системы больше всего подходит архитектура клиент-сервер. Модель данной архитектуры приведена на рисунке 7.

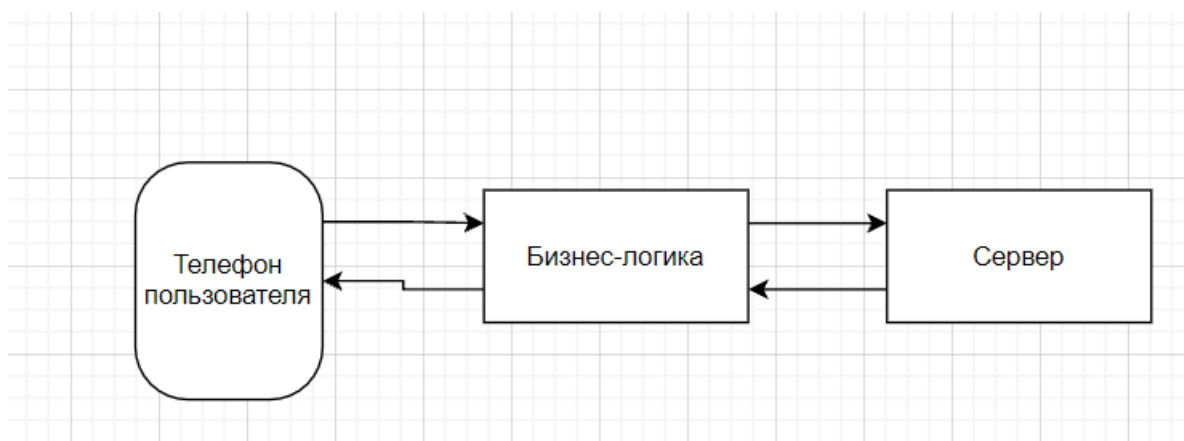


Рисунок 7 – Архитектура системы

В данную архитектуру входит:

- Телефон пользователя.
- Бизнес-логика.
- Сервер.

Элемент «Телефон пользователя» будет являться клиентом.

Элемент «Бизнес-логика» представляет собой программную часть мобильного приложения «Детская безопасность».

Элемент «Сервер» будет отвечать за настройку рассылки и отправку push-уведомлений. Разберем подробнее данные элементы:

а) Телефон пользователя. Данный элемент будет содержать часть слоя View, отвечающего за интерфейс мобильного приложения. Слой view не является частью бизнес-логики, потому что он отвечает за взаимодействие пользователя и приложения.

б) Бизнес-логика. Данный элемент будет полностью отвечать за логику мобильного приложения или программную часть. Основные функции данного элемента:

1) Звонки в определенные службы. На интерфейсе пользователя будут расположены кнопки, которые обеспечат быстрый звонок в необходимые службы.

2) Отображение выбранной инструкции. На интерфейсе будут расположены кнопки для выбора инструкции пользователем.

в) Сервер. Данный элемент отвечает за систему настройки рассылки и отправки push-уведомлений. Основные функции данной системы:

1) Настройка рассылки push-уведомлений. Данная система позволит настроить текст рассылки и периодичность рассылки уведомлений.

2) Рассылка push-уведомлений. Push-уведомление необходимо как напоминание или предупреждение о чем-либо, к примеру, предупреждение о сезоне паводка и необходимости соблюдать меры предосторожности рядом с водой.

## **1.8 Вывод по разделу «Анализ предметной области»**

В данном разделе была обоснована актуальность разработки мобильного приложения «Детская безопасность в Хакасии», а также проведено обследование и описание структуры ЦУР РХ, задач и функций, выполняемых ЦУР РХ. Были проанализированы различные средства разработки, включающие в себя среду разработки и язык программирования. В результате анализа и сравнения сред разработки, была выбрана Android Studio. В результате анализа и сравнения языков программирования был выбран язык

Java. Была построена и представлена DFD модель для более наглядной демонстрации работы приложения с пользователем. Была выбрана и описана архитектура «клиент-сервер».

## 2 Разработка мобильного приложения «Детская безопасность»

Выбрав среду разработки и язык программирования, перейдем к разработке мобильного приложения и начнем с установки необходимых программных средств.

### 2.1 Создание и настройка проекта в Android Studio

После установки Android Studio появится окно, изображенное на рисунке 8.

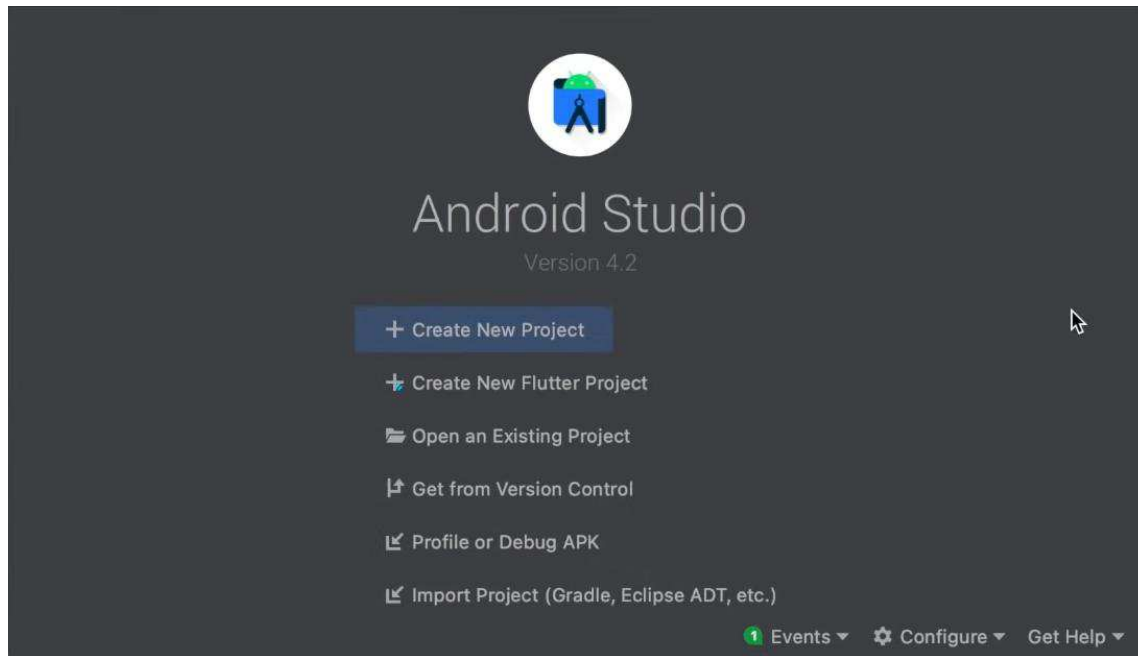


Рисунок 8 – Окно создания проекта

Для начала настроим SDK. Нажимаем на Configure (внизу справа на рисунке 8) и далее SDK Manager. В появившемся окне выбираем необходимые SDK. Выбранные SDK изображены на рисунке 9.



<input type="checkbox"/> Android SDK Build-Tools 31-rc3		Update Available: 31.0.0 rc3
<input checked="" type="checkbox"/> NDK (Side by side)		Installed
<input type="checkbox"/> Android SDK Command-line Tools (latest)		Not Installed
<input type="checkbox"/> CMake		Not Installed
<input type="checkbox"/> Android Auto API Simulators	1	Not installed
<input type="checkbox"/> Android Auto Desktop Head Unit Emulator	1.1	Not installed
<input checked="" type="checkbox"/> Android Emulator	30.6.5	Installed
<input checked="" type="checkbox"/> Android SDK Platform-Tools	31.0.2	Installed
<input checked="" type="checkbox"/> Android SDK Tools	26.1.1	Installed
<input type="checkbox"/> Google Play APK Expansion library	1	Not installed
<input type="checkbox"/> Google Play Instant Development SDK	1.9.0	Not installed
<input type="checkbox"/> Google Play Licensing Library	1	Not installed
<input type="checkbox"/> Google Play services	49	Not installed
<input type="checkbox"/> Google Web Driver	2	Not installed
<input checked="" type="checkbox"/> Intel x86 Emulator Accelerator (HAXM installer)	7.6.5	Installed
<input type="checkbox"/> Layout Inspector image server for API 29-30	6	Not installed
<input type="checkbox"/> Layout Inspector image server for API S	2	Not installed

Рисунок 9 – Лист SDK

Теперь создадим устройство для тестирования приложения. Нажимаем Configure и далее AVD Manager, в открывшемся окне нажимаем на Create virtual device. AVD расшифровывается как Android Virtual Device и позволяет создавать устройства разных моделей. Выберем модель смартфона Pixel 2 и установим API 28. Далее вводим название устройства и нажимаем create. Виртуальное устройство готово.

Нажимаем ОК и Create New Project. После нажатия кнопки появится окно с выбором готовых интерфейсов приложения. Слева выбираем Phone and Tablet. Готовые интерфейсы не подходят, поэтому выберем Empty Activity и нажимаем Next. Окно с интерфейсами изображено на рисунке 10.

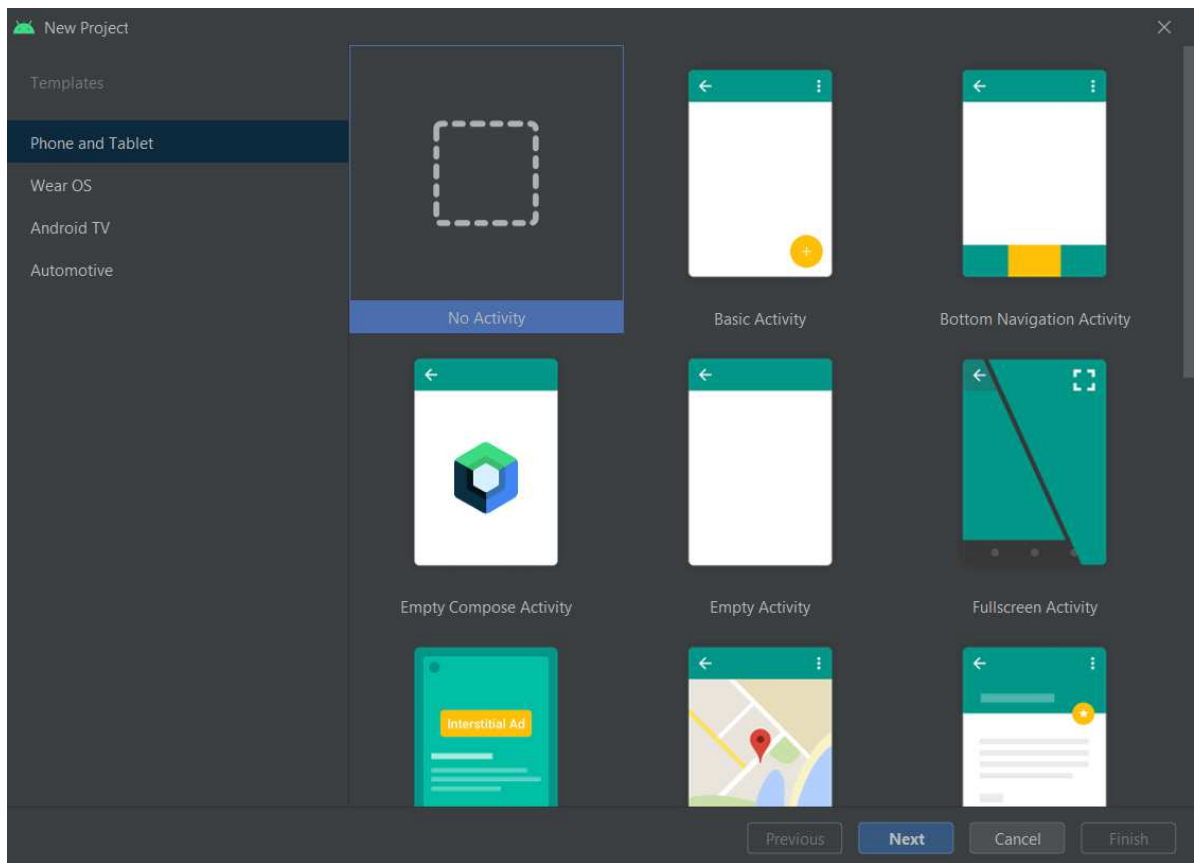


Рисунок 10 – Окно выбора Activity

Следующее окно настройки включает в себя: введение названия проекта, название пакета, место сохранения, язык программирования, минимальная версия SDK. Название проекта и пакета не имеет значения, название проекта вводим любое или оставляем стандартное. Название пакета не меняем. Язык программирования выбираем Java. Минимальная версия SDK влияет на функционал и на более старых версиях часть функций может не работать. Для разработки данного приложения понадобится 9 версия. Нажимаем Finish и создается проект. Проект готов и появилось окно с пустым интерфейсом приложения. Окно изображено на рисунке 11.



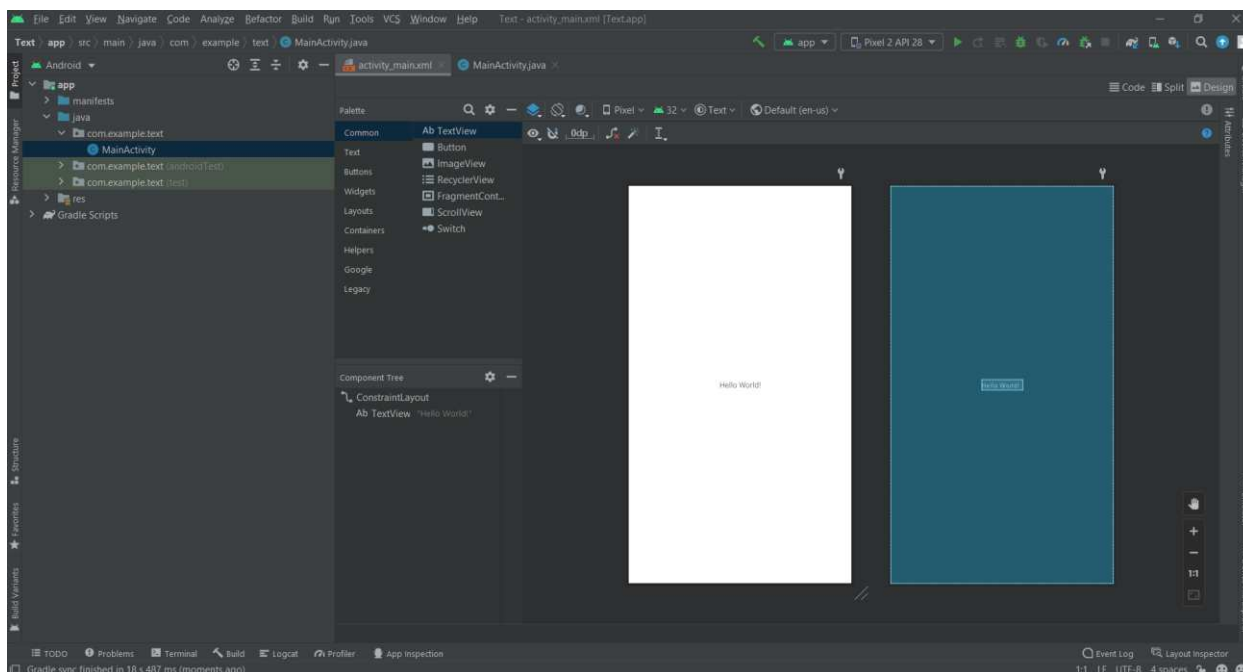


Рисунок 11 – Главное окно приложения

## 2.2 Подключение мобильного приложения к Firebase

Для подключения Firebase используется 2 способа: автоматический и ручной. Опишем ручной способ подключения.

Первым делом стоит уточнить, что Firebase поддерживает API от 19 версии и выше. Так что при создании проекта с API версией ниже, подключить Firebase не получится. Первым делом необходимо настроить Gradle. Gradle – это инструмент для автоматической сборки приложения. Нажмем на кнопку справа, выделенную красным на рисунке 12 и откроется окно, изображенное на рисунке 13.

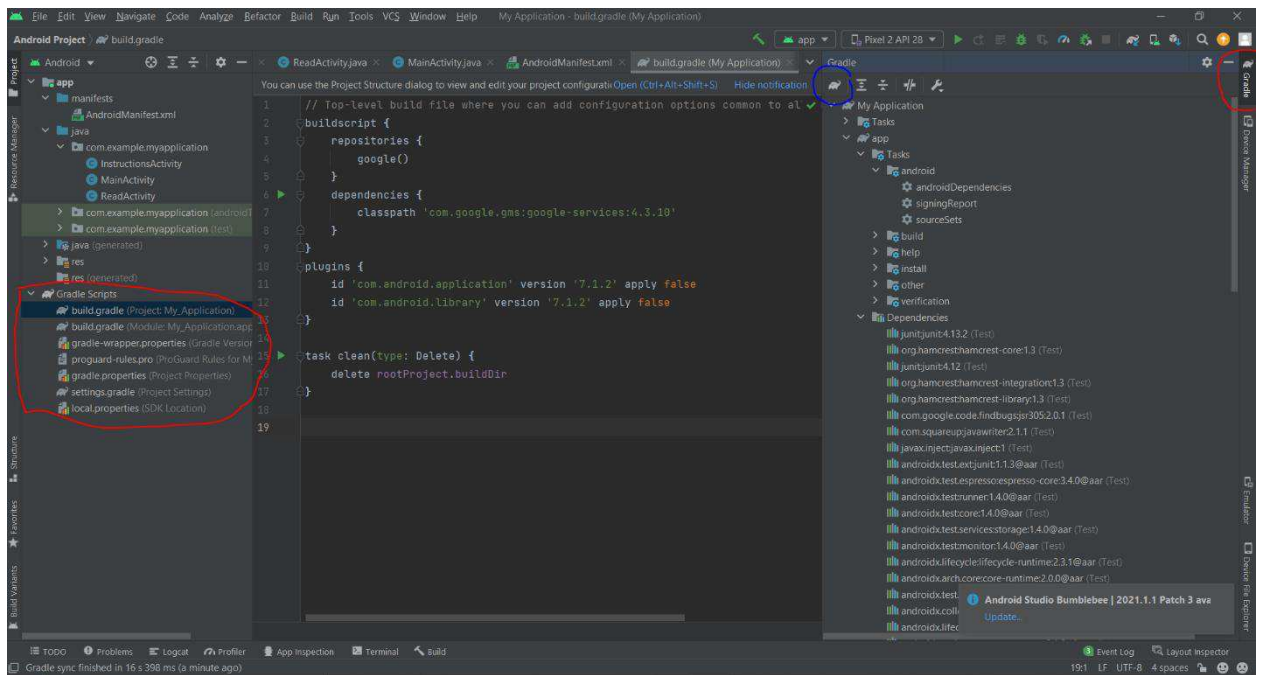


Рисунок 12 – Gradle и его файлы

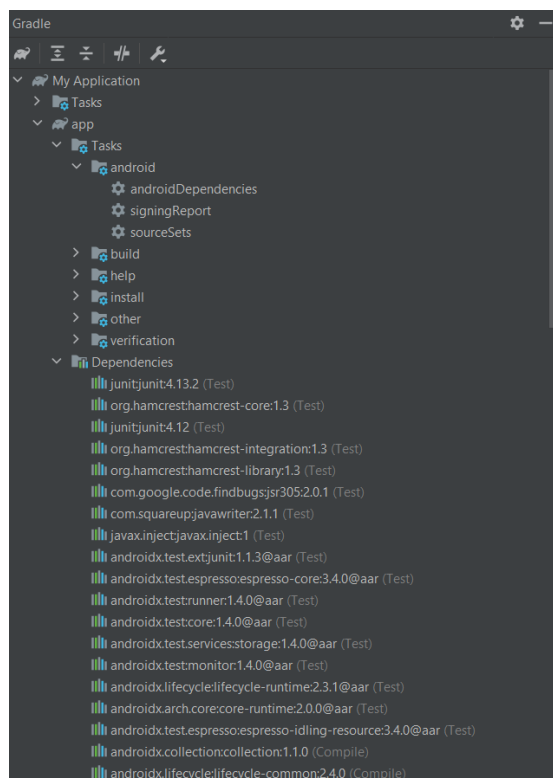


Рисунок 13 – Gragle

Теперь нажмем на кнопку, выделенную синим на рисунке 12, и ждем первичной настройки Gradle. После ожидания создаются файлы, выделенные

красным слева на рисунке 12. Открываем build.gradle проекта и вставляем код, показанный на рисунке 14.

```
buildscript {
    repositories {
        google()
    }
    dependencies {
        classpath 'com.google.gms:google-services:4.3.10'
    }
}

plugins {
    id 'com.android.application' version '7.1.2' apply false
    id 'com.android.library' version '7.1.2' apply false
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

Рисунок 14 – Build Gradle Project

Открываем Build Gradle Module и вставляем код в метод plugins: id 'com.google.gms.google-services'. Вставим код, изображенный на рисунке 15, в метод dependencies.

```
implementation platform('com.google.firebase:firebase-bom:29.2.1')
implementation 'com.google.firebase:firebase-analytics'
implementation 'com.google.firebase:firebase-database:20.0.4'
```

Рисунок 15 – Build Gradle Module

Подключение завершено, проверим подключение. На панели инструментов открываем Tools>Firebase>Realtime Database. Если написано то

же, что и на рисунке 16, подключение выполнено верно. В этом окне также можно подключить Firebase автоматически.

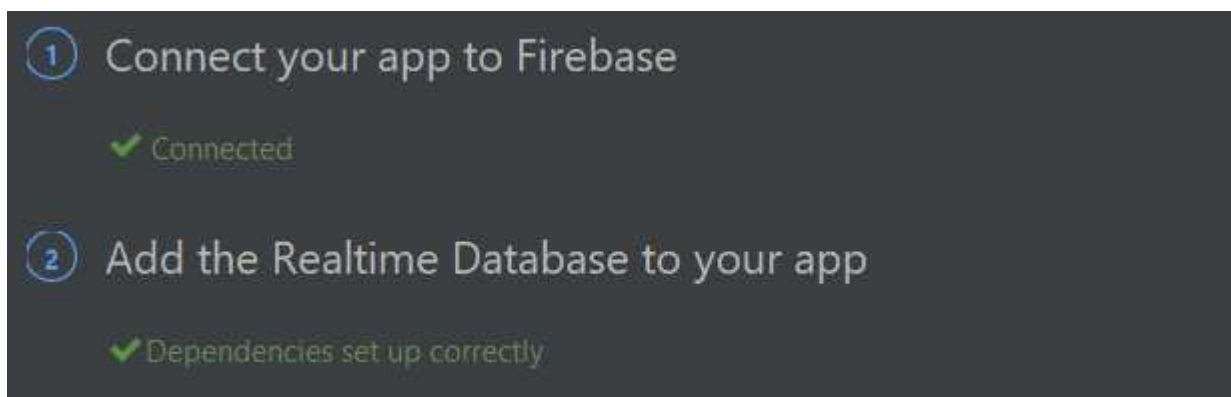


Рисунок 16 – Firebase подключен

Далее создадим проект в Firebase. Открываем сайт <https://firebase.google.com>. Сайт изображен на рисунке 17, справа нажмем Go to console (выделено красным).

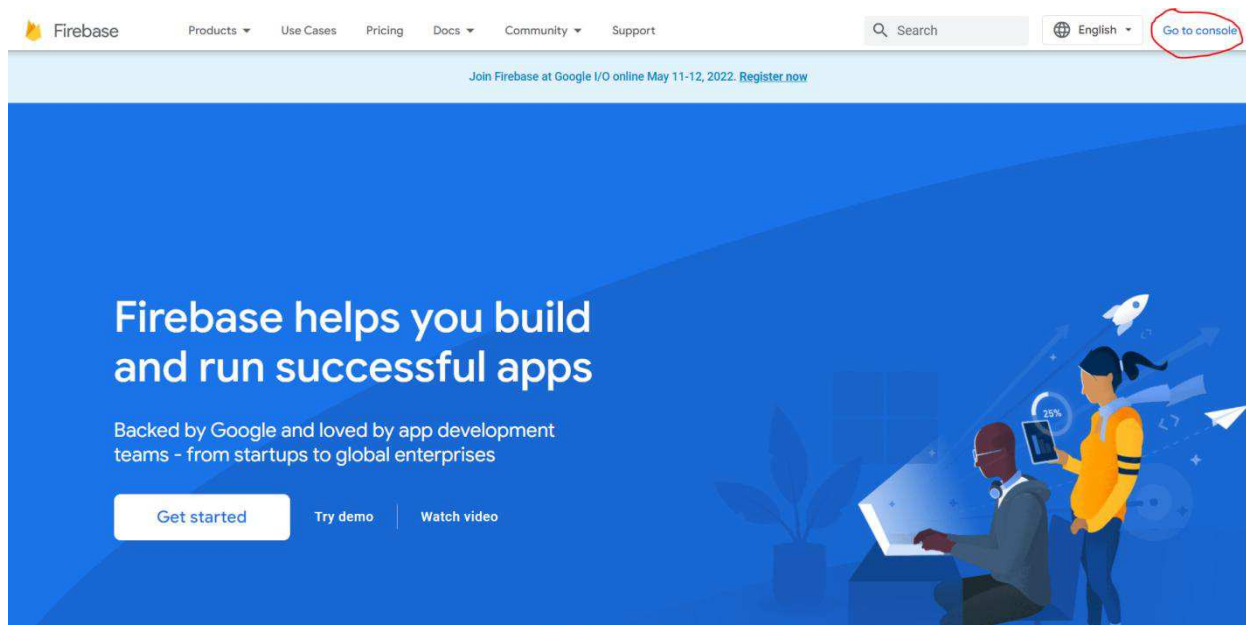


Рисунок 17 – Сайт Firebase

Нажимаем на кнопку Add project, заполняем все данные и создаем проект. После ожидания открывается консоль. Подключаем ее к приложению, нажимая на кнопку с логотипом Android, изображенную на рисунке 18.

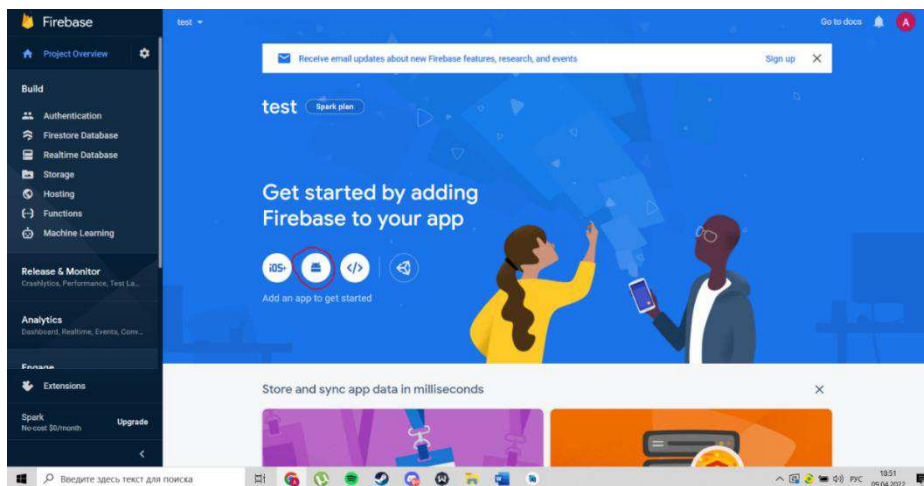


Рисунок 18 – Консоль Firebase

Далее настраиваем все по инструкции. При создании проекта необходимо вставить сертификат приложения SHA-1. Найти его можно, открыв Gradle и запустив signingReport. Местоположение изображено на рисунке 19.

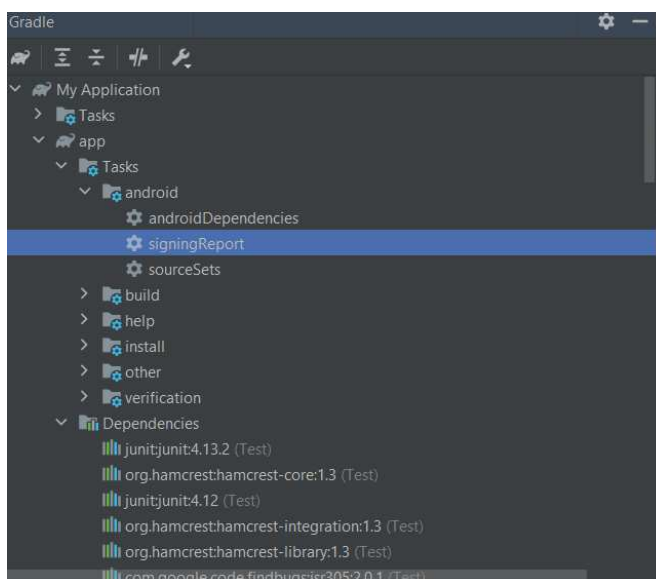


Рисунок 19 – SigningReport

Дважды кликнув на него дважды, в консоли ниже покажут сертификаты приложения. Из соображения безопасности часть сертификата будет закрашена на рисунке 20.

```
MD5: 26: [REDACTED] 1:39:F7:4B
SHA1: 45: [REDACTED] :22:D5:A5:35:A8:32:B3:DE
SHA-256: 8D:11:D1:D0:14:0D: [REDACTED] 92:2E:12:19:98:47:96:AC:DD:46:EC:D2:A3
```

Рисунок 20 – Сертификат приложения

В результате выполнения всех этапов, Firebase подключен к приложению.

### 2.3 Разработка кнопок мобильного приложения

Создание кнопок будет динамическим. Вставим код, изображенный на рисунке 21, в *MainActivity* [6, 7].

```
92 // Берем информацию из базы данных
93 db.collection( collectionPath: "Buttons").get();
94 db.collectionGroup( collectionId: "ButtonPhone").get();
95 db.collectionGroup( collectionId: "ButtonText").get();
96 db.collection( collectionPath: "Instructions").get();
97 db.collection( collectionPath: "Buttons")
98 .get(source)
99 .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
100     @Override
101     public void onComplete(@NonNull Task<QuerySnapshot> task) {
102         if (task.isSuccessful()) {
103
104             // Цикл для прослушивания всех документов в коллекции
105             for (QueryDocumentSnapshot document : task.getResult()) {
106                 // Создание кнопок
107                 Button button = new Button( context: MainActivity.this);
108                 String name = document.getString( field: "name");
109                 button.setText(name);
110                 button.setTextColor(Color.BLACK);
111                 button.setBackground(shape);
112                 button.setTextSize(tvtextsize);
113                 layout.addView(button, params);
114                 //
115
116                 // Настройка кликабельности
117                 button.setOnClickListener(new View.OnClickListener() {
118                     @Override
119                     public void onClick(View v) {
120                         String phone = document.getString( field: "phone");
121                         AlertDialog(phone);
122                     }
123                 });
124             }
125         }
126     }
127 }
```

Рисунок 21 – Код создания кнопок

Результат данного кода изображен на рисунке 22.

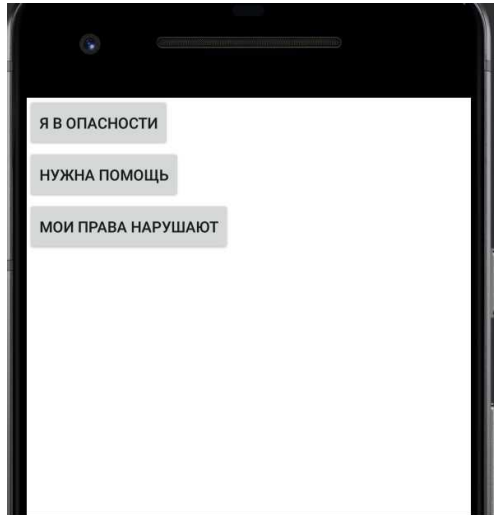


Рисунок 22 – Стартовое окно приложения без добавления графических параметров кнопок

Данный код просматривает коллекцию в firestore с помощью цикла и, основываясь на документах в коллекции, создает кнопки [8]. Документы в Firestore изображены на рисунке 23.

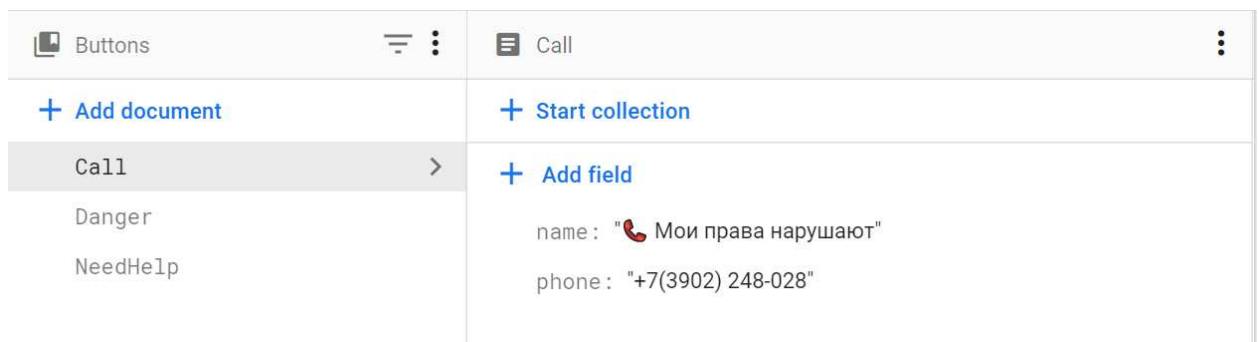


Рисунок 23 – Документы Firestore, лист 1

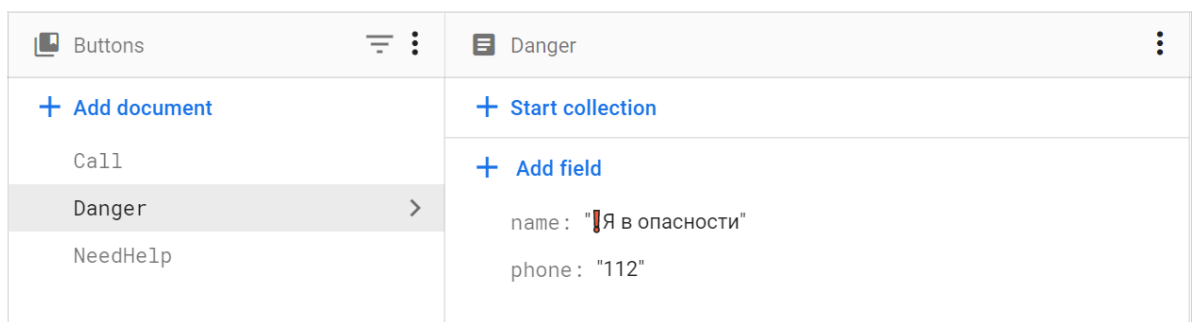


Рисунок 23, лист 2

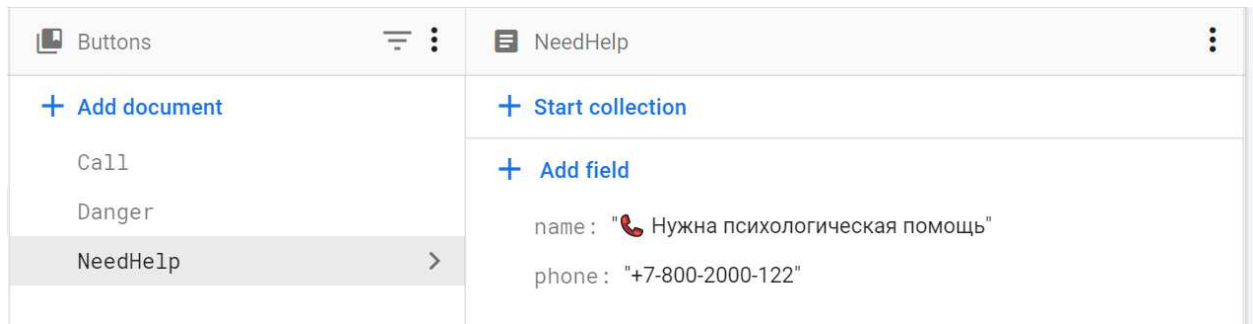


Рисунок 23, лист 3

Текст, записанный в полях Name, программа вставляет в названия кнопок, а в поле phone будет записан номер телефона. Таким образом были созданы динамические кнопки. Соответственно, добавив новый документ с аналогичными полями, будет создана новая кнопка сразу в приложении. Создадим кнопку, которая будет присутствовать всегда. Кнопка будет открывать InstructionsActivity и содержать меню выбора инструкции. Вставим код, изображенный на рисунке 24.

```

127 // Создание кнопки с инструкцией
128 Button button = new Button(context: MainActivity.this);
129 button.setText("Что делать, если...\uD83D\uDCDA");
130 button.setTextColor(Color.BLACK);
131 button.setLayoutParams(new RelativeLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT, |
132     ViewGroup.LayoutParams.WRAP_CONTENT));
133 layout.addView(button, params);
134 button.setTextSize(tvtextsize);
135 button.setBackground(shape);
136
137 // Настройка кликабельности
138 button.setOnClickListener(new View.OnClickListener() {
139     @Override
140     public void onClick(View v) {
141         Intent intent = new Intent(packageContext: MainActivity.this,
142             InstructionsActivity.class);
143         startActivity(intent);
144     }
145 });
146 progressBar.setVisibility(View.INVISIBLE); // Скрытие индикатора прогресса
147 // после завершения чтения данных и создания кнопок
148 }
149 };
```

Рисунок 24 – Код реализации кнопки «Что делать если...»



Установим графические параметры создания кнопок. Вставим код, изображенный на рисунке 25. Данный код создает кнопки с определенными параметрами.

```
64 // Создание и установка параметров Layout
65 layout = findViewById(R.id.linearlayoutid);
66 LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
67     ViewGroup.LayoutParams.WRAP_CONTENT);
68 params.gravity = Gravity.CENTER_HORIZONTAL;
69 params.bottomMargin = 20;
70 params.width = 450;
71 params.height = 175;
72 params.setMargins(left: 0, top: 30, right: 0, bottom: 0);
73 int tvtextsize = 14;
74 //
```

Рисунок 25 – Параметры кнопок

Код на рисунке 26 позволит сделать кнопки закругленными и указать цвет фона кнопок.

```
//
GradientDrawable shape = new GradientDrawable();
shape.setCornerRadius(70);
shape.setColor(Color.argb(alpha: 70, red: 0, green: 153, blue: 153));
```

Рисунок 26 – Код для определения графических параметров кнопки

На рисунке 27 изображен стартовый экран после внесенных изменений выше.



Рисунок 27 – Обновленный стартовый экран

В таком виде будет недостаток – при открытии приложения будет пустой экран, пока все данные не загрузятся. Для решения данной проблемы создадим индикатор загрузки, который будет скрываться при полной загрузке. Код на рисунке 28 добавит индикатор загрузки.

```
81 // Индикатор прогресса
82 RelativeLayout relativeLayout = findViewById(R.id.relativeLayout);
83 RelativeLayout.LayoutParams paramBar = new RelativeLayout.LayoutParams( w, 200, h, 200);
84 paramBar.addRule(RelativeLayout.CENTER_IN_PARENT);
85 ProgressBar progressBar = new ProgressBar( context: MainActivity.this, attrs: null, android.R.attr.progressBarStyleLarge);
86 progressBar.setIndeterminate(true);
87 relativeLayout.addView(progressBar, paramBar);
88 progressBar.setVisibility(View.VISIBLE);
89 //
```

Рисунок 28 – Код для создания индикатора загрузки

Для индикатора загрузки создадим Relative Layout, т.к. Linear Layout позиционирует объекты по порядку, соответственно из-за объекта ProgressBar остальные кнопки будут съезжать вниз. Relative Layout указывает явное и относительное положение объекта, благодаря чему можно установить индикатор загрузки поверх других объектов. Индикатор загрузки изображен на рисунке 29.



Рисунок 29 – Индикатор загрузки приложения

На строке 88 программного кода, показанного на рисунке 28, ставим индикатор видимым, как только выполняется код, изображенный на рисунке 21, индикатор ставим невидимым (строка 146, рисунок 22).

Теперь перейдем к созданию диалогового окна. Метод, представленный на рисунке 30, создает диалоговое окно при нажатии кнопки вызова с положительными и отрицательными ответами.

```

152 // Метод создания диалогового окна
153 public void AlertDialog(String phone) {
154     AlertDialog.Builder builder = new AlertDialog.Builder( context: MainActivity.this);
155     builder.setTitle("Предупреждение");
156     builder.setMessage("Требуется подтверждение вызова. " +
157         "При нажатии «Да» ваш номер телефона и местоположение будут " +
158         "автоматически направлены в Центр обработки вызовов");
159     builder.setPositiveButton( text: "Да", new DialogInterface.OnClickListener() {
160         @Override
161         public void onClick(DialogInterface dialogInterface, int i) {
162             if (phone.trim().length() > 0) {
163                 if (ContextCompat.checkSelfPermission( context: MainActivity.this,
164                     Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
165                     ActivityCompat.requestPermissions( activity: MainActivity.this,
166                         new String[]{Manifest.permission.CALL_PHONE}, REQUEST_CALL);
167                 } else {
168                     String dial = "tel:" + phone;
169                     startActivity(new Intent(Intent.ACTION_CALL, Uri.parse(dial)));
170                 }
171             }
172         }
173     });
174     builder.setNegativeButton( text: "Нет", new DialogInterface.OnClickListener() {
175         @Override
176         public void onClick(DialogInterface dialogInterface, int i) {
177             dialogInterface.dismiss();
178         }
179     });
180     AlertDialog dialog = builder.create();
181     dialog.show();

```

Рисунок 30 – Метод AlertDialog

Создаем метод, сначала устанавливаем заголовок и текст диалогового окна. Следом устанавливаем кнопку с положительным ответом, при нажатии которой будет проводиться проверка на разрешение управлением звонками с последующим звонком. При нажатии на отрицательный ответ, диалоговое окно будет закрываться [9].

Данный метод вызывается на 121 строке кода, представленного на рисунке 21.

В методе происходит проверка наличия разрешения на управление звонками.

Диалоговое окно, полученное после нажатия кнопки вызова, изображено на рисунке 31.



Рисунок 31 – Диалоговое окно, полученное при нажатии кнопки вызова

Создадим меню инструкций. Создадим новый пустой Activity и вставим код, изображенный на рисунке 32.

```

81 db.collection(collectionPath: "Instructions").get(source).addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
82     @Override
83     public void onComplete(@NonNull Task<QuerySnapshot> task) {
84         if (task.isSuccessful()) {
85             for (QueryDocumentSnapshot document : task.getResult()) {
86                 String id = document.getId();
87                 Button button = new Button(context: InstructionsActivity.this);
88                 String name = document.getString(field: "name");
89                 button.setText(name);
90                 button.setTextSize(14);
91                 button.setTextColor(Color.BLACK);
92                 button.setBackground(shape);
93                 layoutmain.addView(button, params);
94                 button.setOnClickListener(new View.OnClickListener() {
95                     @Override
96                     public void onClick(View v) {
97                         Intent intent = new Intent(packageContext: InstructionsActivity.this, MenuActivity.class);
98                         intent.putExtra(name: "id", id);
99                         startActivity(intent);
100                     }
101                 });
102             }
103         }

```

Рисунок 32 – Код для создания меню с инструкцией

В данном коде в цикле на каждой итерации возвращается id документа и при нажатии на кнопку будет возвращен конкретный id конкретного документа и отправлен в другой Activity. Добавим документы с инструкцией в Firestore. Пример документа изображен на рисунке 33.

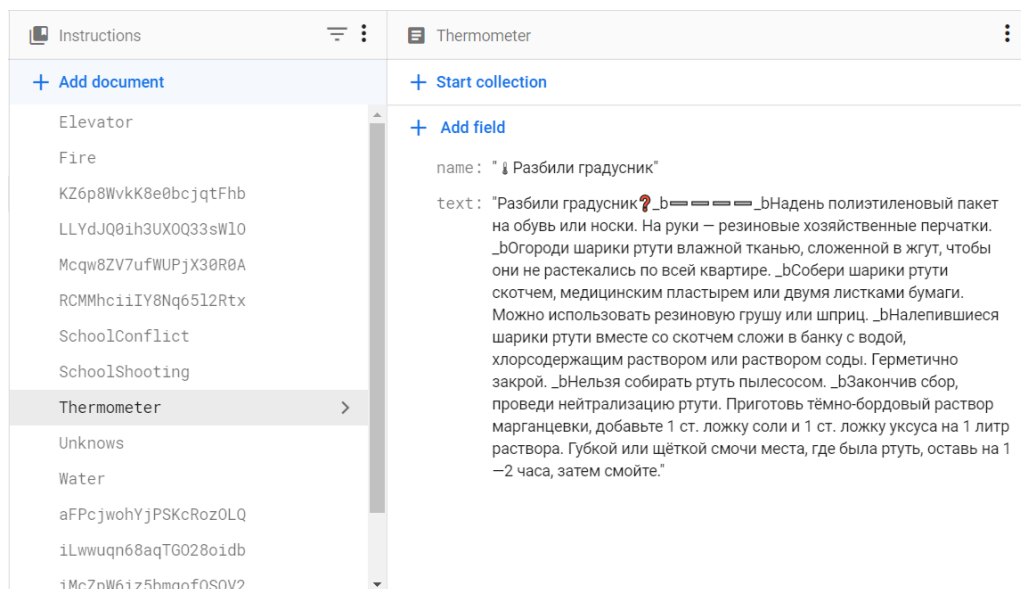


Рисунок 33 – Пример документа в Firestore

Код на рисунке 34 создает кнопку для перехода на сайт «Телефон доверия».

```

104 Button button = new Button( context: InstructionsActivity.this);
105 button.setText("Еще больше информации");
106 button.setTextSize(14);
107 button.setTextColor(Color.BLACK);
108 button.setBackground(shape);
109 button.setPadding( left: 0, top: 20, right: 0, bottom: 0);
110 layoutmain.addView(button, params);
111 button.setOnClickListener(new View.OnClickListener() {
112     @Override
113     public void onClick(View v) {
114         Intent browserIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("https://telefon-doveria.ru/"));
115         startActivity(browserIntent);
116     }
117 });
118 progressBar.setVisibility(View.INVISIBLE);
119
120
121 }
122 });

```

Рисунок 34 – Код для перехода на сайт «Телефон доверия»

В коде на рисунке 34 создается кнопка и присваивается обработчик событий для перехода по указанной ссылке.

Так как кнопок получится слишком много, необходимо настроить Activity для возможности скроллинга. Для этого необходимо создать XML файл и вставить код, изображенный на рисунке 35. Данный код содержит правила для Layout. Необходимо вставить ScrollView и указать размеры. Обязательно стоит учесть, что ScrollView может содержать только один дочерний Layout. В данном случае вставим LinearLayout для автоматического вертикального упорядочивания кнопок. В LinearLayout указываем его ID и вертикальную ориентацию.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/relativeinstructions"
    android:background="@color/white"
    android:layout_gravity="center">
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:tools="http://schemas.android.com/tools"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/linearLayoutinstructions"

            tools:context=".InstructionsActivity"
            android:orientation="vertical">

            </LinearLayout>
        </ScrollView>
    </RelativeLayout>

```

Рисунок 35 – XML файл для меню инструкций

Завершенное меню выбора инструкции изображено на рисунке 35.



Рисунок 36 – Меню выбора инструкции, лист 1





Рисунок 36, лист 2

Теперь необходимо написать код для отображения инструкции. Создадим *MenuActivity* и вставим код, изображенный на рисунке 37.

```

62 db
63 .collection( collectionPath: "Instructions") CollectionReference
64 .document(id) DocumentReference
65 .get(source) Task<DocumentSnapshot>
66 .addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {
67     @Override
68     public void onComplete(@NonNull Task<DocumentSnapshot> task) {
69         DocumentSnapshot document = task.getResult();
70         String text = document.getString( field: "text").replace( target: "_b", replacement: "\n");
71         tv.setText(text);
72         tv.setTextColor(Color.rgb( red: 0, green: 0, blue: 0));
73         tv.setPadding( left: 30, top: 50, right: 30, bottom: 50);
74         tv.setTextSize(18);
75     }

```

Рисунок 37 – MenuActivity, лист 1

```
77 db
78 .collection( collectionPath: "Instructions") CollectionReference
79 .document(id) DocumentReference
80 .collection( collectionPath: "ButtonPhone") CollectionReference |
81 .get(source) Task<QuerySnapshot>
82 .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
83     @Override
84     public void onComplete(@NonNull Task<QuerySnapshot> task) {
85         if (task.isSuccessful()) {
86             for (QueryDocumentSnapshot document : task.getResult()) {
87                 String name = document.getString( field: "name");
88                 Button button = new Button( context: MenuActivity.this);
89                 button.setText(name);
90                 button.setTextSize(14);
91                 button.setTextColor(Color.BLACK);
92                 button.setBackground(shape);
93                 layout.addView(button, params);
94                 button.setOnClickListener(new View.OnClickListener() {
95                     @Override
96                     public void onClick(View v) {
97                         String phone = document.getString( field: "phone");
98                         AlertDialog(phone);
99                     }
100                 });
101             }
102         }
103     }
104 });
```

Рисунок 37, лист 2

```
105 db
106 .collection( collectionPath: "Instructions") CollectionReference
107 .document(id) DocumentReference
108 .collection( collectionPath: "ButtonText") CollectionReference
109 .get(source) Task<QuerySnapshot>
110 .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
111     @Override
112     public void onComplete(@NonNull Task<QuerySnapshot> task) {
113         if (task.isSuccessful()) {
114             for (QueryDocumentSnapshot document : task.getResult()) {
115                 String name = document.getString( field: "name");
116                 Button button = new Button( context: MenuActivity.this);
117                 button.setText(name);
118                 button.setTextSize(14);
119                 button.setTextColor(Color.BLACK);
120                 button.setBackground(shape);
121                 layout.addView(button, params);
122                 button.setOnClickListener(new View.OnClickListener() {
123                     @Override
124                     public void onClick(View v) {
125                         String text = document.getString( field: "text").replace( target: "_b", replacement: "\n");
126                         setContentView(R.layout.read);
127                         TextView tv = (TextView) findViewById(R.id.textViewInstructions);
128                         tv.setText(text);
129                         tv.setTextColor(Color.rgb( red: 0, green: 0, blue: 0));
130                         tv.setPadding( left: 30, top: 50, right: 30, bottom: 50);
131                         tv.setTextSize(18);
132                     }
133                 });
134             }
135         }
136     }
137 });
```

Рисунок 37, лист 3

Код на листе 1 рисунка 37 получает текст из документа, ID которого передан на рисунке 32 и устанавливает его в имеющийся TextView. Так как в Firebase не предусмотрен перенос текста, заменим символы «\_b» на символ новой строки «\n» (строка 70). Пример данного типа инструкции изображен на рисунке 38.

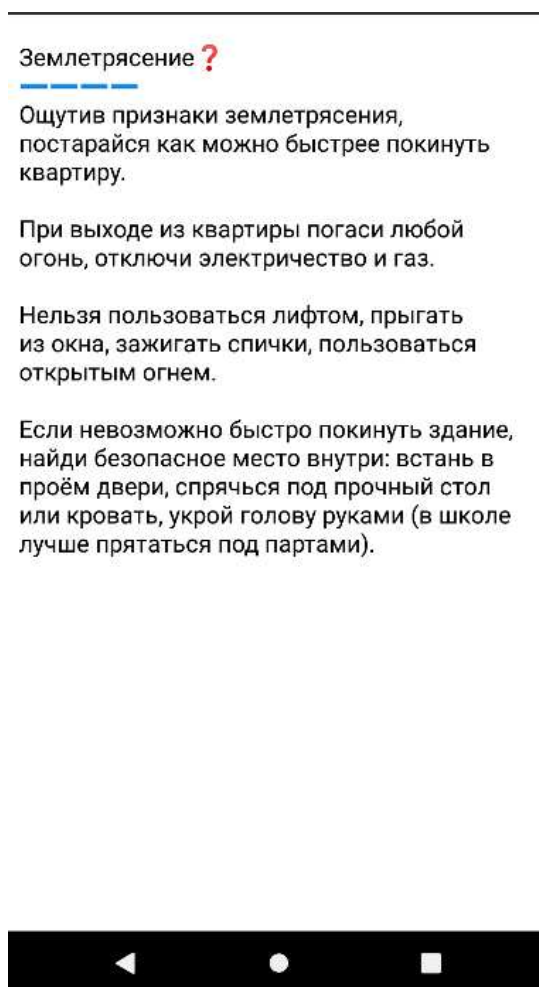


Рисунок 38 – Инструкция «Землетрясение»

Код на листе 2 рисунка 37 создает кнопки для вызова помощи. Такие кнопки создаются в коллекции ButtonPhone документа. Пример такого документа в Firestore показан на рисунке 39.

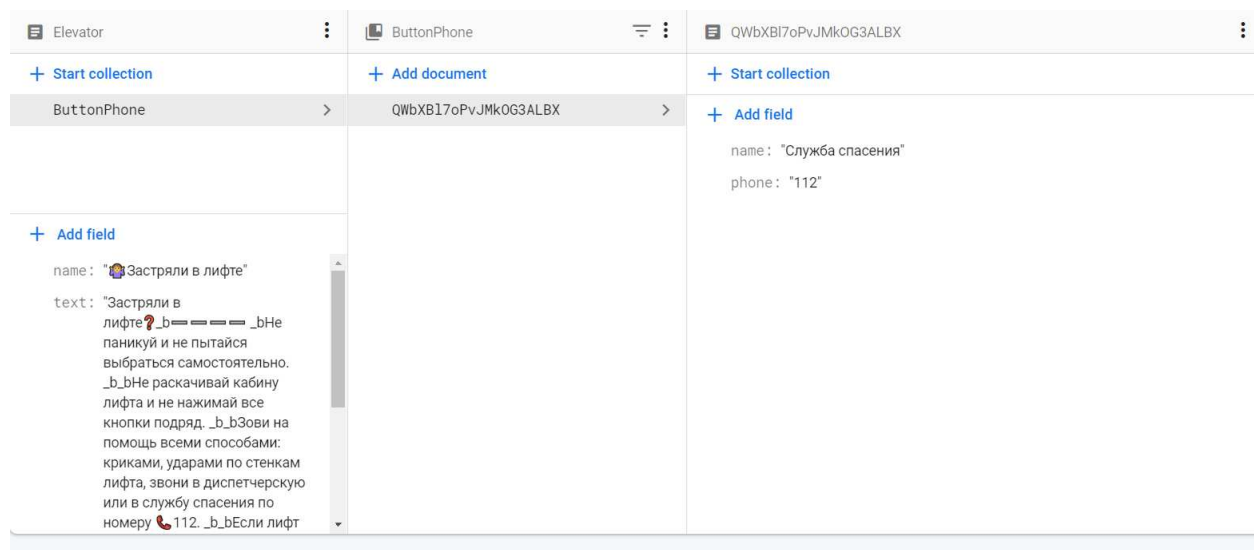


Рисунок 39 – Инструкция «Застрял в лифте»

Как видно на рисунке 39, имеется коллекция ButtonPhone в документе Elevator. В коллекции документы с полями name и phone. На основе данных в полях name и phone документов будут создаваться кнопки. Пример данного типа инструкции в приложении изображен на рисунке 40.

На улице гололед ?

В гололёд надевай обувь с рельефной подошвой.

Передвигайся осторожно, не торопись, наступая на всю ступню.

В момент падения постарайся сгруппироваться и перекатиться, тем самым смягчив удар о землю.

Если упали и сильно пострадали, вызовите спасателей.



Рисунок 40 – Инструкция «Гололёд»

Код на листе 3 рисунка 37 создает кнопки с текстом. Такие кнопки создаются из документов коллекции ButtonText. Пример данной инструкции в Firestore изображен на рисунке 41.

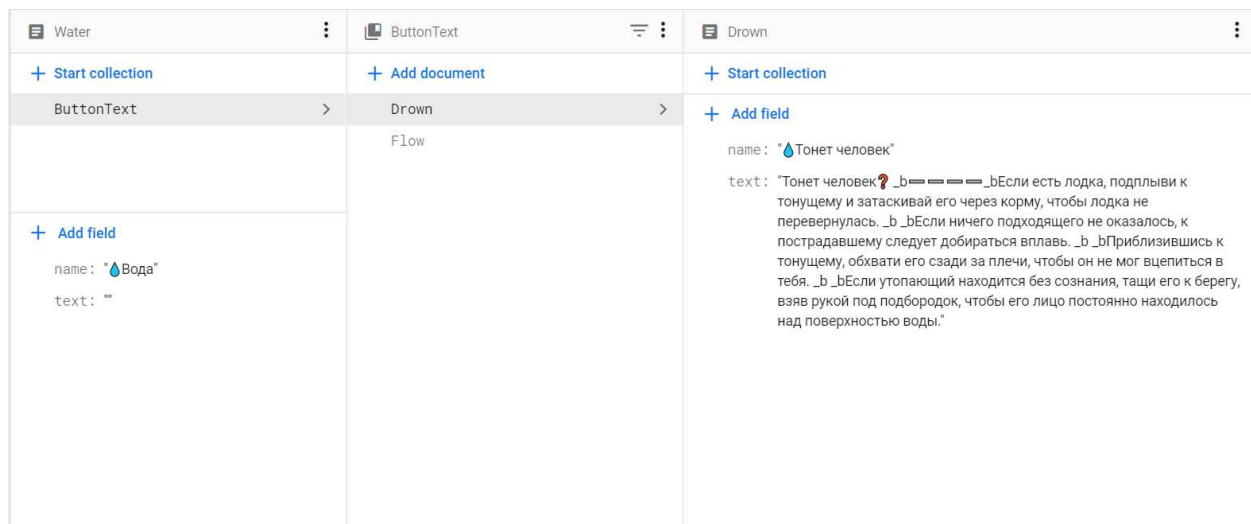


Рисунок 41 – Коллекция ButtonText в Firestore

Как видно на рисунке 41, в документе Water содержится коллекция ButtonText с документами Drown и Flow. В документах содержатся поля name и text. Кнопки будут создаваться на основе данных полей. Пример данной инструкции в приложении изображен на рисунках 42, 43, 44.



Рисунок 42 – Кнопка «Вода»

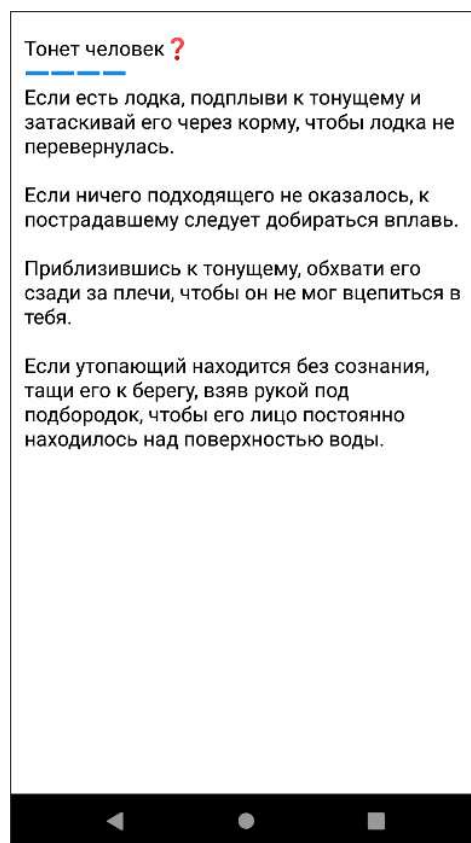


Рисунок 43 – Инструкция «Тонет человек»

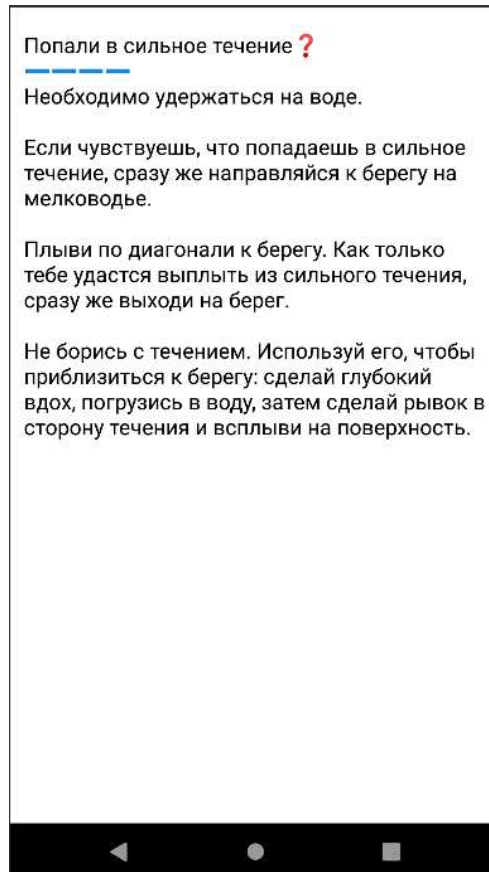


Рисунок 44 – Инструкция «Попали в сильное течение»

Можно создавать одновременно кнопки с текстом и функцией звонка, пример такой инструкции в Firestore изображен на рисунке 45.

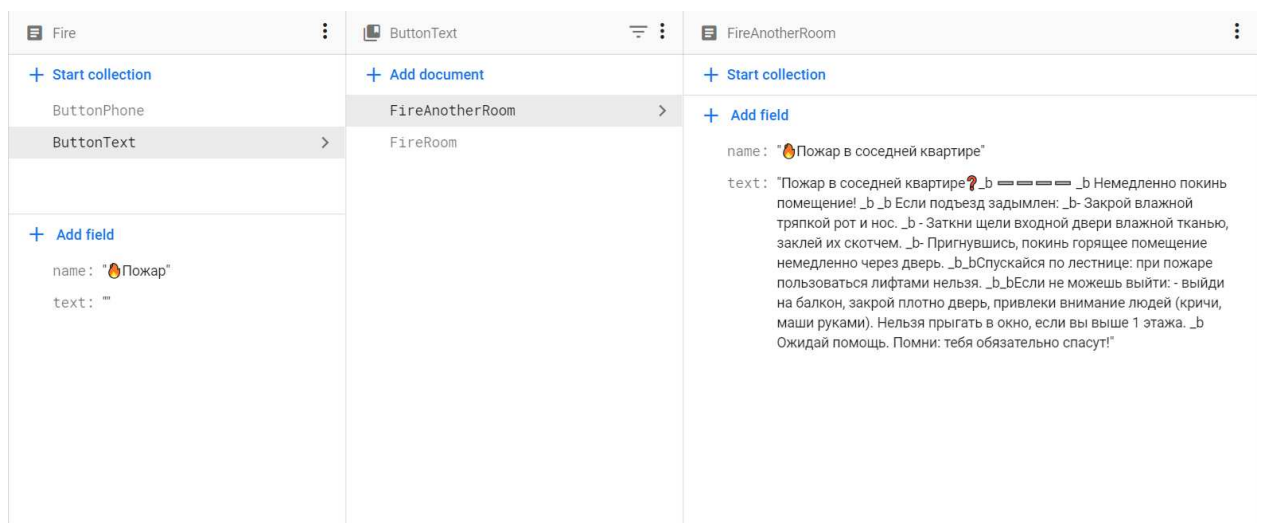


Рисунок 45 – Инструкция «Пожар» в Firestore

Как видно на рисунке 45, в документе Fire присутствуют две коллекции – ButtonText и ButtonPhone. На основе информации из документов будут создаваться кнопки. Пример данной инструкции изображен на рисунках 46, 47, 48.



Рисунок 46 – Кнопка «Пожар»



### Пожар в соседней квартире ?

Немедленно покинь помещение!

Если подъезд задымлен:

- Закрой влажной тряпкой рот и нос.
- Заткни щели входной двери влажной тканью, заклей их скотчем.
- Пригнувшись, покинь горящее помещение немедленно через дверь.

Спускайся по лестнице: при пожаре пользоваться лифтами нельзя.

Если не можешь выйти: - выйди на балкон, закрой плотно дверь, привлеки внимание людей (кричи, маши руками). Нельзя прыгать в окно, если вы выше 1 этажа. Ожидай помощь. Помни: тебя обязательно спасут!



Рисунок 47 – Инструкция «Пожар в соседней квартире»

### Пожар в твоей квартире ?

Закрой влажной тряпкой рот и нос.

Пригнувшись, покинь горящее помещение немедленно через дверь.

Если не можешь выйти:

- выйди на балкон, закрой плотно дверь, привлеки внимание людей (кричи, маши руками). ИЛИ
- зайди в ванную, заткни мокрой тканью дверные щели и вентиляцию, набери в ванную воду.

Нельзя прыгать в окно, если вы выше 1 этажа.

Ожидай помощь. Помни: тебя обязательно спасут!



Рисунок 48 – Пожар в твоей квартире

В MenuActivity так же необходимо будет вставить код для определения параметров расположения кнопок и графических настроек кнопок.

Теперь создадим новый Activity для push-уведомлений и вставим код, изображенный на рисунке 49 [10].

```
16 public class PushNotificationsService extends FirebaseMessagingService {
17     public void onMessageReceived(@NonNull RemoteMessage remoteMessage) {
18         String title = remoteMessage.getNotification().getTitle();
19         String text = remoteMessage.getNotification().getBody();
20         final String CHANNEL_ID = "HEADS_UP_NOTIFICATION";
21         NotificationChannel channel = new NotificationChannel(
22             CHANNEL_ID,
23             name: "Heads Up Notifications",
24             NotificationManager.IMPORTANCE_HIGH
25         );
26         getSystemService(NotificationManager.class).createNotificationChannel(channel);
27         Notification.Builder notification = new Notification.Builder(context: this, CHANNEL_ID)
28             .setContentTitle(title)
29             .setContentText(text)
30             .setSmallIcon(R.drawable.ic_launcher_background)
31             .setAutoCancel(true);
32         NotificationManagerCompat.from(this).notify(id: 1, notification.build());
33         super.onMessageReceived(remoteMessage);
34     }
35 }
```

Рисунок 49 – Код конструктора push-уведомлений

Обычно Push-уведомления работают и без данного кода, нужно только добавить код в манифест, изображенный на рисунке 50. Код на рисунке 49 необходим для отображения уведомлений, если приложение закрыто. Manifest – это файл, в котором содержится основная информация для сборки приложения: структура приложения, объявляет разрешения, объявляет минимальный API и так далее [11].

```
40 <service
41     android:name=".PushNotificationsService" android:exported="false">
42     <intent-filter>
43         <action android:name="com.google.firebase.INSTANCE_ID_EVENT"/>
44     </intent-filter>
45 </service>
```

Рисунок 50 – Добавление сервиса в манифест

## **2.4 Выводы по разделу «Разработка мобильного приложения «Детская безопасность»»**

В данном разделе было поэтапно описано создание и подключение проекта к Firebase. Приложение подключено к Firebase. Кнопки вызова и текста в мобильном приложении разработаны таким образом, что при добавлении документа с определенными полями в базу данных, кнопка создается в приложении на основе полей в документах. Разработана функция отправки push-уведомлений, что позволит уведомлять о начале сезонов, поднятии воды и так далее. Разработаны кнопки для экстренного вызова определенных служб: службы спасения, телефон уполномоченного по правам ребенка, телефон доверия.

## **3 Оценка затрат на реализацию мобильного приложения «Детская безопасность»»**

### **3.1 Капитальные затраты**

Капитальные (единовременные) затраты на ИС носят разовый характер. Свою стоимость они переносят на продукцию по частям за счет амортизационных отчислений. Капитальными их называют потому, что они не утрачиваются, а воспроизводятся.

Применительно к ИС принято группировать капитальные затраты по формуле

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

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

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

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

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

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

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

$K_{оэ}$  – затраты на опытную эксплуатацию.

Для начала рассчитываем затраты на проектирование  $K_{пр}$  по формуле

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

где  $K_{зп}$  – затраты на заработную плату проектировщиков;

$K_{ипс}$  – затраты на инструментальные программные средства;

$K_{свт}$  – затраты на средства вычислительной техники;

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

Рассчитываем заработную плату проектировщика. Минимальный оклад составляет 13 890 рублей в месяц [1]. С учетом того, что разработчик будет работать 22 дня в месяц, получим зарплату 631 рубль в день. Планируемый срок разработки проекта – 43 дня. 631 умножим на количество рабочих дней (43) и в итоге получим, что оплата составит 27 148 рублей за 43 дня.

Консультация с заказчиком – 2 дня.

Проектирование приложения – 2 дня.

Разработка приложения – 30 дней.

Тестирование – 7 дней.

Ввод в эксплуатацию – 2 дня.

Теперь необходимо рассчитать зарплату с учетом северного коэффициента (30%), региональной надбавки (30%) и НДФЛ (13%). Расчет зарплаты проектировщика предоставлен в таблице 3.

Таблица 3 – Состав зарплаты

<b>Состав зарплаты</b>	<b>Планируемая сумма, руб.</b>
Оклад	27 148
Районный коэффициент	8 144
Региональная надбавка	8 144
НДФЛ	5 646
Итого	43 437
На руки	37 791

И в итоге планируемая заработная плата:  $K_{зп} = 43\,437 * 1,302$  (отчисление во внебюджетные фонды) = 56 555 рублей.

Расчет  $K_{инс}$  включает в себя затраты на инструментальное программное обеспечение, необходимое для разработки мобильного приложения. Расчет предоставлен в таблице 4.

Таблица 4 – Таблица стоимости инструментальных средств

<b>Программное обеспечение</b>	<b>Количество</b>	<b>Срок использования</b>	<b>Цена, руб.</b>
Microsoft Windows 10 Home	1	Бессрочно	14 000
Android Studio	1	Бессрочно	Бесплатно

Microsoft Windows 10 Home стоит 14 000 рублей и покупается на бессрочную основу, бесплатно поддерживается и обновляется. Соответственно рассчитываем стоимость за 5 лет и амортизацию за 43 дня.

$$K_{инс} = 14\,000 / 5 / 365 * 43 = 330 \text{ рубль.}$$

Android Studio распространяется на бесплатной основе бессрочно.

Сервер предоставляется компанией, поэтому вставлять в расчет нет необходимости.

В таблице 5 представлен расчет персонального компьютера, необходимый разработчику для создания приложения.

Таблица 5 – Расчет стоимости персонального компьютера

Комплектующие	Количество	Срок службы, лет	Цена, руб.
AMD Ryzen 7 Summit Ridge 1700 OEM	1	5	10 000
MSI MPG X570 GAMING PLUS	1	5	13 000
PowerColor Radeon AXRX 580 4GBD5-DMV2	1	5	12 000
HyperX Predator DDR4 2x8Gb HX426C13PB3K2/16	1	5	6 500
Western Digital Black SN750 WDS500G3XHC	1	5	9 000
Chieftec Polaris PPS-750FC	1	5	7 100
Deepcool Neptwin V2	1	5	3 400
Итого			61 000

В связи с сложившейся экономической ситуацией в стране, цены взяты за 2020 год. Далее необходимо рассчитать амортизацию. Планируемый срок службы комплектующих 5 лет, поэтому рассчитаем амортизацию на срок 5 лет. В первую очередь рассчитаем норму амортизации. Амортизация - это процесс

переноса по частям стоимости основных средств и нематериальных активов по мере их физического или морального износа на себестоимость производимой продукции. Рассчитывается амортизация по формуле

$$H_{\text{ам}} = 100\% / T_{\text{эк}}, \quad (3)$$

где  $H_{\text{ам}}$  – норма эксплуатации;

$T_{\text{эк}}$  – срок эксплуатации компьютера.

Разделим 100% на срок службы комплектующих.

$$H_{\text{ам}} = 100\% / 5 = 20\%$$

Получим норму амортизации за год в 20%. Теперь рассчитаем амортизацию за год использования по формуле

$$A_{\text{год}} = C_{\text{б}} * H_{\text{ам}}, \quad (4)$$

где  $A_{\text{год}}$  – амортизация за год использования;

$C_{\text{б}}$  – балансовая стоимость;

$H_{\text{ам}}$  – норма амортизации.

Итоговую стоимость комплектующих разделим на норму амортизации за год.

$$A_{\text{год}} = 61\,000 * 0,2 = 12\,200 \text{ рублей}$$

Получим амортизацию за год использоваться в сумме 12 200 рублей. Теперь рассчитаем проектную амортизацию. Для расчета проектной амортизации необходима формула

$$A_{\text{пр}} = A_{\text{год}}/K_{\text{рдг}}, \quad (5)$$

где  $A_{\text{пр}}$  – проектная амортизация;

$A_{\text{год}}$  – амортизация за год использования;

$K_{\text{рдг}}$  – рабочих дней в году.

Рассчитываем проектную амортизацию, разделив сумму амортизации за год использования (12 200 рублей) на рабочих дней в году (247) и умножим на число рабочих дней (43).

$$A_{\text{пр}} = 12200/247 * 43 = 2\ 123 \text{ рублей.}$$

В результате получим затраты на средства вычислительной техники  $K_{\text{свт}} = 2\ 123$  рублей.

В  $K_{\text{проч}}$  будет входить затраты на электроэнергию, обслуживание серверов и остальные не учтенные расходы на разработку. Прочие затраты составляют 5% от общей суммы затрат.

$$K_{\text{проч}} = (56555 + 330 + 2123) * 0,05 = 2\ 951 \text{ рублей.}$$

$K_{\text{зп}} = 56\ 555$  рублей. Затраты на заработную плату программисту.

$K_{\text{ипс}} = 330$  рубль. Затраты на инструментальные средства.

$K_{\text{свт}} = 2\ 123$  рублей. Затраты на средства вычислительной техники.

$K_{\text{проч}} = 2\ 951$  рублей. Прочие затраты

$$K_{\text{пр}} = 56555 + 330 + 2123 + 2951 = 61\ 959 \text{ рублей.}$$

$K_{\text{тс}}$  – затраты на технические средства управления учтены в прочих затратах и равняется 0, так как время использования компьютера незначительно.



$K_{лс}$  – затраты на линии связи будут включать в себя затраты на добавление приложения в Play Market. Для добавления приложения необходимо заплатить 25\$, что составляет 2 666 рублей. В разработанной версии на начальном этапе может быть использована бесплатная версия Firebase. При удачном продвижении мобильного приложения в будущем понадобится приобретение платной версии. Данные затраты будут учтены в косвенных затратах проекта.

$K_{по}$  – затраты на программные средства отсутствуют, так как ПО для внедрения уже присутствует на предприятии.

$K_{ио}$  – затраты на формирование информационной базы отсутствуют, так как база заполнена на стадии проектирования.

$K_{об}$  – затраты на обучение персонала. Сотрудник ЦУР компетентен, поэтому обучение не потребуется.

$K_{оэ}$  – затраты на опытную эксплуатацию будут включать в себя этап тестирования и проведение инструктажа для инженера программиста в предприятии ЦУР. Будет учитываться работа программиста и собственная зарплата на период внедрения. Для программиста зарплата за 2 часа. Собственная – 7 дней.

$$K_{оэ} = (631*7)+(79*2) = 4\,417 + 158 = 4\,574 \text{ рублей.}$$

$$K = 61\,959 + 2\,666 + 4\,574 = 69\,199 \text{ рублей.}$$

Состав капитальных затрат изображен на рисунке 51.

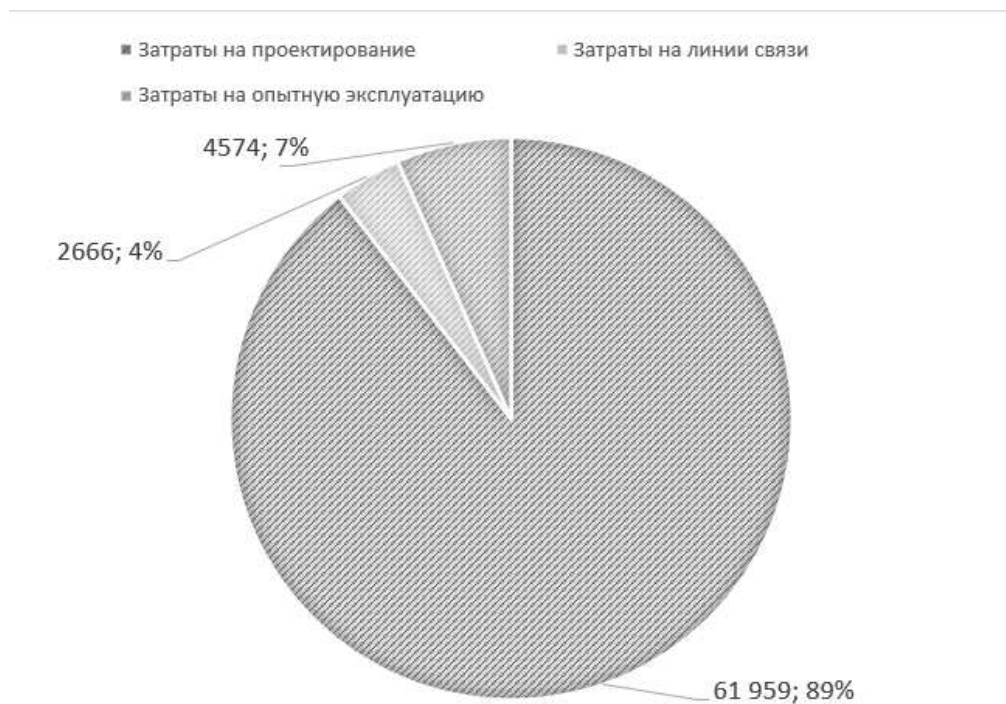


Рисунок 51 – Структура капитальных затрат

Как видно на рисунке 51, большую часть капитальных затрат составит зарплата специалисту и составляет 89% от капитальных затрат.

### 3.2 Расчет эксплуатационных затрат

Расчет эксплуатационных затрат проводится по следующей формуле

$$C = C_{\text{зп}} + C_{\text{ао}} + C_{\text{то}} + C_{\text{лс}} + C_{\text{ни}} + C_{\text{проч}}, \quad (6)$$

- 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.

Зарплата персонала. В течении следующего года работы приложения персонал будет заниматься над исправлением ошибок и доработкой приложения около 15 рабочих дней. Рассчитываем зарплату за 15 дней.

$$13\ 890/22*15 = 9\ 471 \text{ рубль.}$$

Полный расчет и состав зарплаты приведен в таблице 6.

Таблица 6 – Состав зарплаты в период внедрения

<b>Состав зарплаты</b>	<b>Планируемая сумма, руб.</b>
Оклад	9 471
Районный коэффициент	2 842
Региональная надбавка	2 842
НДФЛ	1 969
Итого	15 155
На руки	13 186

Далее рассчитаем отчисления во внебюджетные фонды, и заработная плата составит:

$$C_{\text{зп}} = 15\ 155 * 1,302 = 19\ 732 \text{ рублей.}$$

Затраты на амортизационные отчисления отсутствуют, так как будет применяться для обслуживания приложения компьютер, который уже присутствует у компании. Программное обеспечение для обслуживания приложения бесплатное и не потребует дополнительных затрат.

Затраты на техническое обслуживание отсутствуют, так как будет применяться для обслуживания приложения компьютер, который уже присутствует у компании. Программное обеспечение для обслуживания приложения бесплатное и не потребует дополнительных затрат.

Затраты на использование глобальных сетей отсутствуют, так как уже учтены компанией.

Затраты на носители информации отсутствуют, так как приложения будут располагаться на сервере компании.

Прочие затраты составляют 5% от суммы всех затрат.

$$19732 * 5\% = 987 \text{ рублей.}$$

$$C = 19732 + 987 = 20719 \text{ рублей.}$$

Ниже в таблице 7 приведен состав эксплуатационных затрат.

Таблица 7 – Состав эксплуатационных затрат

Состав эксплуатационных затрат	Затраты, руб.
Затраты на заработную плату персонала	19 732
Прочие затраты	987

На рисунке 52 изображена круговая диаграмма с долей каждой части эксплуатационных затрат.

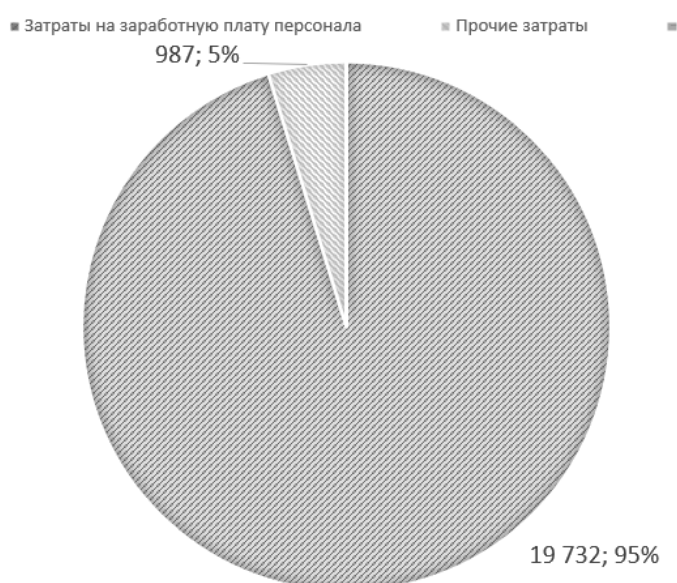


Рисунок 52 – Структура эксплуатационных затрат

Как видно на рисунке 52, структура по большей части состоит из затрат на заработную плату персонала и составит 95% от эксплуатационных затрат.

### **3.3 Расчет совокупной стоимости владения мобильным приложением**

Совокупная стоимость приложения рассчитывается по формуле

$$DE = DE1 + DE2 + DE3 + DE4 + DE5 + DE6 + DE7 + DE8 \quad (7)$$

DE1 будет включать в себя капитальные затраты. Рассчитав капитальные затраты, получим сумму в 69 199 рублей.

DE2 будет включать в себя эксплуатационные затраты. Ранее рассчитанные эксплуатационные расходы составили 20 719 рублей.

DE3 включает в себя расходы на техническую поддержку автоматизированного обеспечения и программного обеспечения.

DE4 включает в себя разработку ПО внутренними силами. Так как такая необходимость отсутствует, DE4 будет равен 0.

DE5 включает в себя аутсорсинг.

В DE6 входят командировочные расходы. Командировочные расходы будут равняться 0.

В DE7 входят расходы на услуги связи.

В DE8 входят другие типы расходов.

$$DE = 69\,199 + 20\,719 = 89\,918 \text{ рублей.}$$

### **3.4 Риски реализации проекта**

На старте разработки проекта невозможно знать все, что случится в процессе его реализации. Для минимизации вреда необходимо изучить риски,

которые могут повлиять на запуск приложения и минимизировать их вероятность

Рабочая группа проекта пытается определить и измерить риски, свойственные IT-проектам, а также неопределенности, возникающие непосредственно на этапе проведения оценки. Оценки рисков приложения приведены в таблице 8.

Таблица 8 – Оценка рисков проекта

№	Группы рисков	Перечень рисков проекта	Уровень влияния риска на проект	Вероятность риска	Возможность предотвращения или снижения риска
1	Реализационный риск	Уход Android Studio из России	Высокий	Низкая	На данный момент нет возможность минимизировать данный риск
		Блокировка Google сервисов (Firebase и Google play)	Средний	Низкая	Использование других не заблокированных аналогов
		Увеличение расходов на разработку и внедрение	Средний	Средняя	В связи с нестабильной ситуацией на рынке, запланировать стоимость проекта с начала разработки сложно.
2	Риск соответствия	Несоответствие требованиям заказчика	Низкий	Низкая	Составление четкого технического задания

### **3.5 Мероприятия по снижению рисков**

Проанализировав риски, был сделан вывод, что в текущие дни разрабатывать проекты довольно рискованно и самое важное, от разработчика зависит не так много. Разберем возможные мероприятия по снижению рисков.

Реализационный риск. Уход Android Studio из российского рынка не позволит обновлять мобильное приложение, написанное в Android Studio. Для снижения данного риска есть возможность перейти на другой аналог под названием IntelliJ IDEA или Eclipse. Но у России нет собственных аналогов, поэтому полностью нивелировать данный риск не выйдет, т.к. у каждой IDE есть вероятность блокировки или ограничений, хоть она довольно низкая.

Блокировка Google сервисов скажется на функционале приложения и распространении. При ограничении Firebase будет отсутствовать доступ к push-уведомлениям и удаленной базе данных, а ограничение Google Play не позволит распространять данное приложение эффективно. На сегодняшний день аналогов у России нет, соответственно отсутствует возможность нивелировать данные риски полностью.

Несвоевременная сдача проекта может быть связана с временными ограничениями или ухода с рынка важных информационных технологий и/или оборудования. Поэтому на текущий день риск остается довольно высок и нивелировать полностью не получится.

Увеличение расходов на разработку и внедрение связано с нестабильной ситуацией с валютой.

### **3.6 Выводы по разделу «Оценка затрат на реализацию мобильного приложения «Детская безопасность»**

Мобильное приложение «Детская безопасность» не несет никакой экономической пользы напрямую. Цель данного приложения принести общественную пользу. Приложение направлено на снижение несчастных

случаев, которые могут произойти чаще всего в текущее время. Зачастую это пожары, скуллшутинг, буллинг, агрессивное поведение незнакомцев и т.д. Так как данный проект не несет экономической эффективности, бюджет на разработку приложения планируется получать из государственного гранта. Предполагаемая стоимость разработки приложения 89 918 рублей. Данная сумма была взята из расчета ТСО.



## ЗАКЛЮЧЕНИЕ

Таким образом, в процессе выполнения выпускной квалификационной работы в первом разделе была проанализирована деятельность ЦУР Республики Хакасия, проанализированы аналогичные мобильные решения, проанализированы и выбраны инструменты для разработки, построена и описана DFD модель, демонстрирующая работу пользователя с приложением. Во втором разделе была поэтапно описана разработка мобильного приложения «Детская безопасность» и работа с базой данных Firebase, разработаны push-уведомления. В третьем разделе проведён расчет капитальных затрат, эксплуатационных затрат и стоимость проекта. Оценены возможные риски и пути их решения. Поставленные задачи осуществлены в полной мере, а именно:

1. охарактеризована основная деятельность Центра управления регионом Республики Хакасия, выявлены основные задачи ЦУР Республики Хакасия, поставлены цели и задачи проекта, описана актуальность проекта;
2. обоснован выбор инструментальных средств разработки, была выбрана среда разработки Android Studio и язык программирования Java.
3. разработано мобильное приложение «Детская безопасность», которое поможет ребёнку в трудной ситуации. Приложение позволит получить быстрый доступ к инструкции и номерам телефона экстренных служб. Push-уведомления позволят уведомить ребёнка о опасном сезоне или напомнить о возможности поговорить о проблемах, позвонив в «Телефон доверия».
4. оценены затраты на реализацию мобильного приложения «Детская безопасность». В результате расчета капитальных и эксплуатационных затрат, затраты на проект составили – 89 918 рублей.
5. проанализированы возможные риски и пути их решения. Все описанные риски получили оценку – какое влияние оказывают на проект и вероятность возникновения.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Разработка игр и приложений: «3 лучшие среды разработки для Android» // app-android.ru : [сайт]. – 2021. – URL: <https://app-android.ru/blog/environment-develop-android> (дата обращения: 29.03.2022).
2. Дизайн и Веб-разработка для бизнеса: «Сайты на Java: плюсы и минусы» // landing.ua : [сайт]. – 2021. – URL: <https://landing.ua/blog/sajty-na-java-plyusy-i-minusy.html> (дата обращения: 02.04.2022).
3. Полежаева, О. «Kotlin: что это за язык, где и как используется» / О, Полежаева // checkroi.r : [сайт]. – 2021. – URL: <https://checkroi.ru/blog/kotlin-cto-eto-za-yazyk/> (дата обращения: 04.04.2022).
4. Lucidchart: «Что такое DFD диаграмма и как её создать?» // lucidchart.com : [сайт]. – 2021. – URL: <https://www.lucidchart.com/pages/ru/диаграмма-dfd> (дата обращения: 06.04.2022).
5. Правительство Республики Хакасия: «О создании и функционировании Центра управления регионом Республики Хакасия» // r-19.ru : [сайт]. – 2021. – URL: <https://r-19.ru/documents/140/105469.html> (дата обращения: 08.04.2022).
6. developers: «Button» // developer.android.com [сайт]. – 2021. – URL: <https://developer.android.com/reference/android/widget/Button> (дата обращения: 09.04.2022).
7. Firebase: «Access data offline » // firebase.google.com [сайт]. – 2021. – URL: <https://firebase.google.com/docs/firestore/manage-data/enable-offline> (дата обращения: 10.04.2022).
8. Firebase: «Get data with Cloud Firestore» // firebase.google.com [сайт]. – 2021. – URL: <https://firebase.google.com/docs/firestore/query-data/get-data> (дата обращения: 11.04.2022).
9. developers: «Dialogs» // developer.android.com [сайт]. – 2021. – URL: <https://developer.android.com/guide/topics/ui/dialogs> (дата обращения: 14.04.2022).

10. Firebase: «Firebase Cloud Messaging» // [firebase.google.com](https://firebase.google.com) [сайт]. – 2021. – URL: <https://firebase.google.com/docs/cloud-messaging> (дата обращения: 17.04.2022).

11. developers: «App Manifest Overview» // [developer.android.com](https://developer.android.com) [сайт]. – 2021. – URL: <https://developer.android.com/guide/topics/manifest/manifest-intro> (дата обращения: 21.04.2022).

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 11 наименований.

Один экземпляр сдан на кафедру.

« \_\_\_\_ » \_\_\_\_\_ 2022 г.


\_\_\_\_\_ Белицкий Андрей Валерьевич

подпись

Министерство науки и высшего образования РФ  
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  
«17» июня 2022 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.03 Прикладная информатика

Разработка мобильного приложения «Детская безопасность» для ЦУР  
Республики Хакасия

Руководитель  17.06.22 ст. преподаватель В. И. Кокова  
подпись, дата

Выпускник  17.06.22 А. В. Белицкий  
подпись, дата

Консультанты  
по разделам:

Экономический  17.06.22 Е. Н. Скуратенко  
подпись, дата

Нормоконтролер  17.06.22 В. И. Кокова  
подпись, дата