

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт информационных технологий и анализа данных

наименование института

Допускаю к защите  
Руководитель

Бахвалова З.А.

Фамилия И.О.

**РАЗРАБОТКА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

наименование темы

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
к курсовой работе по дисциплине  
Технологии разработки программных

1.019.00.00 ПЗ

обозначение документа

Выполнил студент группы ИСТб 17-1  
шифр группы

  
подпись

Мускатин И.А.

Фамилия И.О.

Нормоконтроль

подпись

Бахвалова З.А.

Фамилия И.О.

Курсовая работа защищена с оценкой

Иркутск 2020г.

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ЗАДАНИЕ  
НА КУРСОВУЮ РАБОТУ**

По курсу Технологии разработки программных комплексов  
Студенту Мускату Игору Алексеевичу  
(фамилия, инициалы)

Тема работы: Разработка прикладного программного обеспечения

Исходные данные:

Разработка мобильного приложения для контакта подрядчиков и  
проверяющих

1. Разработка приложения ведётся на объектно-ориентированном языке Java;
2. Используется свободная интегрированная среда разработки приложений  
Android Studio;

Рекомендуемая литература:

1. Пример использования диаграмм UML для описание предметной области  
области (документ Microsoft word). Составители: Бахвалова З.А.
2. Анализ ПО 25.11.16 (презентация). Составители: Бахвалова З.А.
3. Курсовая работа «Разработка прикладного программного обеспечения»  
Руководство и методические указания для студентов направления: 09.03.02  
«Информационные системы и технологии». Составитель Бахвалова З.А. –  
Иркутск, Изд-во ИРНИТУ, 2018, 61 с.
4. ВКР «АИС «Мониторинг обслуживания мостовых сооружений» для МУП  
«Служба эксплуатации мостов» г.Иркутска»

Графическая часть на \_\_\_\_\_ листах.

Дата выдачи задания \_\_\_\_\_

Задание получил студент \_\_\_\_\_  
 подпись Мускатин И.А.  
Фамилия И.О.

Дата представления работы  
руководителю \_\_\_\_\_

Руководитель курсового  
проектирования \_\_\_\_\_  
\_\_\_\_\_ подпись Бахвалова З.А.  
Фамилия И.О.

## Содержание

Введение.....	4
1 Выработка системных требований .....	5
1.1 Описание предметной области .....	5
1.2 Требования к разрабатываемой системе.....	9
1.3 Моделирование.....	12
1.4 Требования пользователя .....	17
1.5 Описание поведения разрабатываемой системы .....	22
2 Разработка технического задания.....	23
3 Архитектурное и интерфейсное проектирование программного продукта. ....	29
3.1 Описание инструментов ПО .....	29
3.2 Логическая модель представления данных .....	29
3.3 Архитектура ПО (Функциональная модель).....	31
3.4 Интерфейсные классы программы.....	32
3.5 Проектирование макетов интерфейса.....	34
4 Планирование и управление проектом .....	38
4.1 Иерархическая структура проекта .....	40
4.2 Сетевая модель проекта.....	40
4.2.1 Управление временем проекта .....	40
4.2.2 Учет критичности .....	41
4.3 Планирование трудовых ресурсов проекта .....	43
4.4 Учет неопределенностей проекта методом «оценки и пересмотра планов (ПЕРТ)» .....	44
4.5 Программное обеспечение проекта.....	47
5 Программная реализация.....	49
6 Тестирование .....	66
7 Разработка документации.....	79
Заключение .....	83
Список использованных источников .....	84

## **Введение**

Целью данной работы является разработка программного продукта, автоматизирующего работу предоставления данных от подрядчика проверяющему.

С применением цифровых технологий появилась возможность оказывать влияние на такие характеристики ЖКХ как производительность труда, функциональная эффективность, качество обслуживания потребителей и объектов инфраструктуры.

Автоматизированные информационные системы являются разновидностью цифровых технологий и теперь уже широко применяются в государственных учреждениях и муниципальных унитарных предприятиях. Стоит отметить, что после выхода постановления Правительства Российской Федерации от 16 ноября 2015 г. N 1236 большее предпочтение стали отдавать именно отечественной разработке.

Автоматизация позволит повысить уровень контроля за выполнением работ подрядчиками, связанных с эксплуатацией мостовых объектов города.

## **1 Выработка системных требований**

### **1.1 Описание предметной области**

Муниципальное унитарное предприятие «Служба эксплуатации мостов» г. Иркутска создано в апреле 1995 года и является подразделением комитета городского обустройства. В обслуживании у предприятия находятся различные городские мостовые сооружения (18 автодорожных мостов, 16 путепроводов, 4 пешеходных моста, 1 участок берегоукрепления реки Ушаковка, а также очистные сооружения левого и правого берега Академического моста).

Предприятие сотрудничает с организациями-подрядчиками, которые осуществляют обслуживание мостовых сооружений и выполняют свои обязанности согласно заключенному договору.

Подрядчик самостоятельно составляет логику своих работ. Результатом проводимой логики является календарный график обслуживания каждого мостового сооружения, который передается в МУП.

Подрядчики обязуются обслуживать мостовые сооружения согласно составленному календарному графику выполнения работ. После проведения некоторой работы подрядчик должен отправить следующие сведения о выполненной работе:

- фотоматериалы состояния мостового объекта до проведения работ;
- фотоматериалы состояния мостового объекта после проведения работ;
- пояснения и уточнения о выполненной работе. (Информация о том, когда и где был проведен тот или иной вид работ, какое время было затрачено на выполнение работ и сколько человек приняли участие в ее выполнении)

На момент предпроектного обследования отчеты о выполненных работах отправлялись в чат мессенджера, в котором происходило все информационное взаимодействие между представителями МУП и всеми подрядчиками, выполняющими работы на объектах. Управляющие лица МУП «СЭМ» могут просматривать информацию выполненных работах, в чате мессенджера они являются проверяющими. У каждого проверяющего на контроле находятся несколько подрядчиков, их общение в чате не разграничено. Отчеты о выполненных работах поступают неструктурированно, т.к. подрядчики могут одновременно отправлять сообщения в мессенджере. Это затрудняет проверку выполненных работ.

Проверяющий оценивает качество выполнения работ по отправленным подрядчиком материалам. Работы по обслуживанию мостового сооружения могут быть приняты или не приняты с указанием замечаний к исправлению. Подрядчик должен провести работы по устранению замечаний и также отправить отчет в том же формате сообщения о выполненной работе.

Фиксация выполненных работ или возникших замечаний на момент предпроектного обследования осуществлялась в журналах, проверяющих.

Такой журнал представлен в бумажной форме, а сведения в нем – записи, выполненные вручную.

Вся информация о выполненных работах хранится в чате мессенджера около 30 дней. Хранение дольше срока одного месяца не требуется, так как ежемесячно по результатам выполнения работ происходит их оплата. После подписания отчетных документов, связанных с оплатой выполненных работ, все фотоматериалы и сопутствующая информация удаляются.

Для проверки качества деятельности подрядчиков проверяющими периодически проводятся выездные проверки (внешние осмотры). Главная цель таких осмотров – выявление недобросовестных подрядчиков, которые могут отправить старые фотоматериалы или же фотоматериалы с другого объекта. Все выявленные замечания также сообщаются подрядчику, в чьем подчинении находится проверяемое мостовое сооружение.

Во время таких осмотров проверяющие создают фотоматериалы самостоятельно. Накопленные фотоматериалы в совокупности образуют архив, фото из которого в дальнейшем используются для отчетности о качестве обслуживания мостовых объектов подрядчиками. Распределение фото в архив происходит вручную. Все фотоматериалы с внешних осмотров проверяющего хранятся на сервере в виде набора папок, отсортированных по годам, месяцам и объектам.

Распределение фотоматериалов в архив происходит путем просмотра фото мостового сооружения для определения его территориальной принадлежности. Такой подход к распределению материалов требует существенных затрат времени и не защищает от ошибок, связанных с человеческим фактором.

В таблице 1 приведены основные профессиональные термины, а также наиболее часто встречающиеся сокращения.

Таблица 1- Термины и сокращения

Термин	Сокращение	Описание
Автоматизированная информационная система	АИС	Совокупность содержащейся в базах данных информации и обеспечивающих её обработку информационных технологий и технических средств
База данных	БД	Организованная структура, предназначенная для хранения информации
Искусственное сооружение	ИССО	Автомобильный мост, надземный пешеходный переход, путепровод и остальные виды мостовых сооружений
Муниципальное унитарное предприятие	МУП	

Продолжение таблицы на следующей странице

Продолжение таблицы 1

Комитет городского обустройства	КГО	
Служба эксплуатации мостов	СЭМ	
Подрядчик		Представитель подрядной организации, который является ответственным за выполнение работ на объекте и отчитывается за выполнение этих работ
Проверяющий		Лицо, осуществляющее проверку качества обслуживания мостовых сооружений
Внешний осмотр		Выездная проверка на мостовое сооружение, которая осуществляется проверяющим с целью лично убедиться в качестве работ по обслуживанию
Мостовое сооружение		Мостовой объект, подлежащий обслуживанию
Задание		Обозначение работы в контексте АИС
Фото		Фотоматериал сделанный подрядчиком

Таблица 2 – Словарь терминов предметной области

Термин	Назначение
База данных	Организованная структура, предназначенная для хранения информации
Подрядчик	Представитель подрядной организации, который является ответственным за выполнение работ на объекте и отчитывается за выполнение этих работ
Проверяющий	Лицо, осуществляющее проверку качества обслуживания мостовых сооружений
Задание	Обозначение работы в контексте АИС
Фото	Фотоматериал сделанный подрядчиком
Сообщение	Текст (и, необязательно, фото) отправленные с целью проинформировать собеседника.

Таблица 3 – Словарь предметной области по методу Аббота

Существительное	Глагол
База данных	хранить
Подрядчик	читать/загружать/отправлять

Продолжение таблицы на следующей странице

Продолжение таблицы 3

Проверяющий	читать/отправлять/создавать
Задание	создать/выполнить
Фото	загрузить
Сообщение	отправить/прочитать

Таблица 4 – Объектно-ориентированный словарь

Класс/Сущность	Свойство/Состояние	Метод/Функция/Поведение
База данных	Хранение данных пользователей системы, хранение данных из сообщений	Запись в БД, Считывание из БД
Подрядчик	Идентификатор, ФИО, уровень доступа, почта (логин)	Регистрация, авторизация, сохранение профиля в БД
Проверяющий	Идентификатор, ФИО, уровень доступа, почта (логин)	Регистрация, авторизация, назначение задания, сохранение профиля в БД
Задание	Название, дата для выполнения, ID проверяющего, ID подрядчика, ID	создание, сохранение задания в БД
Фото	Название, Ссылка на местоположение в БД	Загрузка фото в БД, доступ к фото в БД
Сообщение	Текст, автор сообщения, дата и время сообщения, адрес фотографии в БД	Отправить, отобразить, сохранить в БД

**Цель создания приложения:**

Основной целью является обеспечение информационного взаимодействия между подрядчиками и проверяющими в электронном виде, а также автоматизация ввода информации в систему о выполнении работ подрядчиками (система должна принимать фотоматериалы и сведения о выполненной работе от подрядчика) и автоматизация проверки достоверности отправляемых подрядчиками фотоматериалов (необходимо выполнять проверку фотоматериалов на соответствие геолокации отправленных подрядчиком фотоматериалов и сравнивать запланированную дату проведения работы с датой фактического проведения). Так как разрабатываемое приложение является частью большей АИС предприятия, то такие действия как проверка и выставления результатов работы будут происходить не на мобильном приложении, а на компьютерном, т.к. проверяющему будет удобнее работать с полученной информацией на мониторе компьютера.



#### Функции системы:

- Регистрация нового пользователя:
  - Ввод необходимых данных;
  - Проверка корректного ввода данных при регистрации пользователя. Вывод информации об ошибке при неправильном вводе;
  - Присвоение автоматически уникального идентификатора новому пользователю;
  - Сохранение данных пользователя в БД.
- Авторизация пользователя:
  - Доступ к БД для осуществления корректного процесса входа (сопоставление логина/пароля пользователя для успешной авторизации в приложении);
  - Проверка корректного ввода данных при регистрации пользователя. Вывод информации об ошибке при неправильном вводе.
- Создание чатов (диалогов):
  - Возможность формировать диалоги, являющиеся формой для общения конкретного подрядчика с конкретным проверяющим, по уникальному заданию.
  - Доступ к БД для сохранения/считывания данных о созданном диалоге (пользователи которые имеют к нему доступ, сообщения которые были отправлены в данном диалоге);
- Отправление сообщений:
  - Возможность ввода текста и добавления к сообщению фотоматериала;
  - При наличии прикрепленного к сообщению фотоматериала, загрузка его копий в БД, хранилище данных с формированием ссылки на него;
  - Автоматическая фиксация даты/времени и геопозиции при отправке сообщения;
  - Фиксация всех данных о сообщении в БД (текст сообщения, ссылка на фотоматериал (если он есть), дата/время отправки сообщения, геопозиция отправки сообщения, идентификатор отправителя).
- Просмотр доступных пользователю диалогов.

## 1.2 Требования к разрабатываемой системе

#### Функциональные требования:

- экран авторизации пользователя;
- список заданий (диалогов);
- список сообщений конкретного диалога;

- возможность загружать фотографии, как из памяти телефона, так при фото из приложения;
- разграничение уровня доступа, в зависимости от роли пользователя (подрядчик/проверяющий);
- возможность добавления заданий для конкретного подрядчика проверяющим;
- доступ к чатам(заданиям), только тем пользователям, которые имеют к ним доступ;

#### Интерфейсные требования:

- интерфейс должен быть на русском языке;
- интерфейс должен быть в лаконичном дизайне и не должен содержать слишком ярких или слишком тёмных цветов;
- символы, значки и текст в интерфейсе должны отображаться корректно и быть понятными и читабельными.

#### Программные:

- Программа должна представлять собой мобильное приложение для ОС Android.

#### Системные требования для смартфона:

- Программа должна работать под операционной системой Android 7.0 и выше.

Предоставление услуг предприятием «СЭМ» включает в себя выполнение запросов заказчика по обслуживанию мостовых сооружений (Рис. 1). Во время этой работы ведется контроль выполнения задач, заключающийся в предоставлении проверяющему лицу медиафайлов в виде набора фото «До» и фото «После».

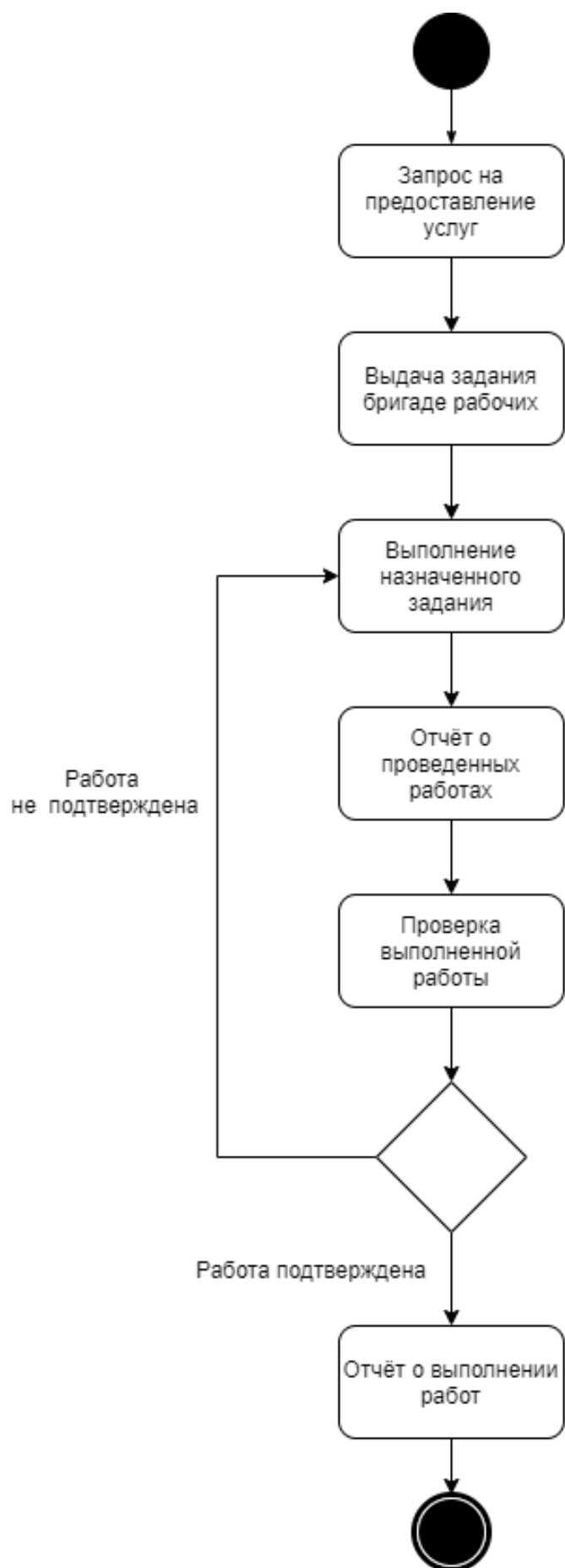


Рисунок 1 –Диаграмма действий работы предприятия «СЭМ»

### 1.3 Моделирование

**Предмет моделирования** – деятельность предприятия «СЭМ» в процессе оказания услуг по обслуживанию искусственных сооружений.

**Область моделирования** – модель охватывает процессы выполнения задания, связанные с обслуживанием мостовых сооружений.

**Цель моделирования** – моделирование всего процесса выполнения задания для определения процессов необходимых для переноса в мобильное приложение.

**Точка зрения** – предприятие.

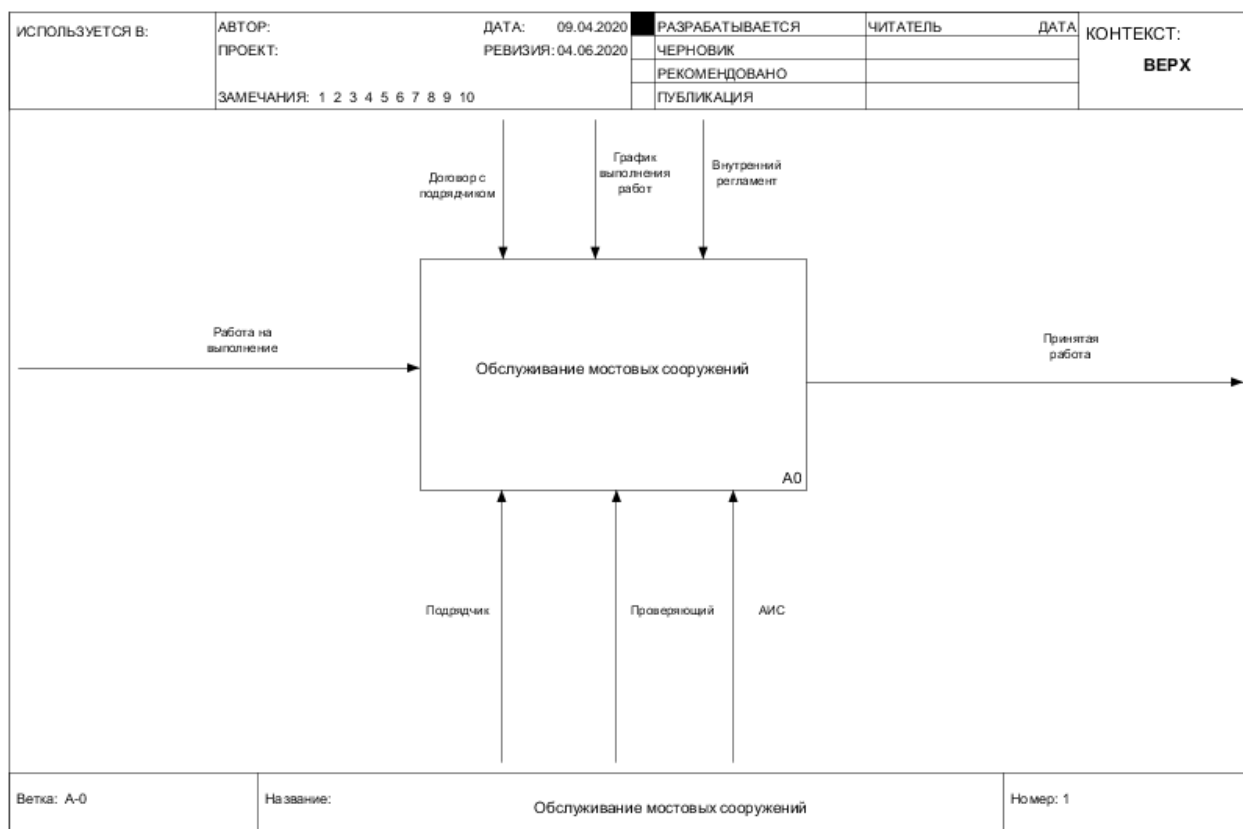
Диаграмма (А-0) является вершиной структуры и представляет собой общее описание системы и ее взаимодействия с внешней средой (Рис. 2).

Основные потоки:

- Входящие потоки:
  - Работа на выполнение.
- Управляющие потоки:
  - Договор с подрядчиком;
  - График выполнения работ;
  - Внутренний регламент.
- Ресурсные потоки:
  - Подрядчик;
  - Проверяющий;
  - АИС (компьютерное приложение, мобильное приложение, БД).
- Выходящие потоки:
  - Принятая работа

Созданная модель описывает деятельность предприятия «СЭМ».

Диаграмма (А-0) является вершиной структуры и представляет собой общее описание системы и ее взаимодействия с внешней средой (Рис. 2). Так как перед нами стоит задача разработки мобильного приложения, являющегося частью АИС, то нам будет интересен этап с рисунка 4 «Выполнение работы», т.к. он включает в себя работу с фотоматериалом и отправку его на проверку, что является одной из основных задач разрабатываемого мобильного приложения.



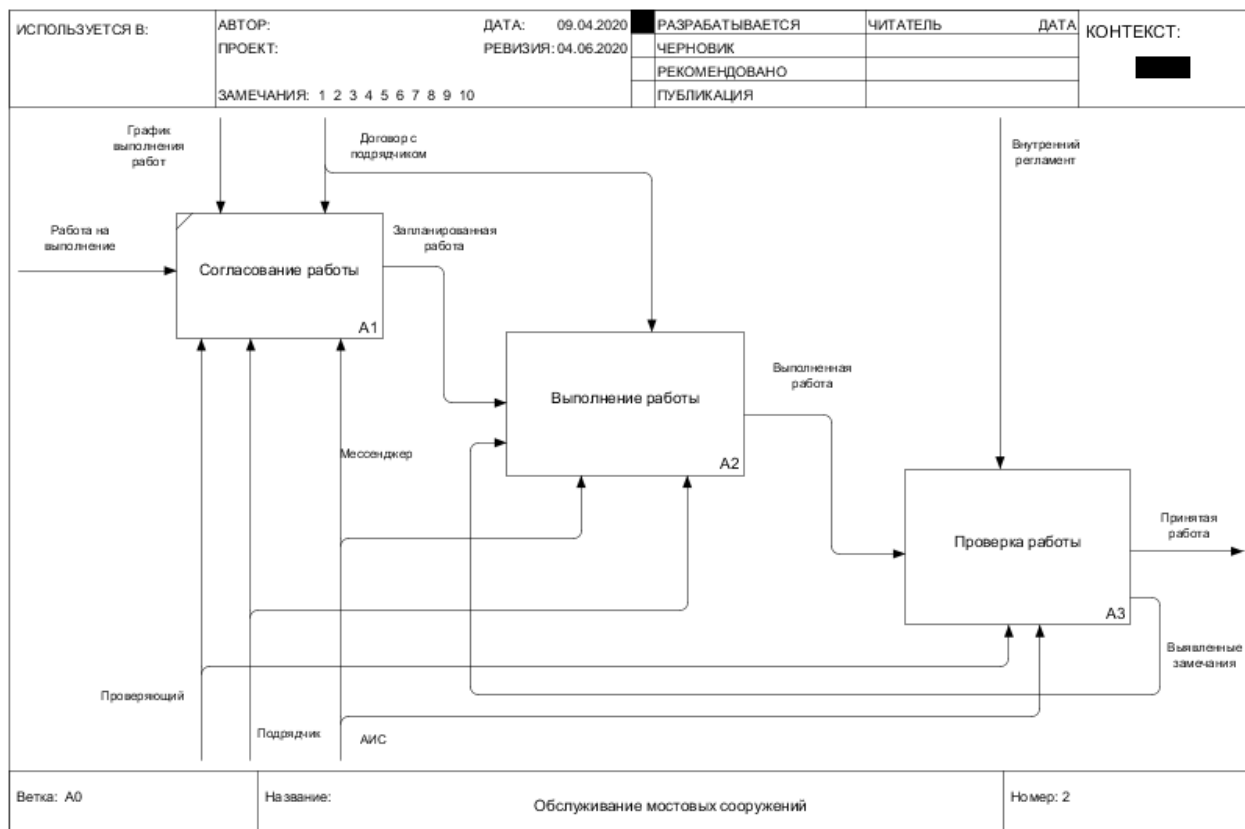
**Рисунок 2 - Контекстная диаграмма процесса**

Входными данными для процесса всегда является работа на выполнение, установленная согласно календарному графику выполнения работ. Также выполнение процесса регулируется договором с подрядчиком и внутренним регламентом проверки выполненных работ. В осуществлении процесса задействованы два лица – подрядчик и проверяющий, взаимодействие которых происходит с помощью АИС.

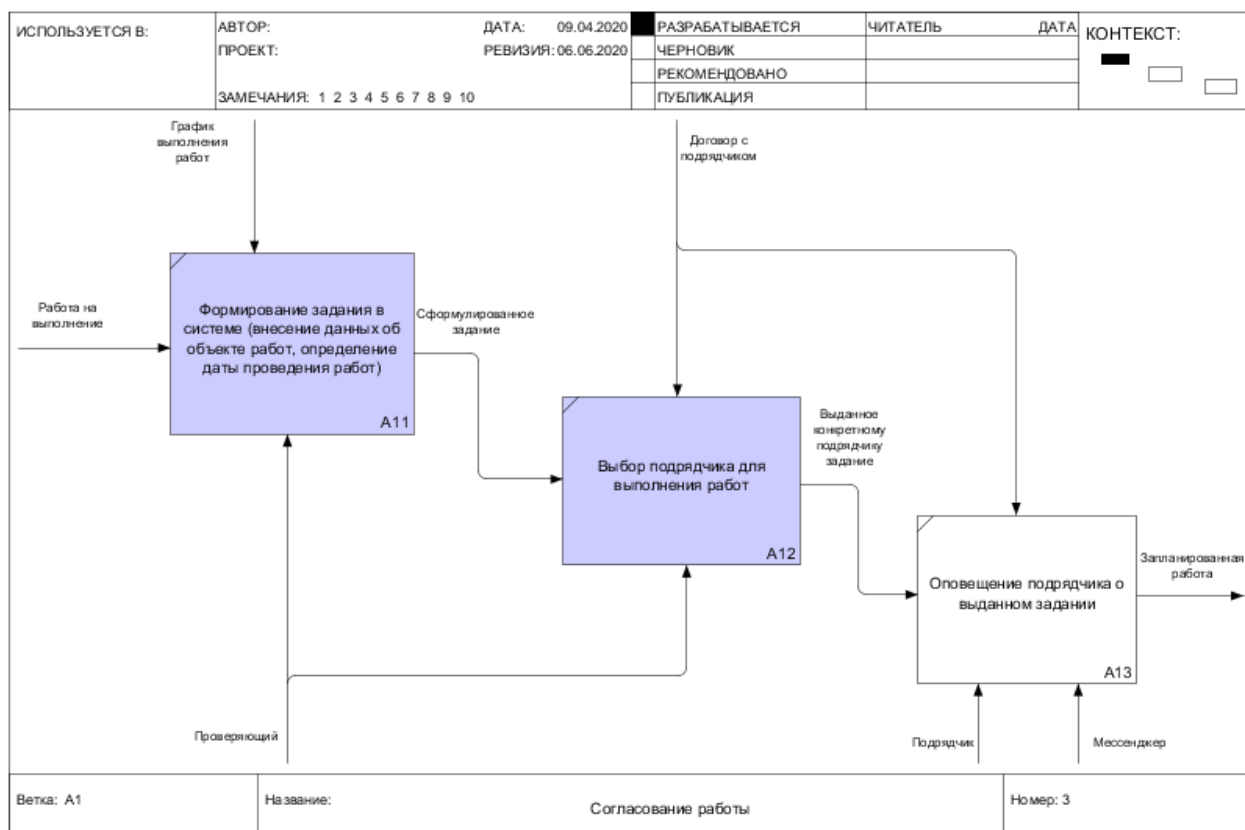
Результатом выполнения процесса является принятая работа.

Работа по обслуживанию конкретного мостового сооружения начинается с согласование данной работы между подрядчиком и компанией «СЭМ», в лице которой на диаграммах будет выступать проверяющий. Так как проектируемой мобильное приложение входит в состав АИС предприятия, то некоторые действия с рисунка 4, подразумевают задействование иной части, данной АИС, а именно компьютерного приложения, данные действия выделены цветом. Таким образом рисунок 4 показывает, что формирование самого задания, его описания и необходимых ему данных, таких как выбор объекта обслуживания и подрядчика должны будут проходить на компьютерной части АИС, с последующим оповещением подрядчика в мобильном приложении о поступившем задании.

Для удобства демонстрации в разрабатываемом мобильном приложении будет добавлен модуль эмулирующий данные этапы и позволяющий создавать «шаблоны» заданий через него.



**Рисунок 3 - Декомпозиция контекстной диаграммы «Обслуживание мостовых сооружений»**



**Рисунок 4 - Декомпозиция контекстной диаграммы «Согласование работ»**

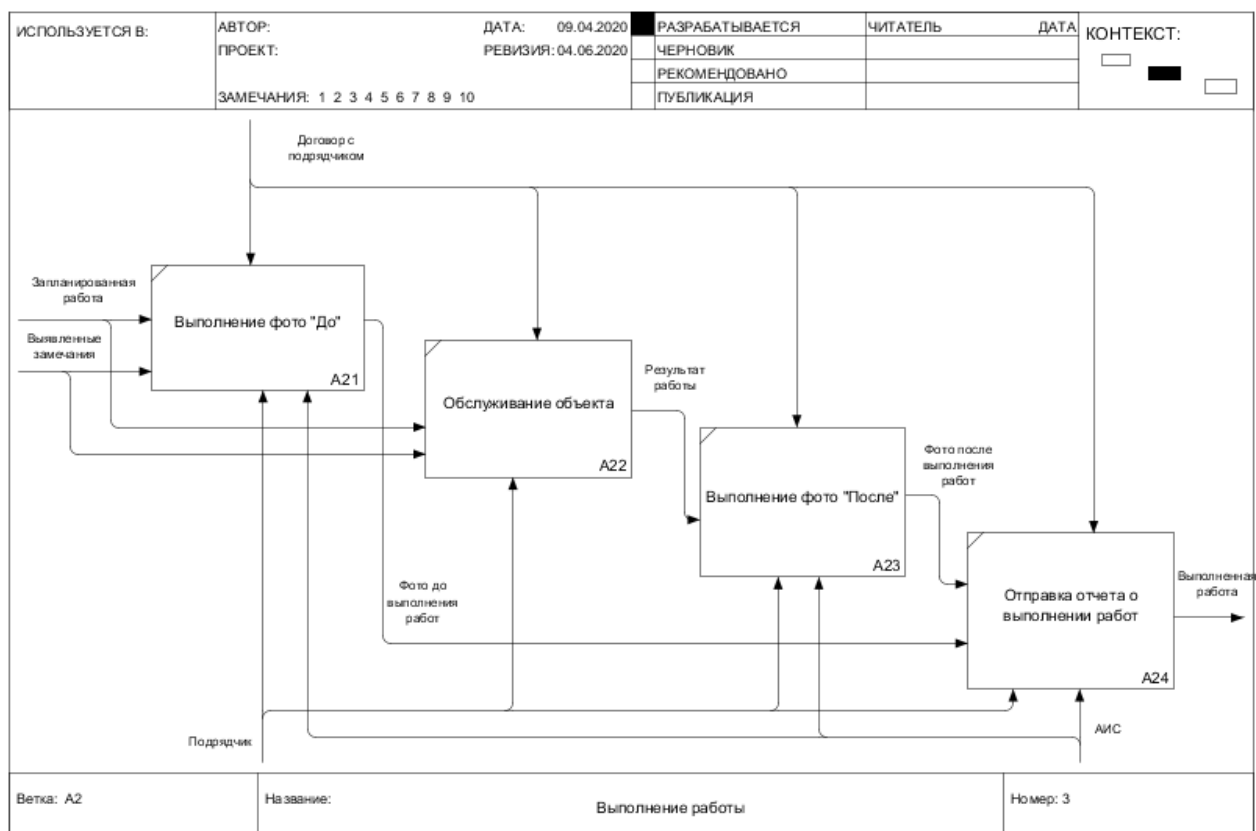


Рисунок 5 - Диаграмма декомпозиции процесса «Выполнение работы»

Перед выполнением работы подрядчик создает фото «До», свидетельствующие о текущем состоянии объекта, а затем выполняет работу.

После выполнения работы подрядчик создает фото «После», которые должны в достаточной мере отражать состояние объекта после выполнения работы.

Все фотоматериалы и пояснения к ним в совокупности составляют отчет о выполнении работы и отправляются по средствам мобильного приложения на проверку.

Декомпозиция контекстной диаграммы «Выполнение работы» представлена на рисунке 5.

В обязанности проверяющего входит проверка качества выполненной работы, а также проверка соответствия полученных фотоматериалов тому мостовому сооружению, на котором было необходимо выполнить работу. Если работа выполнена качественно и на нужном объекте, она считается принятой, а данные о выполнении работы заносятся проверяющим в журнал. Иначе подрядчику отправляются замечания по работе для их исправления.

Диаграмма декомпозиции процесса «Проверка работы» представлена на рисунке 6.

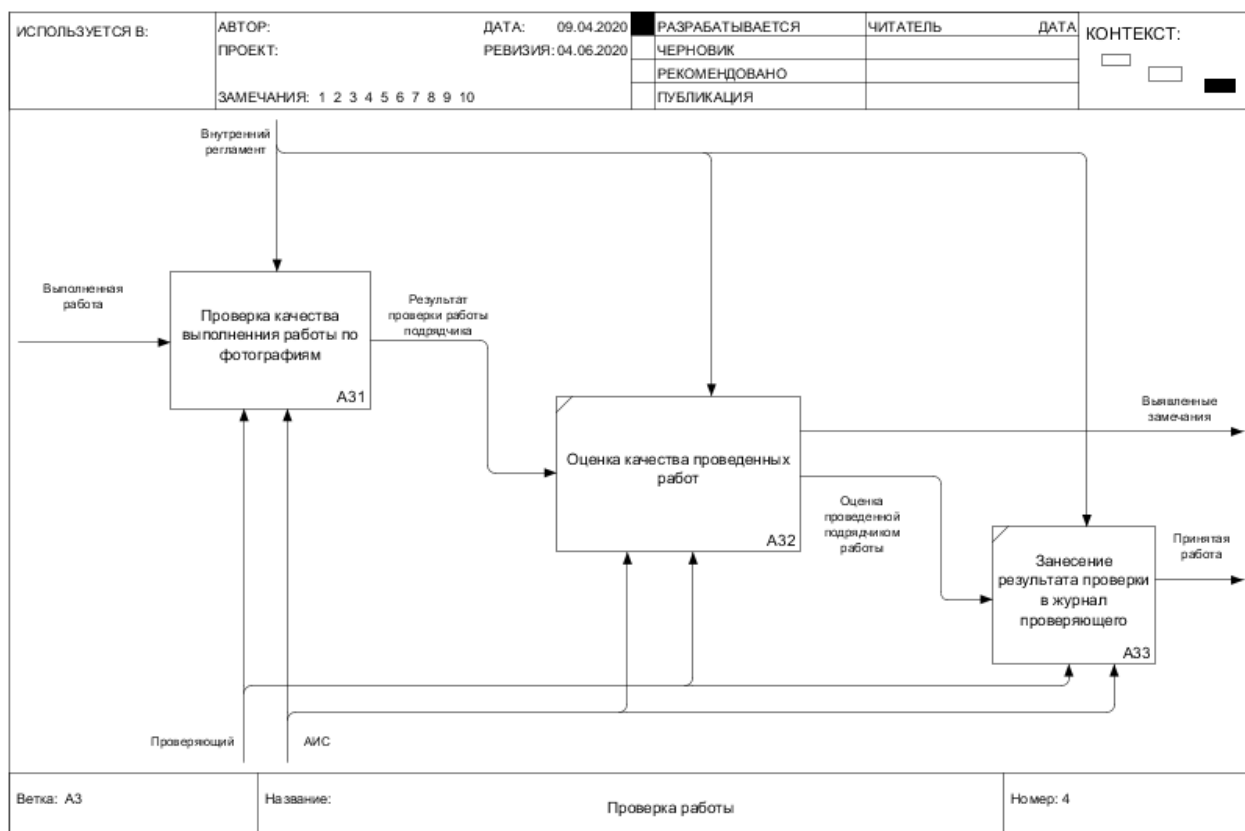


Рисунок 6 - Диаграмма декомпозиции процесса «Проверка работы»

Проверка качества выполненной работы происходит на основе фотографий, полученных от подрядчика. Проверяющий должен сопоставить сооружения на полученных фотографиях с сооружениями обсуживаемыми данным подрядчиком и оценить качество работ по полученному фотоматериалу. При проверке проверяющий структурирует полученный фотоматериал и заносит его в архив.

Диаграмма декомпозиции процесса «Проверка качества выполнения работы по фотографиям» представлена на рисунке 6.





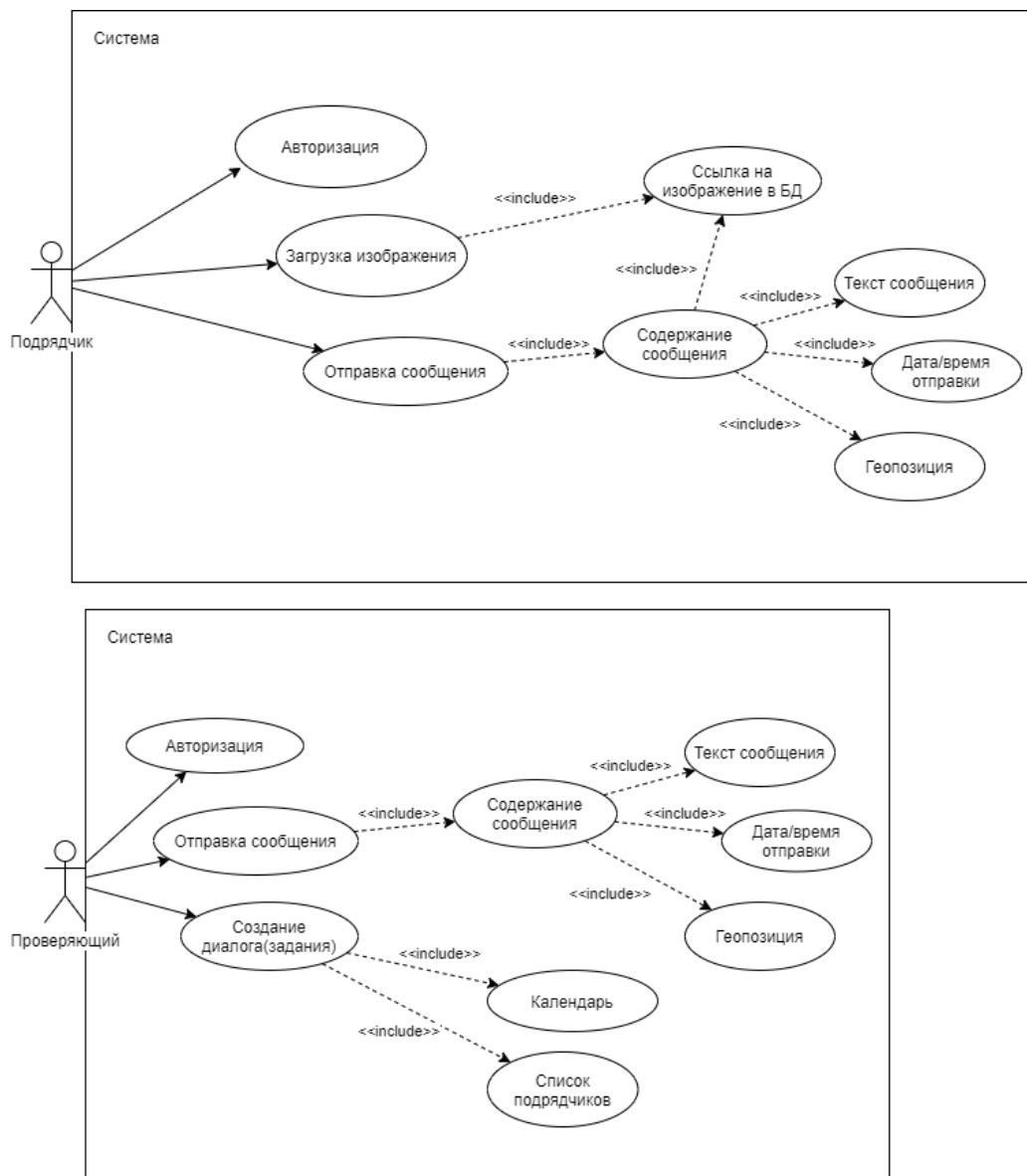


Рисунок 8 – Диаграммы вариантов использования мобильного приложения для ролей «подрядчик» и «проверяющий»

### Спецификация варианта использования «Регистрация пользователя»

**Цель:** описать действия пользователя при регистрации в системе.

**Активные субъекты:** пользователь.

**Краткое описание:** ввод данных, для регистрации пользователя в системе, занесение их в БД.

**Предусловия:** установленное на телефон мобильное приложение.

**Основной сценарий:**

1. Пользователь открывает форму регистрации.
2. Заполняет все необходимые поля личными данными.
3. Программа проверяет правильность введенных данных.
4. Если поля заполнены некорректно, то программа сообщает об ошибке и просит их исправить.

5. Если поля корректно заполнены, то программа формирует профиль пользователя с уникальным идентификатором.
6. Запись о читателе сохраняется в базе данных библиотеки.

#### **Сценарий обработки ошибок:**

Предусловие: на шаге 3 основного сценария происходит ошибка.

##### Ошибка 1

1. Если заполнены не все обязательные поля, то программа требует заполнить всё, что необходимо.
2. Запись не сохраняется.
3. Возврат к шагу 2.

##### Ошибка 2

1. Если поля заполнены неправильно, то программа выводит на экран подсказку и требует заполнить поля корректно.
2. Запись не сохраняется.
3. Возврат к шагу 2.

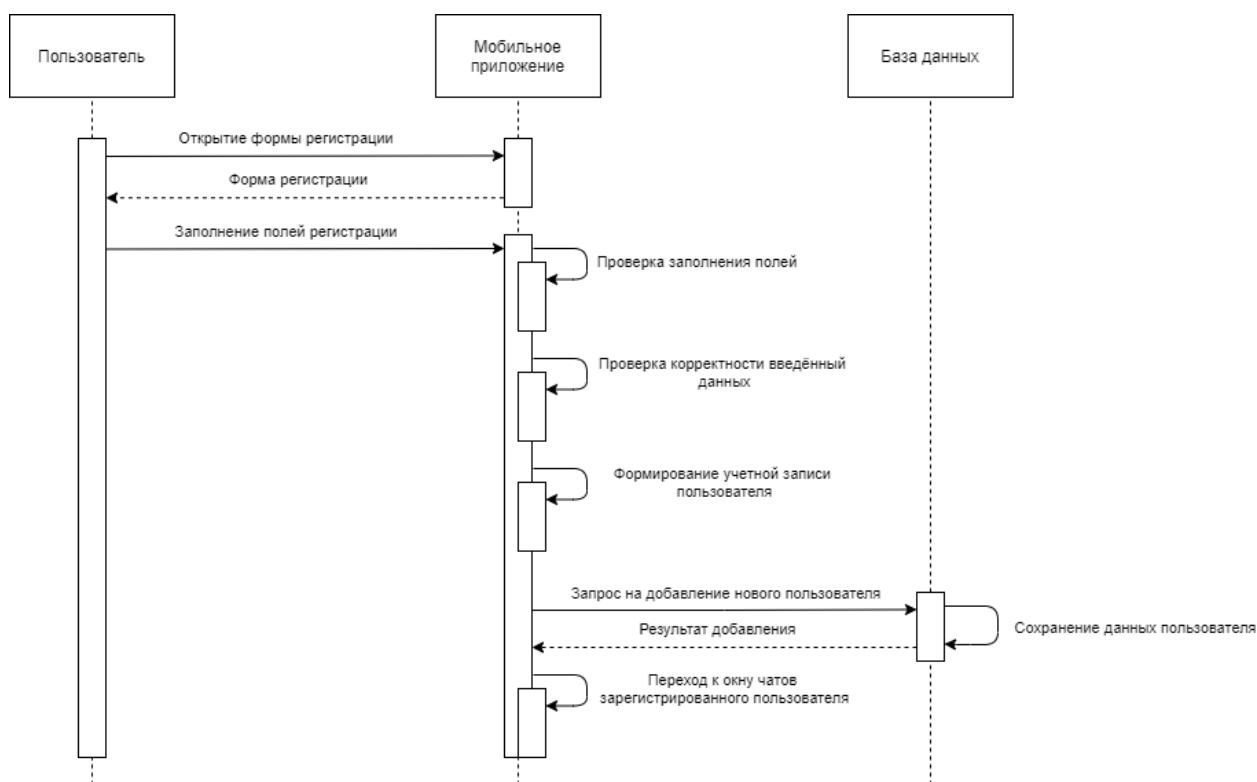


Рисунок 9 – Диаграмма последовательности «Регистрация пользователя»

#### **Спецификация варианта использования «Авторизация пользователя»**

**Цель:** описать действия пользователя при авторизации в системе.

**Активные субъекты:** пользователь.

**Краткое описание:** ввод данных, которые пользователь использовал для регистрации в системе, подтверждение этих данных и предоставление доступа к системе.

**Предусловия:** установленное на телефон мобильное приложение, пользователь зарегистрировано в системе.

**Основной сценарий:**

1. Пользователь открывает форму авторизации.
2. Заполняет все необходимые поля личными данными (логин/пароль).
3. Программа проверяет правильность введенных данных.
4. Если поля заполнены некорректно, то программа сообщает об ошибке и просит их исправить.
5. Если поля корректно заполнены, то программа сверяет введенные данные с БД в поиске совпадений с зарегистрированными пользователями.
6. Если совпадения не найдены, то программа демонстрирует подсказку о проблеме авторизации.
7. Если совпадения найдены, то программа авторизует пользователя и предоставляет доступ к его заданиям (чатам).

**Сценарий обработки ошибок:**

Предусловие: на шаге 3 основного сценария происходит ошибка.

Ошибка 1

1. Если заполнены не все обязательные поля, то программа требует заполнить всё, что необходимо.
2. Пользователь не авторизуется.
3. Возврат к шагу 2.

Ошибка 2

1. Если поля заполнены, но совпадений в БД нет, то программа выводит на экран подсказку и просит проверить правильность заполнения полей.
2. Пользователь не авторизуется.
3. Возврат к шагу 2.

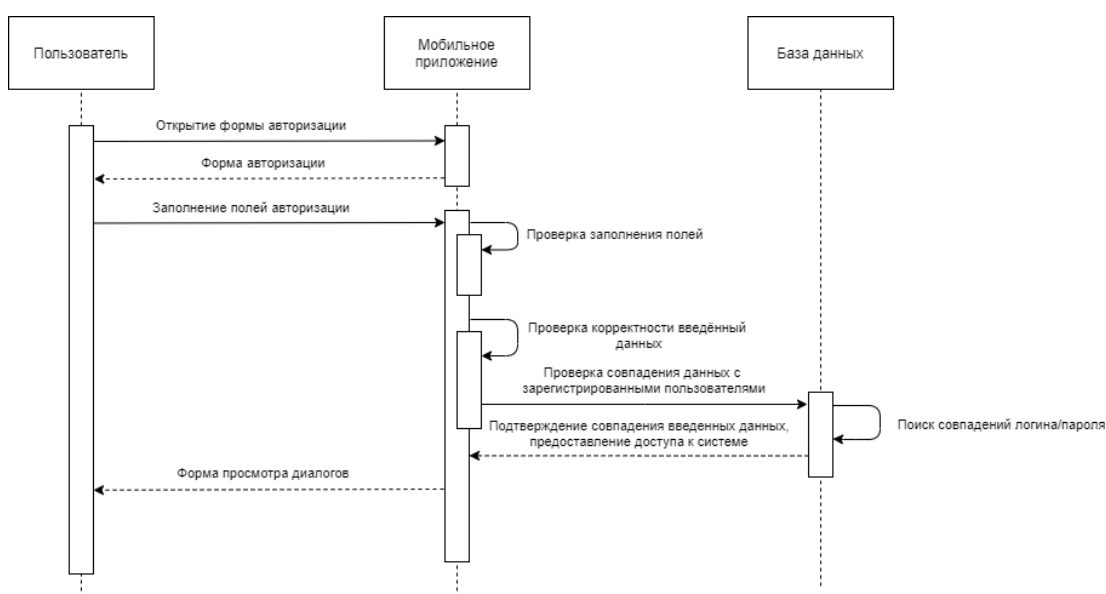


Рисунок 10 – Диаграмма последовательности «Авторизация пользователя»

## Спецификация варианта использования «Отправка сообщения пользователем»

**Цель:** описать действия пользователя при отправке сообщения.

**Активные субъекты:** пользователь.

**Краткое описание:** ввод сообщения пользователем и его отправка.

**Предусловия:** установленное на телефон мобильное приложение, пользователь авторизован в системе.

### Основной сценарий:

1. Пользователь открывает нужный ему чат.
2. Заполняет поле ввода сообщения.
3. Прикрепляет фотоматериал.
4. Программа проверяет наличие введенных данных.
5. Если поле текста не заполнено или фотоматериал не прикреплен, то программа не производит отправку сообщения.
6. Если поле текста заполнено или прикреплен фотоматериал (или заполнено поле текста и прикреплен фотоматериал), то программа записывает время/дату и геопозицию на момент отправки сообщения.
7. Данные о сообщении сохраняются в БД.

### Сценарий обработки ошибок:

Предусловие: на шаге 4 основного сценария происходит ошибка.

#### Ошибка 1

1. Если не заполнено поле текста и не прикреплен фотоматериал, то программа не обрабатывает нажатие на кнопку отправить.
2. Возврат к шагу 2.

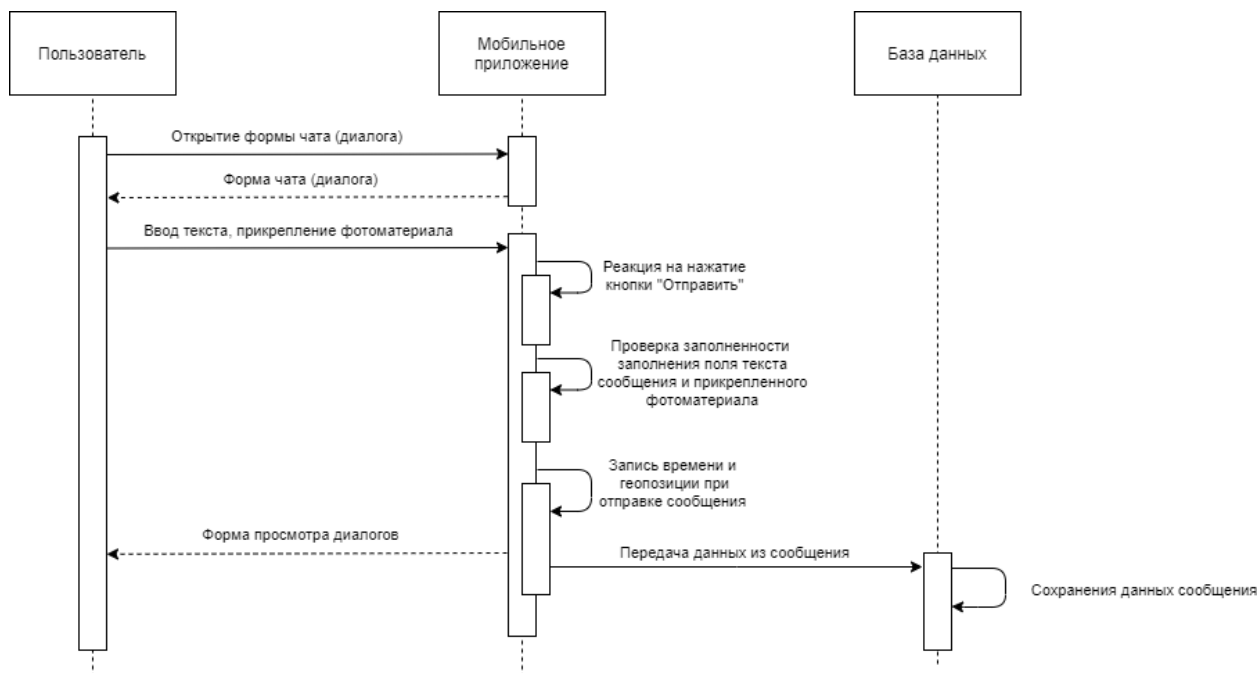


Рисунок 11 – Диаграмма последовательности «Отправление сообщения»

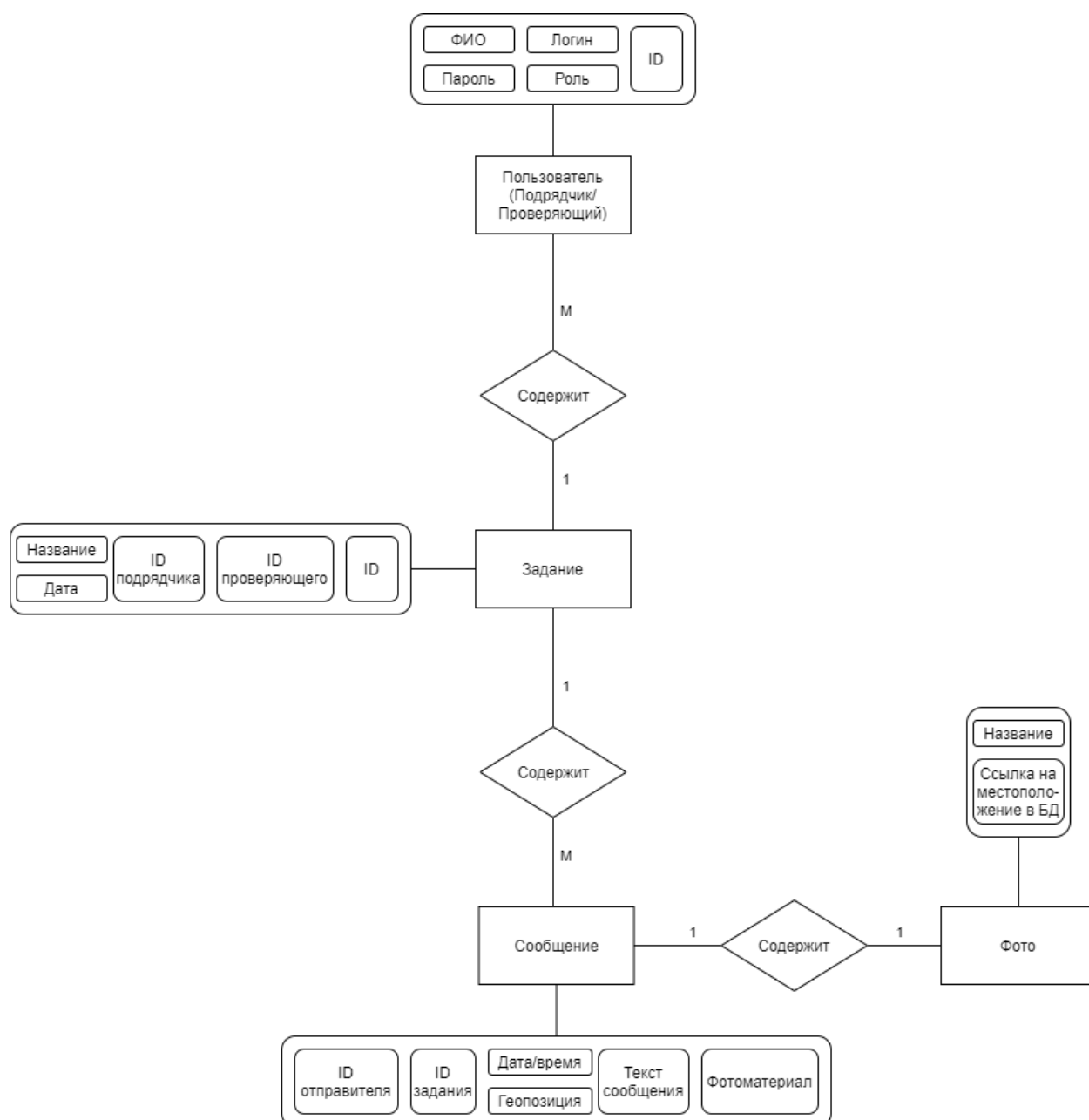


Рисунок 12 – Диаграмма «сущность-связь» по Чену

На рисунке 12 изображена диаграмма сущность-связь по Чену для разрабатываемой системы, в ней: прямоугольниками обозначены активные части системы и её пользователи, овалами – атрибуты, многоугольниками активные действия.

### 1.5 Описание поведения разрабатываемой системы

На рисунке 13 представлено поведение системы при открытии чата пользователем, описанное с помощью диаграммы активности.

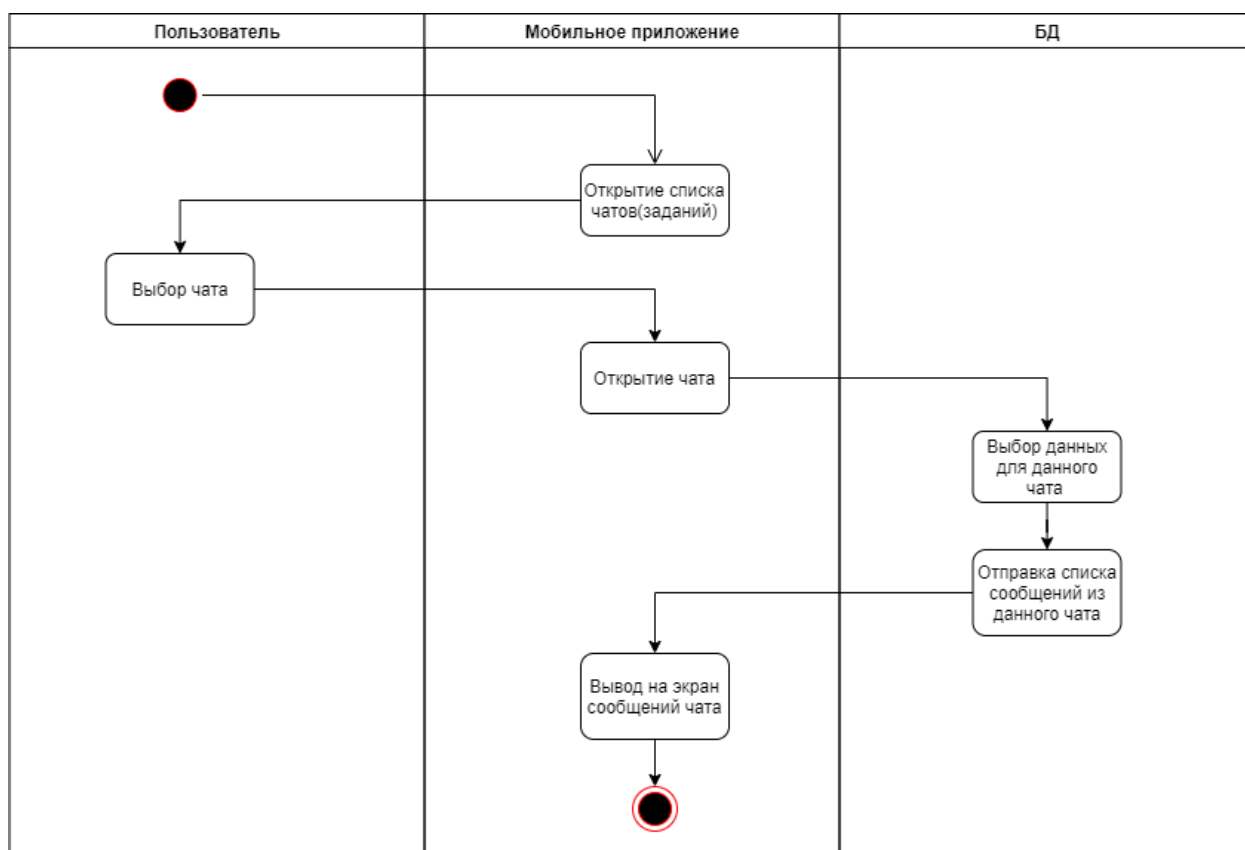


Рисунок 13 – Диаграмма активности открытие чата

## 2 Разработка технического задания

### ВВЕДЕНИЕ

Настоящее техническое задание распространяется на разработку мобильного приложения для автоматизации работы предоставления и обработки данных от подрядчиков «СЭМа».

Автоматизации подлежит способ коммуникации между подрядчиками и проверяющими с целью ведения учета данных о времени и местоположении отправленных сообщений с фотоматериалами, что позволит определять территориальное местоположение сделанных фотоматериалов и в следствии расположение мостового сооружения.

Программный продукт должен быть построен с использованием стандартизованных и эффективно сопровождаемых решений. Программный продукт должен быть реализован как открытая система, и должен допускать наращивание функциональных возможностей.

### 1 Общие сведения

#### 1.1 Наименования и шифры

##### 1.1.1 Полное наименование системы

Мобильное приложение «Служба эксплуатации мостов»

##### 1.1.2 Краткое наименование системы

«СЭМ»

## **1.2. Основания для проведения работ**

### **1.2.1. Основания для разработки автоматизированной системы**

Основанием для исполнения работ по созданию автоматизированной системы, предусмотренных в настоящем ТЗ, является учебный план кафедры «Автоматизированные системы»

### **1.2.2. Основания для разработки документа**

Основанием для разработки настоящего Документа является п.6 к Договору №322.

### **1.2.3. Нормативные документы**

Настоящее Техническое Задание разработано в соответствии с требованиями ГОСТ 34.602-89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы».

При разработке и вводе в эксплуатацию автоматизированной системы Исполнитель должен руководствоваться требованиями следующих нормативных документов Госстандарта:

ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания;

ГОСТ 34.603-92. Информационная технология. Виды испытаний автоматизированных систем;

ГОСТ 19. Единая система программной документации;

РД 50-34.698-90. Методические указания. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.

## **1.3. Наименование организаций – Заказчика и Разработчика**

### **1.3.1. Заказчик**

Заказчик: ИРНИТУ

Адрес фактический: г.Иркутск, ул.Лермонтова, 83

Телефон/Факс +7 (3952) 40-57-46

### **1.3.2. Разработчик**

Разработчик: Мускатин И.А.

Адрес фактический: г.Иркутск, М.Конева

Телефон/Факс: +7 (800) 555-35-35

## **1.4. Плановые сроки начала и окончания работы**

На основании Договора между Заказчиком и Разработчиком устанавливаются сроки от 01.02.2020 г. до 04.06.2020 г.

## **1.5. Источники и порядок финансирования**

Финансирование проекта происходит за счет Заказчика. Порядок финансирования работ определяется условиями Договора №322 от 01.02.2020г.



## **1.6. Порядок оформления и предъявления заказчику результатов работ**

Порядок оформления и предъявления работы Заказчику производится согласно Договору. Приемка системы осуществляется комиссией, созданной Заказчиком.

### **2. Назначение и цели создания системы**

#### **2.1. Назначение системы**

Информационная система ИС «Учёт курсовых» предназначена для автоматизации учебного процесса в институте.

#### **2.2 Цели создания**

##### **2.2.1 Общие цели проекта**

Цели создания и внедрения мобильного приложения «СЭМ» – сокращение ручных операций для определения территориальной принадлежности полученных фотоматериалов, и автоматизированная запись данных о геопозиции и загрузки фотоматериалов и отчетов о выполнении работы подрядчиком.

##### **2.2.2 Цели текущего этапа**

Целью работ по данному этапу проекта является создание первой экспериментальной версии мобильного приложения.

### **3 Характеристика объектов автоматизации**

#### **3.1 Краткие сведения об объекте автоматизации**

Объектом автоматизации является один из процессов работы АИС «СЭМ», задача которого состоит в мониторинге выполнения выданных подрядчиком заданий и сбор соответствующих сведений.

#### **3.2 Сведения об условиях эксплуатации**

Автоматизированная система будет эксплуатироваться на мобильных устройствах операционной системы Android.

## **4 Требования к программе или программному изделию**

### **4.1 Требования к функциональным характеристикам**

Проектируемая программа должна реализовывать следующие задачи:

- Авторизация в мобильном приложении по личному идентификатору (напр. почте, телефону и тд.) и паролю, с определением прав доступа. При авторизации для пользователя определяется его роль, от которой зависит доступные функции приложения, эти данные берутся из БД.
- Регистрация в системе по введенным личным данным, например, ФИО, логин, пароль, название компании подрядчика. Также в другой части комплекса программ АИС «СЭМ» расположенном на компьютерах, администратор может распределять роли доступа пользователям.

- Добавление заданий (чатов), для уровня доступа «проверяющий». Создание чата с названием, характеризующем суть задание, назначенной датой проведения работ и выбранным подрядчиком.
- Возможность коммуникации в чатах с занесением сообщений чата, включающих в себя текст, ссылку на медиафайл, дату/время отправления, геопозицию в БД для хранения и последующей обработки.
- Возможность загружать медиафайлы в чат, а именно изображения, как сделанные с камеры телефона, по средствам произведения снимка через приложение, так и по средствам выбора уже, сделанного фотоматериала с телефона.

#### **4.2 Требования к надежности**

Уровень надежности должен достигаться согласованным применением организационных, организационно-технических мероприятий и программно-аппаратных средств. Надежность должна обеспечиваться за счет:

- применения технических средств, системного и базового программного обеспечения, которые будут соответствовать требованиям решаемых задач;
- своевременного выполнения процессов администрирования системы;
- применение разграничение доступа в приложении;
- применение блокировки некорректных действий пользователей при работе с системой, в частности, проверка некорректности введенных данных;
- предварительного обучения пользователей.

Время восстановления работоспособности ИС при любых сбоях и отказах не должно превышать одного рабочего дня, исключая случаи неисправности серверного оборудования.

#### **4.3 Требования к составу и параметрам технических средств**

Система должна иметь графический пользовательский интерфейс. Интерфейс системы должен быть удобным и простым для пользователя, а также он должен обеспечивать быстрое отображение экранных окон. Навигационные элементы должны быть понятными для пользователя. Все экранные формы пользовательского интерфейса должны быть выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации. Так как планируется использовать данное приложение на смартфонах/планшетах под ОС Android, дополнительные устройства в виде мыши или клавиатуры не требуются.

#### **4.4 Требования к защите информации**

Защита Системы должна обеспечиваться комплексом программно-технических средств и поддерживающих их организационных мер.

Защита Системы должна обеспечиваться на всех технологических этапах обработки информации и во всех режимах функционирования, в том числе при проведении ремонтных и регламентных работ.

Программно-технические средства защиты не должны существенно ухудшать основные функциональные характеристики системы (надежность, быстродействие, возможность изменения конфигурации).

#### **4.5 Требования к информационной и программной совместимости**

Система должна работать под управлением операционных систем Android 7.0 и выше.

#### **4.6 Требования к способам и средствам связи для информационного обмена между компонентами системы**

Информационный обмен между подсистемами должен осуществляться через единое информационное пространство и посредством использования стандартизированных протоколов и форматов обмена данными.

Все компоненты подсистем должны функционировать в пределах единого логического пространства, обеспеченного интегрированными средствами серверов, данных и серверов приложений.

Программное обеспечение системы должно обеспечивать интеграцию и совместимость на информационном уровне с другими системами.

#### **4.7 Требования к квалификации персонала:**

Для работы с ИС пользователи не обязаны обладать какими-либо техническими знаниями, т.к. интерфейс интуитивно понятен.

Техническое обслуживание и администрирование БД должно выполняться специалистами, имеющими соответствующую квалификацию и навыки выполнения работ.

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

Таблица 5 – состав и содержание работ

<b>№</b>	<b>Название этапа</b>	<b>Срок</b>	<b>Отчетность</b>
<b>1</b>	Выработка системных требований (постановка задачи)	11.02.2020- 13.02.2020	Постановка целей, для реализации поставленной задачи
<b>2</b>	Формальное описание требований к программному продукту	21.02.2020- 10.03.2020	Техническое задание
<b>3</b>	Архитектурное и интерфейсное проектирование программного продукта	11.03.20 - 13.04.20	Описание бизнес- процессов
<b>4</b>	Анализ и управление рисками	12.03.20 - 15.03.20	Пояснительная записка к КР
<b>5</b>	Программирование	16.03.20 - 30.04.20	Программный код

Продолжение таблицы 5 на следующей странице

Продолжение таблицы 5

<b>6</b>	Тестирование (автоматизированное)	01.05.20 - 10.05.20	Код, пояснительная записка к КР
<b>7</b>	Разработка документации	12.05.20 - 30.05.20	Документация
<b>8</b>	Планирование и управление процессом разработки	Планирование: 10.02.20-14.02.20 Управление процессом разработки: весь проект	План

По окончании каждого этапа заказчику будет предоставлен отчёт о выполненной работе.

## **6 Порядок контроля и приемки**

### **6.1 Сдача-приемка работ**

Сдача-приёмка работ производится поэтапно, в соответствии с календарным графиком. Основанием для сдачи-приёмки работ служит отчёт о завершении работ по этапу.

### **6.2 Предварительные испытания**

При сдаче-приёмке готовых подсистем информационной системы, создаваемых в рамках настоящей работы, проводятся предварительные испытания с целью подтверждения работоспособности соответствующей подсистемы и соответствия требованиям ТЗ. Предварительные испытания должны проводиться представителями Заказчика по программе тестирования, составленной Исполнителем и согласованной с Заказчиком.

По итогам испытаний по мере необходимости Исполнителем проводится корректировка проектной и эксплуатационной документации и устранение выявленных недостатков программного обеспечения.

### **6.3 Опытная эксплуатация**

Передача системы в опытную эксплуатацию производится приемо-сдаточной комиссией на основании положительного заключения, сделанного по итогам предварительных испытаний.

Опытная эксплуатация должна проводиться на технических средствах Заказчика.

## **7 Требования к программной документации**

Разрабатываемые программные модули должны быть самодокументированы, т.е. тексты программ должны содержать все необходимые комментарии.

В состав сопровождающей документации должны входить:

- Пояснительная записка, содержащая описание разработки.
- Руководство пользователя ИС.

### 3 Архитектурное и интерфейсное проектирование программного продукта

#### 3.1 Описание инструментов ПО

Программа будет реализована на языке программирования Java, в среде разработки Android Studio. Интегрированная среда разработки Android Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения. Помимо стандартного редактора и отладчика, которые существуют в большинстве сред IDE, Android Studio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для упрощения процесса разработки.

База данных будет создана на основе программного продукта Firebase-БД, 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. Другими словами, обновление в Firebase происходят мгновенно, что подходит для реализации приложения основывающегося на передаче сообщений между пользователями. Вместе с хранилищем, Firebase также предоставляет пользовательскую аутентификацию, и поэтому все данные передаются через защищенное соединение SSL. Также Firebase предоставляет доступ к Storage (хранилищу), куда можно сохранять файлы и обращаться к ним по сгенерированной URL-ссылке.

#### 3.2 Логическая модель представления данных

При логическом проектировании строится набор предварительных отношений с указанием первичного ключа для каждого отношения. Составляется список атрибутов, затем эти атрибуты распределяются по отношениям. На рисунке 14 представлена логическая модель представления данных.

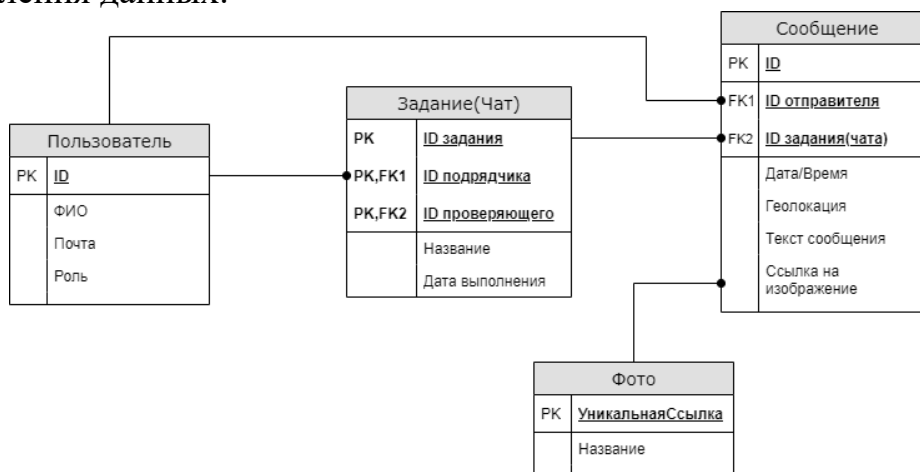


Рисунок 14 - Логическая модель представления данных

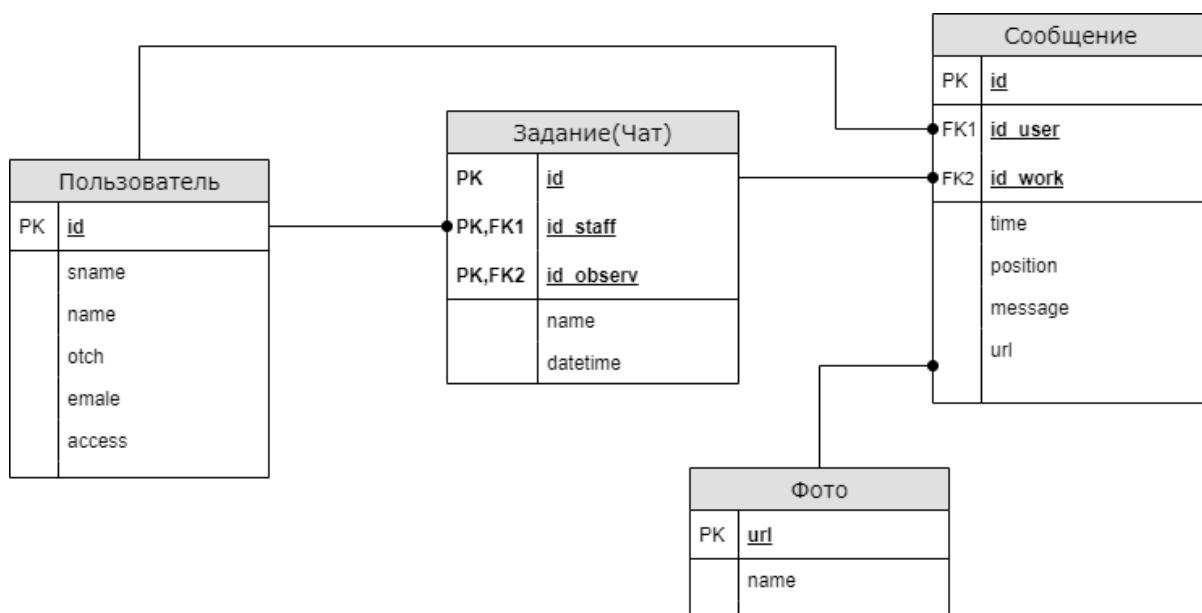


Рисунок 15 – Физическая схема БД

Таблица 6 – Описание сущностей и атрибутов

Имя таблицы	Имя атрибута	Описание
Пользователь	ID	Уникальный идентификатор пользователя
	ФИО	3 поля: Фамилия, Имя, Отчество пользователя
	Почта	Почта пользователя, используемая как логин для входа в систему
	Роль	Роль и уровень доступа пользователя (подрядчик/проверяющий)
Задание	ID	Идентификатор, присваиваемый в заданию при создании.
	ID_подрядчика	Уникальный идентификатор подрядчика, который закреплен к этому заданию
	ID_проверяющего	Уникальный идентификатор проверяющего, который закреплен к этому заданию
	Название	Название задания
	Дата выполнения	Дата, на которую назначено задание
Фото	УникальнаяСсылка	Уникальная ссылка на фотоматериал
	Название	Название фотоматериала

Продолжение таблицы 6 на следующей странице

Продолжение таблицы 6

Сообщение	ID	Идентификатор сообщения
	ID_отправителя	Идентификатор отправителя данного сообщения
	ID_задания	Идентификатор задания к которому относится данное сообщение
	Дата/Время	Время и дата отправки сообщения
	Геолокация	Координаты местоположения пользователя во время отправки сообщения
	Текст сообщения	Текст отправленный пользователем
	Ссылка на изображение	Ссылка на расположение изображения в БД

Таблица 7 – Соответствие между объектами логического и физического уровней

Сущность	Описание	Атрибут	Поле	Тип
Пользователь	Содержит данные о пользователях	ID	id	String
		ФИО	name/sname/otch	String
		Почта	emale	String
		Роль	access	String
Задание	Содержит данные о заданиях	ID	id	String
		ID_подрядчика	id_staff	String
		ID_подрядчика	id_observ	String
		Название	name	String
		Дата выполнения	datetime	String
Фото	Содержит данные о фото	УникальнаяСсылка	url	Uri
		Название	name	String
Сообщение	Содержит данные о напоминаниях о событиях	ID	id	String
		ID_отправителя	id_user	String
		ID_задания	id_work	String
		Дата/Время	time	String
		Геолокация	position	String
		Текст сообщения	message	String
		Ссылка на изображение	url	Uri

### 3.3 Архитектура ПО (Функциональная модель)

На рисунке 16 с помощью функциональной модели представлена архитектура разрабатываемой системы.

Данная модель представляет собой набор функций, определяющих преобразования, осуществляемые системой при выполнении своего назначения.

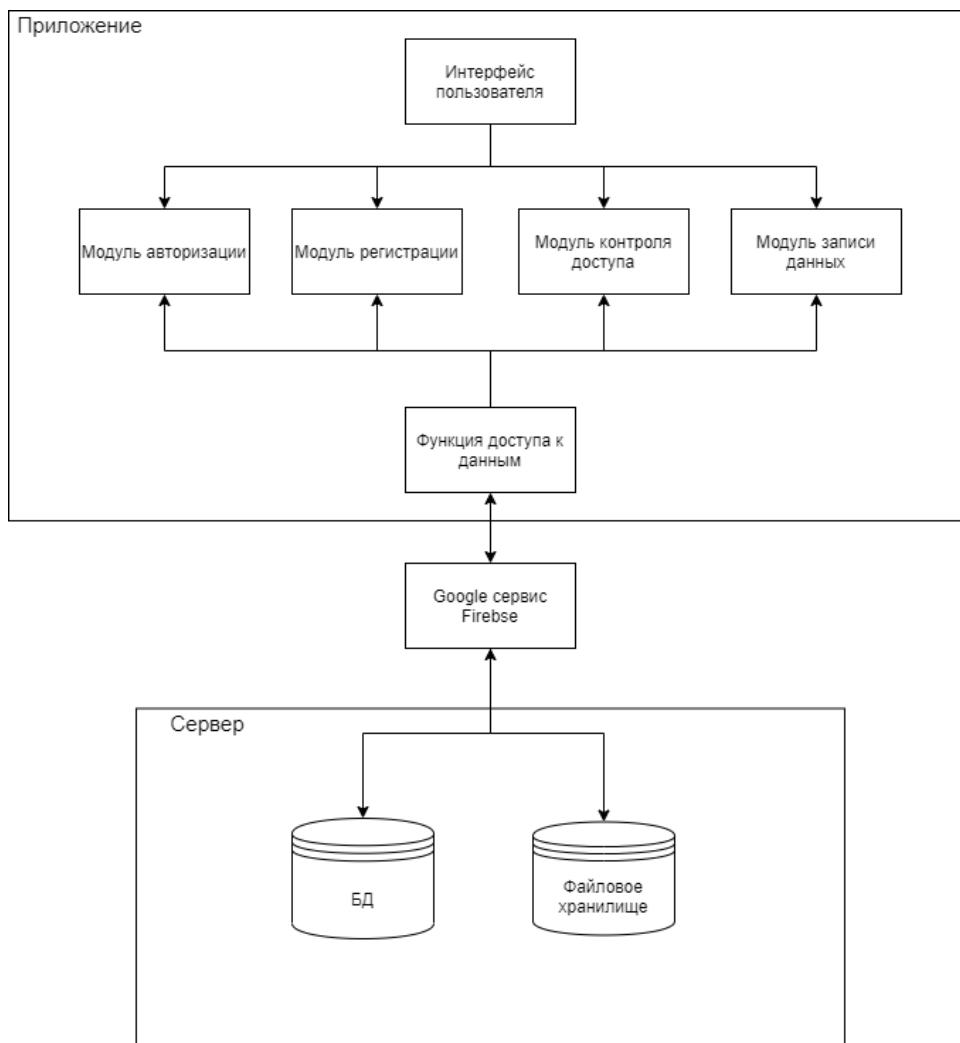


Рисунок 16 – Функциональная модель

Как видно на рисунке, разрабатываемая система будет иметь двухуровневую архитектуру, то есть иметь подуровни в виде приложения и сервера. «Посредником» между приложением и базой данных будет сервис Google, предоставляющий возможность соединения приложения с базой данных и файловым хранилищем. Данные между приложением и базой будут синхронизированы с помощью модуля «функции доступа к данным».

### 3.4 Интерфейсные классы программы

Разрабатываемая система будет иметь двухуровневую архитектуру, то есть иметь подуровни в виде приложения и базы данных. Данные между приложением и базой будут синхронизированы с помощью модуля «функции доступа к данным».



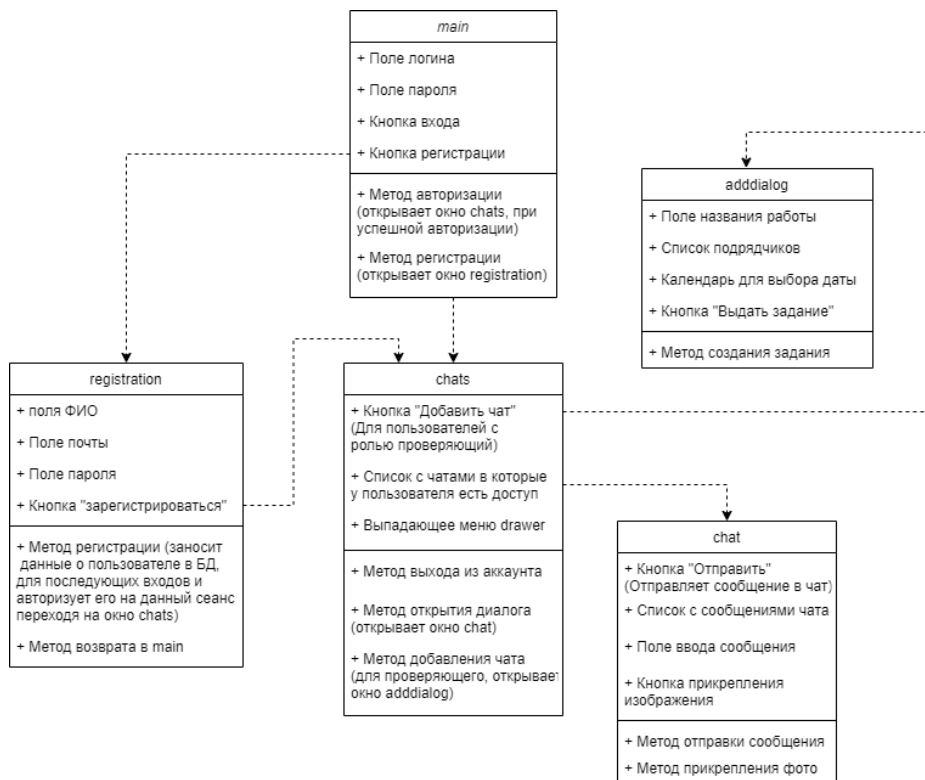


Рисунок 17 – Диаграмма интерфейсных классов

Описание интерфейсных форм приложения:

- **main** – начальная форма приложения, имеет 2 поля логин/пароль, для осуществления авторизации, и две кнопки «войти» и «зарегистрироваться», также имеет слушателя, который при условии запуска приложения уже авторизованным пользователем сразу переносит его на форму chats.
- **registration** – форма на которой осуществляется добавление пользователя в систему, имеет поля для заполнения личной информацией (напр. ФИО, почта, номер телефона, пароль и т.д.), имеет кнопку регистрации, которая при верном заполнении полей регистрирует пользователя в системе и добавляет его профиль в БД. Имеет аналогичного слушателя, который при успешной регистрации авторизует клиента в системе под только что заполненными данными.
- **chats** – форма на которой находятся список всех заданий (чатов), к которым имеет доступ пользователь, элементы списка активные и при нажатии на элемент списка он открывает соответствующий чат в форме chat, при нажатии также срабатывает слушатель определяющий геопозицию пользователя. Форма содержит выдвижное меню содержащее полезные ссылки (напр. «О программе», «Сообщить о проблеме» и т.д., а также имеет клавишу выхода из аккаунта). Если пользователь является проверяющим, то у него также имеется клавиша добавления новых чатов (заданий), открывающая форму adddialog.

- **adddialog** – форма добавления чатов, содержащая в себе виджет календаря, при выборе даты на котором определяется дата выполнения назначенной работы, также форма имеет поля для написания названия работы и список с подрядчиками. Нажимая на кнопку выдать задание формируется объект с полями: идентификатор подрядчика, проверяющего, идентификатор задания, дата выполнения задания и название задания, далее эти поля записываются в соответствующую таблицу в БД.
- **Chat** – форма для осуществления диалога между подрядчиком и проверяющим, имеет возможность написания сообщения и прикрепления к нему медиафайлов (фотографий), также элементом списка выводится на экран уже написанные ранее сообщения. При нажатии на кнопку отправки сообщения формируется объект с данными о сообщении: кто отправил, в каком чате, во сколько, откуда, какое содержание, в последствии поля этого объекта записываются в таблицу БД – «сообщения».

### 3.5 Проектирование макетов интерфейса

Стартовое окно приложения – main, как описано выше, включает в себя функцию авторизации и возможность перехода к регистрации пользователя в системе. В центре формы расположен логотип организации поля email и password являются изменяемые и предназначены для записи в них почты и пароля, соответственно, поле password имеет свойство, скрывающее написанный ранее символ и заменяющий его точкой, для повышения безопасности учетной записи пользователя (см. Рис. 18).

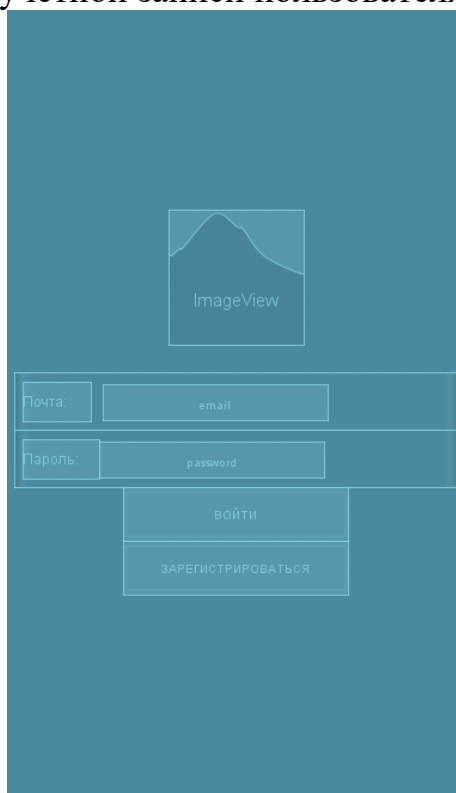


Рисунок 18 – Макет формы main

На рисунке 19 представлен макет формы registration, имеющей 5 полей для личных данных и кнопку регистрации. Также форма имеет toolbar, в котором располагается кнопка «назад», для возвращения на форму main, если это понадобится пользователю.

На рисунке 20 изображена макет формы chats, на нем находятся список listview, который будет включать все доступные пользователю чаты, для пользователя с ролью «проверяющий» становится видимым фрагмент макета «добавить задание» при нажатии на которое будет вызываться форма adddialog, также на форме будет присутствовать боковое меню макет которого можно увидеть на рисунке 21, а так как у listview есть возможность использовать объекты customview в качестве макета отображения элементов своего списка, то макет элемента списка listview представлен на рисунке 22, на котором TextView располагающийся в середине будет являться названием диалога, а располагающийся в правом углу будет информировать о том на какую дату назначено данное задание.

The image shows a vertical UI mockup for a registration form. At the top is a light blue toolbar containing a back arrow icon and the text "\_toolbar". Below the toolbar are five input fields, each with a label: "sname", "name", "otch", "email", and "password". These fields are arranged vertically and have a light blue background with a darker blue border. Below the input fields is a button with the text "ЗАРЕГИСТРИРОВАТЬСЯ" in white capital letters on a dark blue background. The bottom half of the mockup is a large, solid dark blue rectangle, likely representing a listview or a content area.

Рисунок 19 – Макет формы registration



Рисунок 20 – Макет формы chats

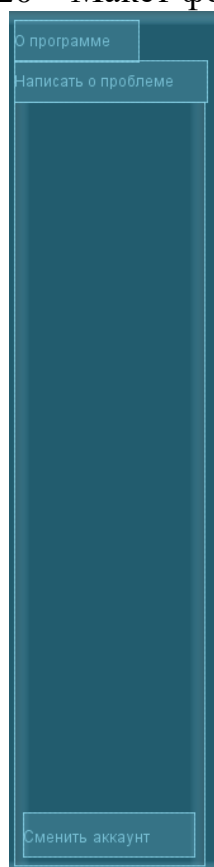


Рисунок 21 – Макет бокового меню формы chats

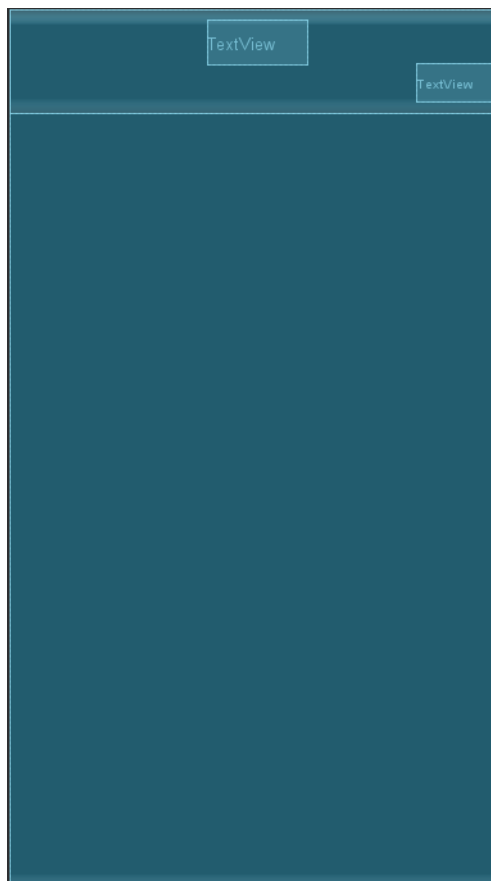


Рисунок 22 – Макет отображения элемента списка listview формы chats

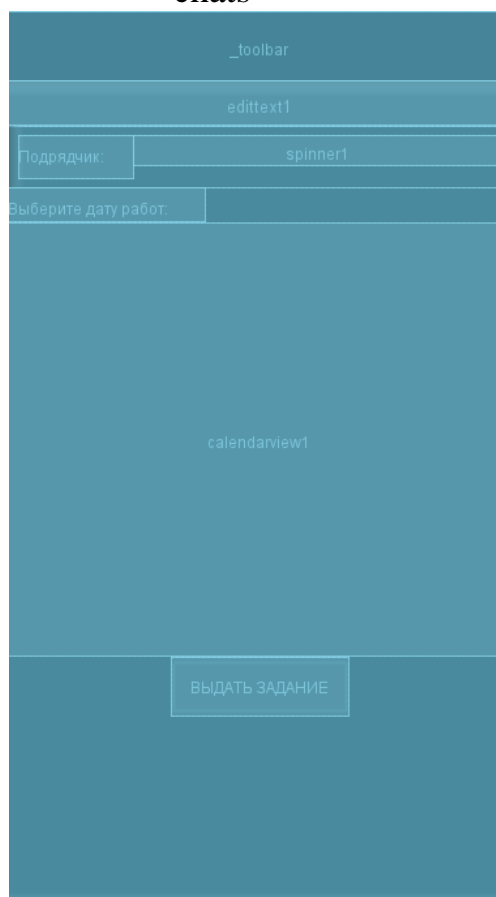


Рисунок 23 – Макет формы adddialog

На рисунке 23 изображен макет формы adddialog, на котором будет расположен widget – calendarview показывающий календарь, edittext является полем для введения названия задачи, spinner является способом отображения списка подрядчиков.



Рисунок 24 – Макет формы chat

На рисунке 24 изображен макет формы chat, отвечающий за отображение чата подрядчика и проверяющего по конкретному заданию. Поле message – поле ввода сообщения, изображение левее кнопки «Отправить» – кнопка для прикрепления фотоматериалов, listview – это список отображающий сообщения данного чата, в toolbar находится кнопка возврата к форме chats.

#### **4 Планирование и управление проектом**

##### **Документация**

В ходе проекта необходимо разработать следующие виды документов: техническое задание (ТЗ), документацию по проведенному маркетинговому анализу.

##### **Характеристика работ проекта**

###### **1. Составление документации**

Составление технического задания, которое оговаривает набор требований к мобильному приложению и утверждено заказчиком и исполнителем.

Также к документам можно отнести разработанный дизайн сайта и техническую документацию проекта.

## 2. Назначение руководителя и работа с персоналом

Руководитель проекта (менеджер проекта) - лицо, которое несет ответственность за проект. Функции руководителя проекта:

- Решение проблем, возникающих в проекте. Как тех, которые исходят от команды проекта, так и тех, которые могут возникнуть при управлении заинтересованными сторонами проекта.
- Определение ролей и обязанностей в проекте для всех членов команды проекта.
- Разработка, утверждение и своевременное обновление плана проекта.
- Мониторинг хода исполнения проекта, прогнозирование отклонений и принятие своевременных мер по их устранению.
- Координация коммуникаций между всеми участниками проекта и его заинтересованными сторонами.
- Контроль изменений в проекте.
- Проведение анализа эффективности этапов проекта и проведение после проектного анализа.

Под персоналом нами понимается проектная команда, которая должна состоять из людей с навыками, отвечающими требованиям выполняемой работы.

## 3. Работа с программным обеспечением

На данном этапе будет проводиться работы, связанные непосредственной разработкой мобильного приложения, это его программирование.

## 4. Работа с данными

Данный этап характеризуется работой с БД, это анализ структуры БД предприятия, поиски способов интеграции работы мобильного приложения на основе БД, при невозможности интеграции, предложение новой структуры БД с переносом данных из уже существующей.

## 5. Ввод в эксплуатацию

На данном этапе проводится тестирование приложения согласно выбранной стратегии тестирования (которая согласуется на этапе разработки документации), составляется отчетная документация по выполненному проекту, осуществляются распространение приложения среди сотрудников.

## 4.1 Иерархическая структура проекта

Были определены следующие задачи функционального процесса:

1. Анализ существующих решений;
2. Сбор и анализ требований заказчика;
3. Составление и согласование ТЗ;
4. Разработка дизайна;
5. Программирование;
6. Связывание БД с приложением;
7. Согласование выполненных этапов с заказчиком;
8. Исправление замечаний заказчика;
9. Подключение БД к серверу;
10. Тестирование;
11. Внедрение;
12. Написание технической документации.

## 4.2 Сетевая модель проекта

### 4.2.1 Управление временем проекта

Сетевой график – это граф, вершины которого отображают состояния или события некоторого объекта, а дуги – работы, ведущиеся на этом объекте.

Построением сетевого графика обеспечивается решение многих задач управления проектами, начиная от определения продолжительности проекта по известным (предполагаемым) величинам продолжительности работ и заканчивая разработкой управления стоимостью проекта, которая должна предусматривать распределение ресурсов во времени.

Введём буквенное обозначение работ. Весь перечень представлен в таблице 8, а сетевой график представлен на рисунке 25.

Таблица 8 – Последовательность выполнения работ

№	Мероприятие	Продолжительность (дней)	Предшествующие задания
1	А. Анализ существующих решений;	2	-
2	Б. Сбор и анализ требований заказчика;	9	А
3	В. Составление и согласование ТЗ;	5	Б
4	Г. Разработка дизайна;	5	В
5	Д. Программирование;	14	Г
6	Е. Связывание БД с приложением;	6	В
7	Ж. Согласование выполненных этапов с заказчиком;	7	Б
8	З. Исправление замечаний заказчика;	5	Е
9	И. Подключение БД к серверу;	4	Д, Е
10	К. Тестирование;	5	В
11	Л. Внедрение;	5	З, К, И
12	М. Написание технической документации.	7	Л



#### 4.2.2 Учет критичности

«Критический путь» проекта – это наиболее длительная, непрерывная (неразрывная) цепь критических действий (работ) от начала и до конца проекта с предельно наименьшей продолжительностью.

Анализ методом «критического пути» заключается в использовании сетевых графиков для определения «критических» работ (действий) проекта согласно определенным условиям критичности, установления «критического пути» проекта как непрерывной последовательности всех критических работ, а также и выявления резервов времени, которые характеризуют степень управляемости («гибкости или подвижности») этих действий в составе проекта.

Резерв времени события – промежуток времени, на который может быть отсрочено наступление этого события без нарушения сроков завершения проекта в целом. Резерв времени определяется как разность между поздним и ранним сроками начала и окончания работы. Принято рассчитывать суммарный, свободный и независимый резервы времени.

Составим таблицу резервов времени (таблица 9).

Таблица 9 – Резервы времени

Действие	Продолжительность	Самое раннее время начала	Самое позднее время начала	Самое раннее время окончания	Самое позднее время окончания	Суммарный резерв времени	Свободный резерв времени	Независимый резерв времени
						6-3-2	5-3-2	5-4-2
№	№2	№3	№4	№5	№6	№7	№8	№9
А	2	0	0	2	2	= 0	= 0	= 0
Б	9	2	2	11	11	= 0	= 0	= 0
В	5	11	11	16	16	= 0	= 0	= 0
Г	5	16	16	21	21	= 0	= 0	= 0
Д	14	21	21	35	35	= 0	= 0	= 0
Е	6	16	16	35	35	= 13	= 13	= 13
Ж	7	2	2	9	34	= 25	= 0	= 0
З	5	9	34	39	39	= 25	= 25	= 0
И	4	35	35	39	39	= 0	= 0	= 0
К	5	16	16	39	39	= 18	= 18	= 18
Л	5	39	39	44	44	= 0	= 0	= 0
М	7	44	44	51	51	= 0	= 0	= 0

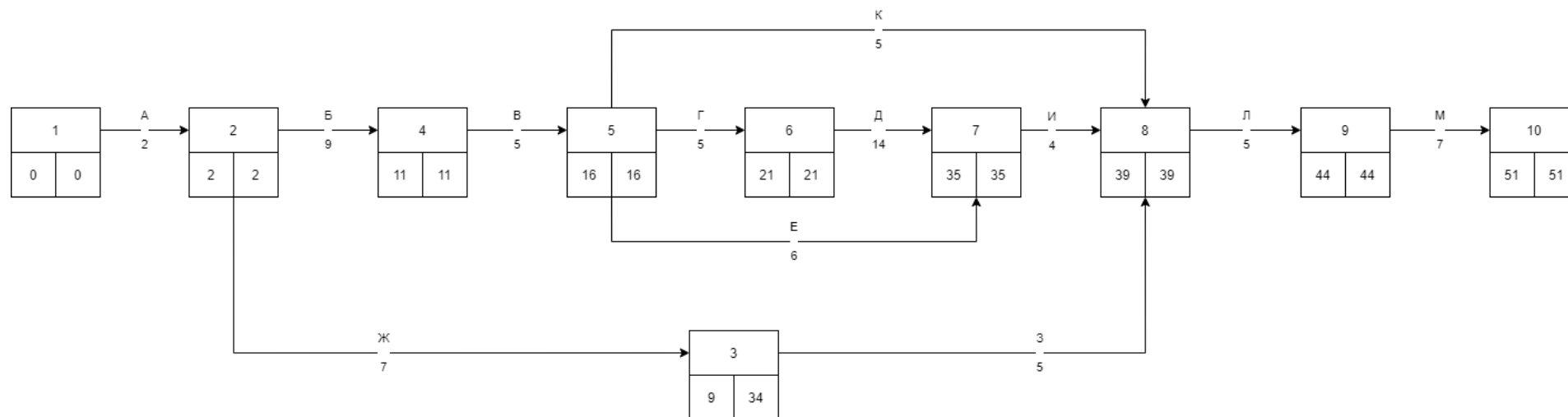


Рисунок 25 – Сетевой график проекта

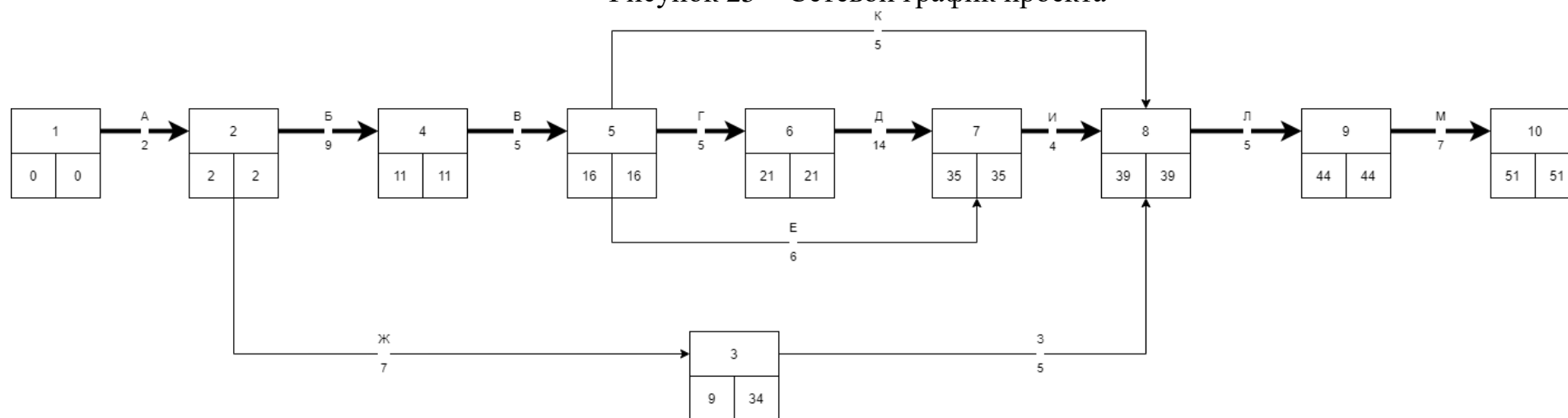


Рисунок 26 – Сетевой график с указанием «критического пути» проекта

### 4.3 Планирование трудовых ресурсов проекта

В качестве ресурсных диаграмм выступает инструментарий, называемый «графиком Ганта» или «ленточной диаграммой Ганта». Это тип гистограммы в виде столбчатой диаграммы, который используется для представления работ проектов зависимости от времени, представленной в виде шкалы. Является одним из популярных методов сетевого планирования ресурсов. Эти диаграммы широко используются в управлении проектами, поскольку удобно отражают не только работы, но и ресурсы, привязанные во времени к этим работам.

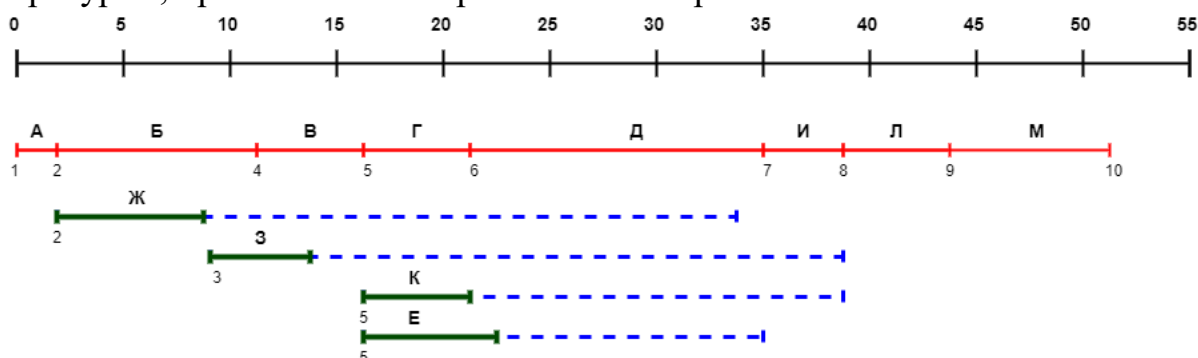


Рисунок 27 – Исходный график Ганта

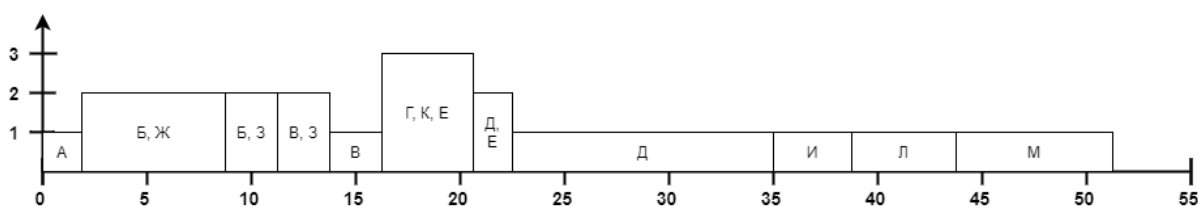


Рисунок 28 – Исходная суммарная потребность в рабочей силе проекта

Сгладить ресурсы получилось на этапах К, Е. Потребности в пересчете периодов работ и продолжительности проекта не потребовалось так как критический путь не изменился. Этапы располагаться параллельно с критическим путем.

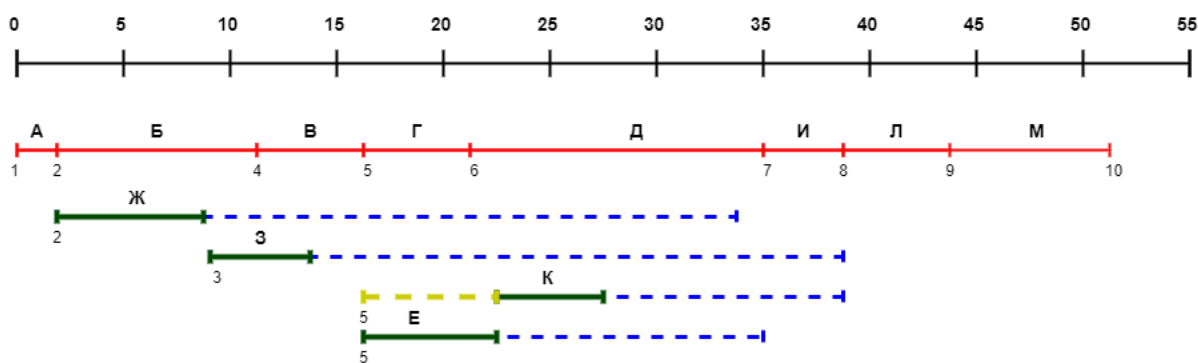


Рисунок 29 – Скорректированный график Ганта

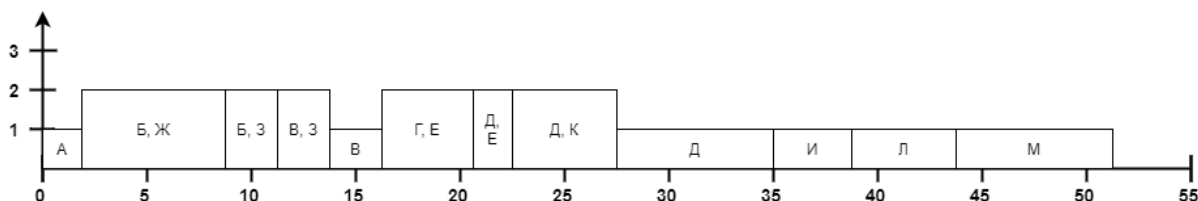


Рисунок 30 – Скорректированная суммарная потребность в рабочей силе проекта

#### 4.4 Учет неопределенностей проекта методом «оценки и пересмотра планов (ПЕРТ)»

Анализ проекта методом «оценки и пересмотра планов» - ПЕРТ имеет цель – прогноз продолжительности проекта с учетом возможного диапазона вероятностного изменения временных характеристик действий проекта.

Вероятностные характеристики действий определяются как средневзвешенные показатели: наиболее вероятной (НВ), максимальной возможной (пессимистической, П) и минимально возможной (оптимистической, О) оценок.

Ожидаемая средняя продолжительность проекта определяется суммой ожидаемых средних продолжительностей всех критических действий в составе проекта с учетом возможного вероятностного разброса их значений, оцениваемых величиной среднеквадратического отклонения.

Формула для подсчета ожидаемой продолжительности:

$$E = (O + П + 4 \times НВ) / 6$$

Формула для подсчета СКО:

$$\sigma = (П - O) / 6$$

Учет неопределенностей проекта позволяет оценить вероятность превышения продолжительностью проекта установленных ограничений по времени как расчетную гарантию завершения проекта в срок.

Таблица 10 – Учет неопределенностей проекта методом ПЕРТ

Действие	Очередность	Оценочная продолжительность (дней)			Ожидаемая продолжительность (дней)	Среднеквадратическое отклонение
		(О)	(НВ)	(П)		
А.	-	2	2	3	2,17	0,17
Б.	А	9	10	12	10,17	0,5
В.	Б	5	6	7	6	0,33
Г.	В	5	6	7	6	0,33
Д.	Г	14	16	20	16,33	1
Е.	В	6	7	9	7,17	0,5
Ж.	Б	7	8	11	8,33	0,67
З.	Е	5	5	6	5,17	0,17
И.	Д, Е	4	6	7	5,83	0,5
К.	В	5	7	8	6,83	0,5
Л.	З, К, И	5	5	6	5,17	0,17
М.	Л	7	9	10	8,83	0,5

Общая ожидаемая продолжительность проекта:

$$E = E_A + E_B + E_V + E_G + E_D + E_I + E_L + E_M = 60,5 \text{ дней}$$

Среднеквадратическое отклонение продолжительности проекта:

$$\sigma = \sqrt{\sigma_A^2 + \sigma_B^2 + \sigma_V^2 + \sigma_G^2 + \sigma_D^2 + \sigma_I^2 + \sigma_L^2 + \sigma_M^2} = 1,42 \text{ дня}$$

Вероятностная продолжительность проекта:  $E \pm \sigma \approx 60,5 \pm 1,42$  дня.

Теперь нужно определить гарантию (в %) завершения проекта в определенный срок, вычисляя СНВ по математическим таблицам функции Гаусса.

Расчетная формула СНВ: разность между заданной (по обязательствам сдачи) и плановой (по срокам внутреннего завершения работ проекта) продолжительностями разделить на СКО проекта.

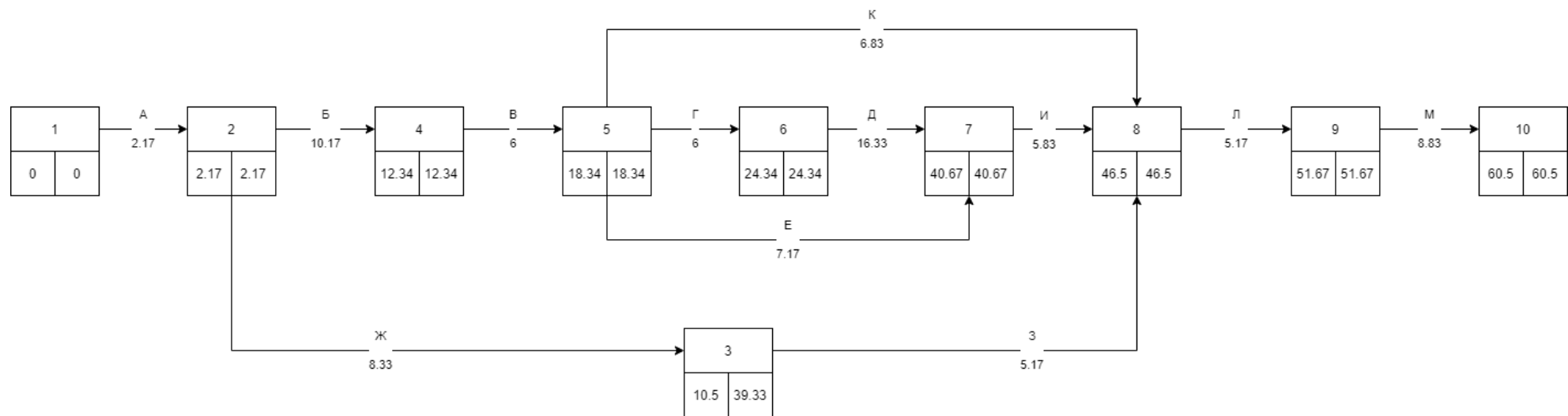


Рисунок 31 – Сетевая модель проекта по методу ПЕРТ

## 4.5 Программное обеспечение проекта

Для решения задач управления проектами на практике используются специальные программные продукты. Наиболее рациональным подходом к автоматизации поддержки соответствующих процессов является разработка информационной системы (ИС) управления проектами.

Для разработки ИС управления проектом воспользуемся программой OpenProject 1.4.

Согласно полученному критическому пути программы для реализации проекта потребуется 51 день.



		Название	Продолжи...	Начало	Окончание	Предшествующие
1		Анализ существующих ре...	2 дней	11.03.20 8:00	12.03.20 17:00	
2		Сбор и анализ требований	9 дней	13.03.20 8:00	25.03.20 17:00	1
3		Составление и согласовани	5 дней	26.03.20 8:00	01.04.20 17:00	2
4		Разработка дизайна	5 дней	02.04.20 8:00	08.04.20 17:00	3
5		Программирование	14 дней	09.04.20 8:00	28.04.20 17:00	4
6		Связывание БД с приложен	6 дней	02.04.20 8:00	09.04.20 17:00	3
7		Согласование выполненны	7 дней	26.03.20 8:00	03.04.20 17:00	2
8		Исправление замечаний за	4 дней	06.04.20 8:00	09.04.20 17:00	7
9		Подключение БД к серверу	4 дней	29.04.20 8:00	04.05.20 17:00	5;6
10		Тестирование	5 дней	02.04.20 8:00	08.04.20 17:00	3
11		Внедрение	5 дней	05.05.20 8:00	11.05.20 17:00	9;8;10
12		Написание технической до	7 дней	12.05.20 8:00	20.05.20 17:00	11

Рисунок 32 – Таблица для построения графика Ганта

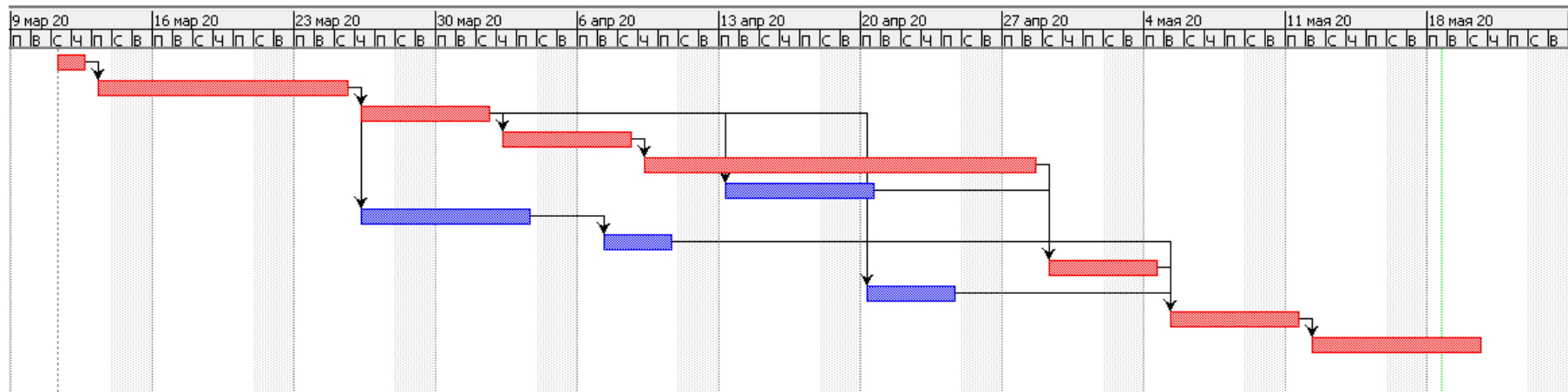


Рисунок 33 – График Ганта в OpenProject

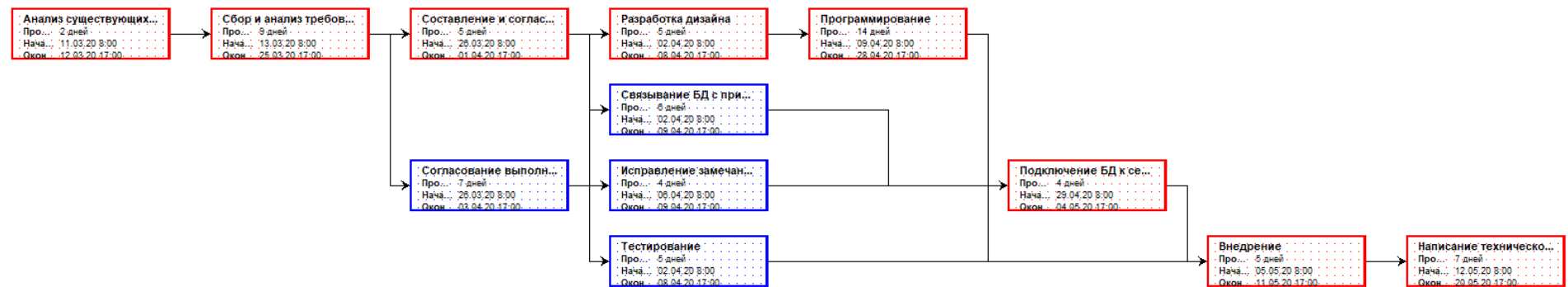


Рисунок 34 – Фрагмент сетевого графика в OpenProject



## 5 Программная реализация

В рамках данной курсовой работы был создан графический интерфейс, реализованы функции регистрации, авторизации, создания диалога, обработка прикрепленных медиаданных, сохранение сообщений в БД, сохранение фотоматериала в хранилище, связь приложения с БД.

Для создание пробной версии приложения будем использовать общедоступный сервис Google под название Firebase, предоставляющий разработчикам возможность создавать и использовать БД, работающие в реальном времени, также сервис имеет функцию добавления регистрации/авторизации различными способами, от возможности анонимной авторизации пользователя, до регистрации по номеру телефона. Также преимуществом данного сервиса является предоставление не только БД, но и хранилища для ваших файлов – FB Storage. Для подключения модулей авторизации, БД и хранилища к проекту следует зарегистрироваться на сайте Firebase: <https://console.firebase.google.com> с помощью вашего аккаунта гугла, создать на сайте проект, а в самом проекте создать и настроить интересующие нас модули, а именно Authentication, Database и Storage см. рисунок 35.

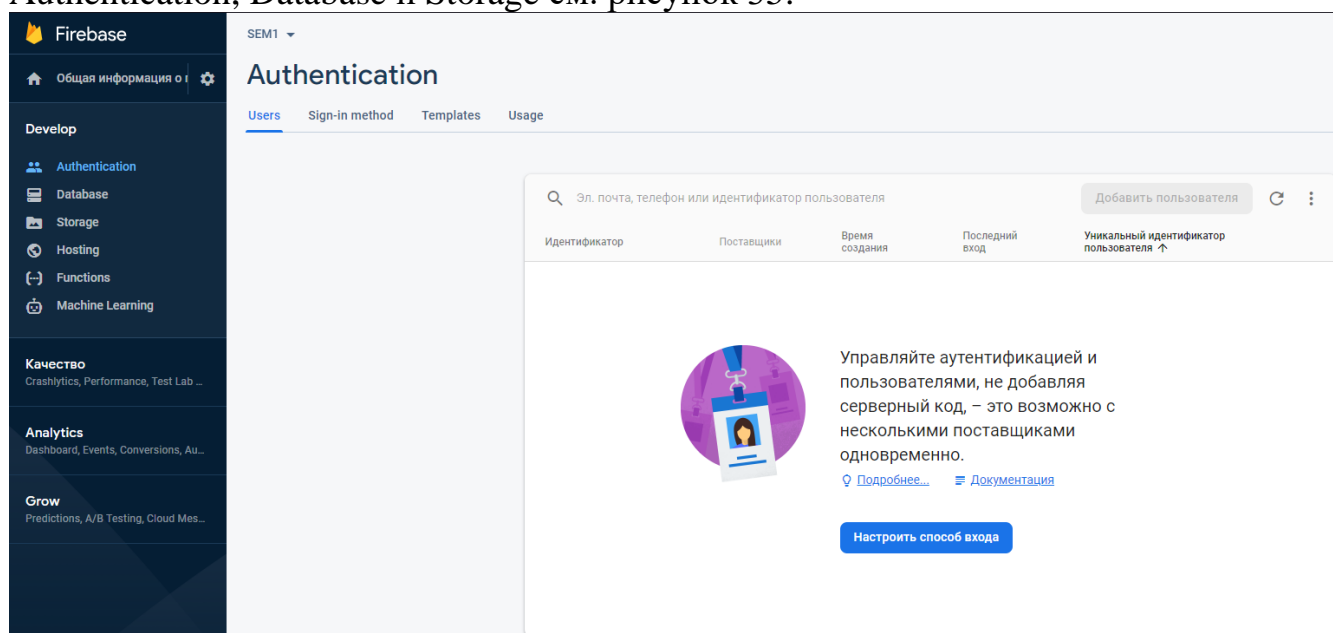


Рисунок 35 – Окно сайта Firebase

Далее в Android Studio авторизуемся с того же аккаунта и во вкладке Tools выбираем Firebase подключая интересующие нас модули прим. рисунок 36.

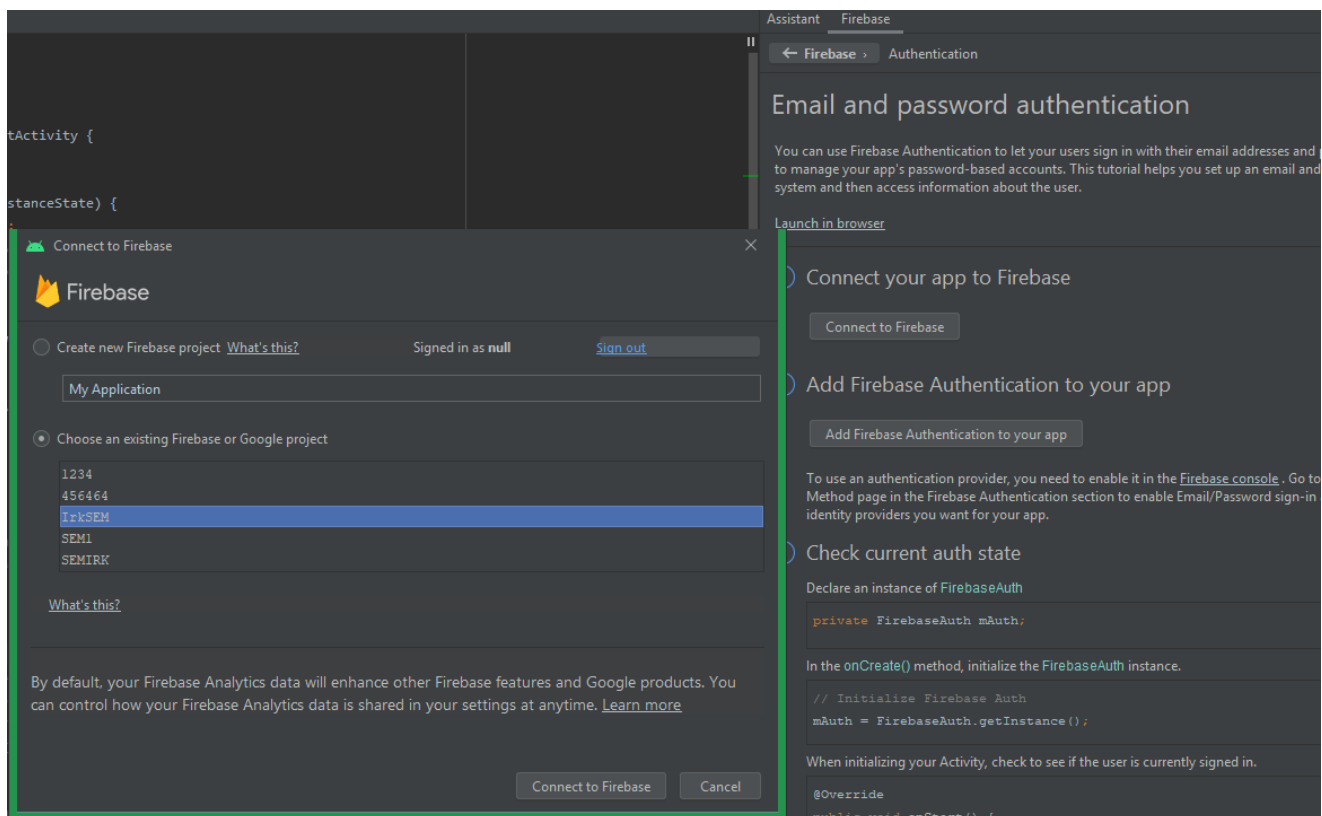


Рисунок 36 – Окно подключения модулей Firebase к проекту в Android Studio

После подключения нашего Firebase добавляем модуль к нашему приложению выполнив 2 пункт – add for your app (рисунок 37).

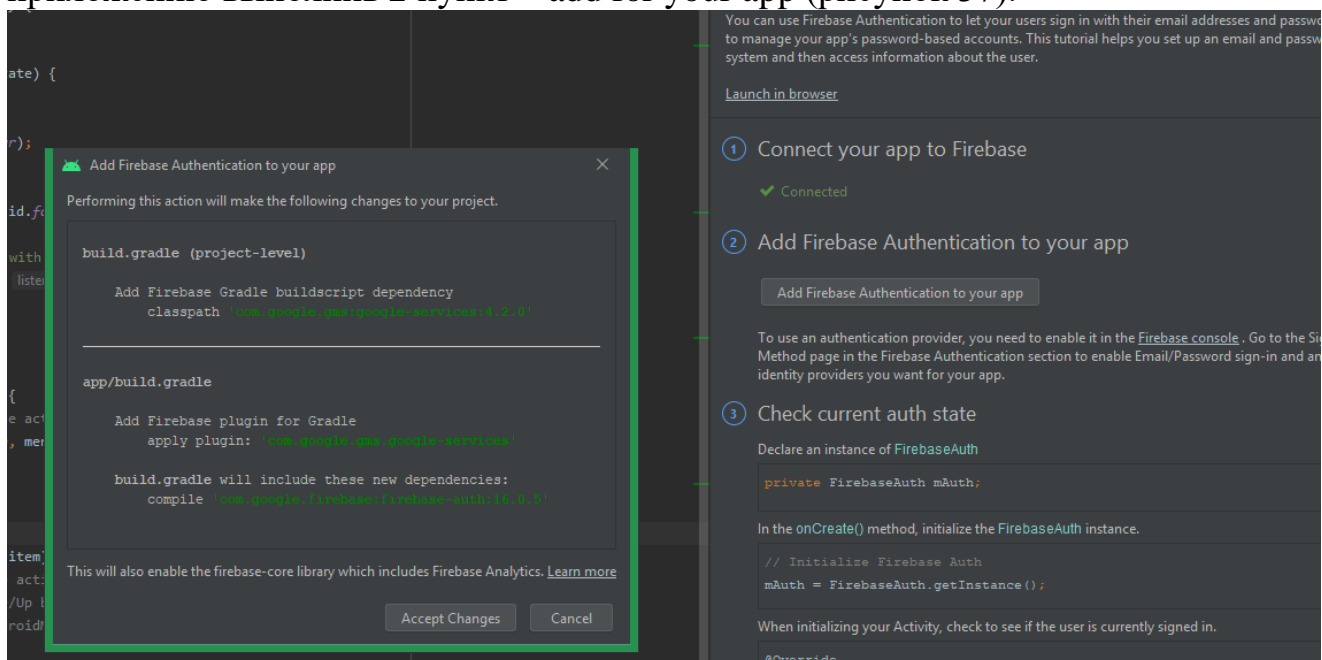


Рисунок 37 – Окно подключения модулей Firebase к проекту в Android Studio

После данных манипуляций мы можем работать с ресурсами Firebase, записывать и считывать данные, сохранять и загружать файлы, регистрировать и проверять авторизацию пользователей.

Пример кода для регистрации ниже.

```
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.GenericTypeIndicator;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;

public class RegistrationActivity extends AppCompatActivity {

    private FirebaseDatabase _firebase = FirebaseDatabase.getInstance();

    private Toolbar _toolbar;
    private HashMap<String, Object> userProfile = new HashMap<>();

    private ArrayList<HashMap<String, Object>> user = new ArrayList<>();

    private EditText sname;
    private EditText name;
    private EditText otch;
    private EditText email;
    private EditText password;
    private Button goreg;

    private Intent view = new Intent();
    private FirebaseAuth auth;
    private OnCompleteListener<AuthResult> _auth_create_user_listener;
    private OnCompleteListener<AuthResult> _auth_sign_in_listener;
    private OnCompleteListener<Void> _auth_reset_password_listener;
    private DatabaseReference userProf = _firebase.getReference("data/profiles");
    private ChildEventListener _userProf_child_listener;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.registration);
        com.google.firebase.FirebaseApp.initializeApp(this);
        initialize(savedInstanceState);
    }

    private void initialize(Bundle savedInstanceState) {
        _toolbar = (Toolbar) findViewById(R.id._toolbar);
        setSupportActionBar(_toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setHomeButtonEnabled(true);
        _toolbar.setNavigationOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View _v) {
                onBackPressed();
            }
        });
    }
}
```

```

    }
});
sname = (EditText) findViewById(R.id.sname);
name = (EditText) findViewById(R.id.name);
otch = (EditText) findViewById(R.id.otch);
email = (EditText) findViewById(R.id.email);
password = (EditText) findViewById(R.id.password);
goreg = (Button) findViewById(R.id.goreg);
auth = FirebaseAuth.getInstance();

//при нажатии на клавишу и прохождении проверки, создаем аккаунт с
логинем/паролем из элементов email и password

goreg.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View _view) {
        if ((email.getText().toString().equals("") ||
password.getText().toString().equals("")) ||
(sname.getText().toString().equals("") || name.getText().toString().equals(""))) {
            SketchwareUtil.showMessage(getApplicationContext(), "Заполнены не
все обязательные поля");
        }
        else {
            if (password.getText().toString().length() < 8) {
                SketchwareUtil.showMessage(getApplicationContext(), "Пароль
должен содержать не менее 8 символов");
            }
            else {
                auth.createUserWithEmailAndPassword(email.getText().toString(),
password.getText().toString()).addOnCompleteListener(RegistrationActivity.this,
_auth_create_user_listener);
            }
        }
    }
});

//при успешной регистрации создаем в БД профиль пользователя с данными из полей
email – почта, sname – фамилия, name – имя, otch – отчество, id – уникальный
идентификатор пользователя, созданный при регистрации, access – уровень
доступа/роль пользователя.

-
_auth_create_user_listener = new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(Task<AuthResult> _param1) {
        final boolean _success = _param1.isSuccessful();
        final String _errorMessage = _param1.getException() != null ?
_param1.getException().getMessage() : "";
        if (_success) {
            userProfile = new HashMap<>();
            userProfile.put("id",
FirebaseAuth.getInstance().getCurrentUser().getUid());
            userProfile.put("email",
FirebaseAuth.getInstance().getCurrentUser().getEmail());
            userProfile.put("sname", sname.getText().toString());
            userProfile.put("name", name.getText().toString());
            userProfile.put("otch", otch.getText().toString());
            userProfile.put("access", "user");
            userProf.push().updateChildren(userProfile);
            auth.signInWithEmailAndPassword(email.getText().toString(),
password.getText().toString()).addOnCompleteListener(RegistrationActivity.this,
_auth_sign_in_listener);

```

```

        view.setClass(getApplicationContext(), ChatsActivity.class);
        view.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);
        startActivity(view);
        finish();
    }
    else {
        SketchwareUtil.showMessage(getApplicationContext(), _errorMessage);
    }
}
};
-
}
}

```

На рисунке 38 изображено окно «Авторизации» для входа программы.

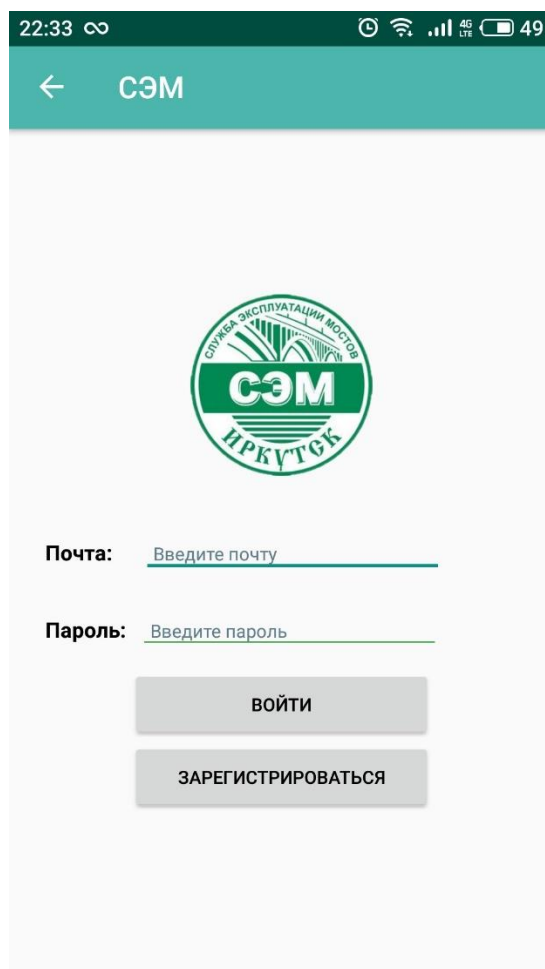


Рисунок 38 – Окно «Авторизации», форма main

Ниже представлен фрагмент кода файла, отвечающего за визуальную составляющую данной формы – main.xml.

```

<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    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">
    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">
        <androidx.appcompat.widget.Toolbar
            android:id="@+id/_toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay"/>
    </com.google.android.material.appbar.AppBarLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        app:layout_behavior="@string/appbar_scrolling_view_behavior">
        <LinearLayout
            android:id="@+id/linear2"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:padding="8dp"
            android:gravity="center_horizontal|center_vertical"
            android:orientation="vertical"
            android:layout_gravity="center_horizontal|center_vertical">
            <ImageView
                android:id="@+id/imageview1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginBottom="25dp"
                android:src="@drawable/icon"
                android:scaleType="center"/>
            <LinearLayout
                android:id="@+id/linear3"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:padding="8dp"
                android:orientation="horizontal">
                <TextView
                    android:id="@+id/textview1"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginRight="11dp"
                    android:padding="8dp"
                    android:text="Почта:"
                    android:textSize="14sp"
                    android:textStyle="bold"
                    android:textColor="#000000"/>
                <EditText
                    android:id="@+id/email"
                    android:layout_width="200dp"
                    android:layout_height="wrap_content"
                    android:padding="8dp"
                    android:textSize="12sp"
                    android:textColor="#000000"
                    android:hint="Введите почту"
                    android:textColorHint="#607D8B"
                    android:singleLine="true"/>
            </LinearLayout>
        <LinearLayout
            android:id="@+id/linear4"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

```

```

        android:padding="8dp"
        android:orientation="horizontal">
        <TextView
            android:id="@+id/textview2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="8dp"
            android:text="Пароль:"
            android:textSize="14sp"
            android:textStyle="bold"
            android:textColor="#000000"/>
        <EditText
            android:id="@+id/password"
            android:layout_width="200dp"
            android:layout_height="wrap_content"
            android:padding="8dp"
            android:textSize="12sp"
            android:textColor="#000000"
            android:hint="Введите пароль"
            android:textColorHint="#607D8B"
            android:singleLine="true"
            android:inputType="textPassword"
            android:imeOptions="actionGo"/>
    </LinearLayout>
    <Button
        android:id="@+id/authorization"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:padding="8dp"
        android:gravity="center_horizontal|center_vertical"
        android:text="Войти"
        android:textSize="12sp"
        android:textColor="#000000"
        android:layout_gravity="center_horizontal|center_vertical"/>
    <Button
        android:id="@+id/registration"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:padding="8dp"
        android:gravity="center_horizontal|center_vertical"
        android:text="Зарегистрироваться"
        android:textSize="12sp"
        android:textColor="#000000"
        android:layout_gravity="center_horizontal|center_vertical"/>
    </LinearLayout>
</LinearLayout>
</androidx.coordinatorlayout.widget.CoordinatorLayout>

```

Ниже представлен фрагмент кода отвечающего за логику работы данного окна – MainActivity.java

```

@Override
protected void onCreate(Bundle _savedInstanceState) {
    super.onCreate(_savedInstanceState);
    setContentView(R.layout.main);
    com.google.firebase.FirebaseApp.initializeApp(this);
    initialize(_savedInstanceState);
    initializeLogic();
}

private void initialize(Bundle _savedInstanceState) {

```

```

_toolbar = (Toolbar) findViewById(R.id._toolbar);
setSupportActionBar(_toolbar);
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setHomeButtonEnabled(true);
_toolbar.setNavigationOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View _v) {
        onBackPressed();
    }
});
linear2 = (LinearLayout) findViewById(R.id.linear2);
imageView1 = (ImageView) findViewById(R.id.imageview1);
linear3 = (LinearLayout) findViewById(R.id.linear3);
linear4 = (LinearLayout) findViewById(R.id.linear4);
authorization = (Button) findViewById(R.id.authorization);
registration = (Button) findViewById(R.id.registration);
textview1 = (TextView) findViewById(R.id.textview1);
email = (EditText) findViewById(R.id.email);
textview2 = (TextView) findViewById(R.id.textview2);
password = (EditText) findViewById(R.id.password);
auth = FirebaseAuth.getInstance();

authorization.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View _view) {
        if (email.getText().toString().equals("") ||
password.getText().toString().equals("")) {
            SketchwareUtil.showMessage(getApplicationContext(), "Заполнены не
все поля");
        }
        else {
            auth.signInWithEmailAndPassword(email.getText().toString(),
password.getText().toString()).addOnCompleteListener(MainActivity.this,
_auth_sign_in_listener);
        }
    }
});

registration.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View _view) {
        view.setClass(getApplicationContext(), RegistrationActivity.class);
        startActivity(view);
        view.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);
    }
});

_auth_create_user_listener = new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(Task<AuthResult> _param1) {
        final boolean _success = _param1.isSuccessful();
        final String _errorMessage = _param1.getException() != null ?
_param1.getException().getMessage() : "";
    }
};

_auth_sign_in_listener = new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(Task<AuthResult> _param1) {
        final boolean _success = _param1.isSuccessful();

```



```

        final String _errorMessage = _param1.getException() != null ?
_param1.getException().getMessage() : "";
        if (_success) {
            view.setClass(getApplicationContext(), ChatsActivity.class);
            startActivity(view);
            view.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);
        }
        else {
            SketchwareUtil.showMessage(getApplicationContext(), _errorMessage);
        }
    }
};

_auth_reset_password_listener = new OnCompleteListener<Void>() {
    @Override
    public void onComplete(Task<Void> _param1) {
        final boolean _success = _param1.isSuccessful();
    }
};

private void initializeLogic() {
    if ((FirebaseAuth.getInstance().getCurrentUser() != null)) {
        view.setClass(getApplicationContext(), ChatsActivity.class);
        startActivity(view);
        view.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);
    }
}

@Override
protected void onActivityResult(int _requestCode, int _resultCode, Intent
_data) {
    super.onActivityResult(_requestCode, _resultCode, _data);

    switch (_requestCode) {

        default:
            break;
    }
}
}

```

На рисунке 39 представлено окно регистрации пользователя.

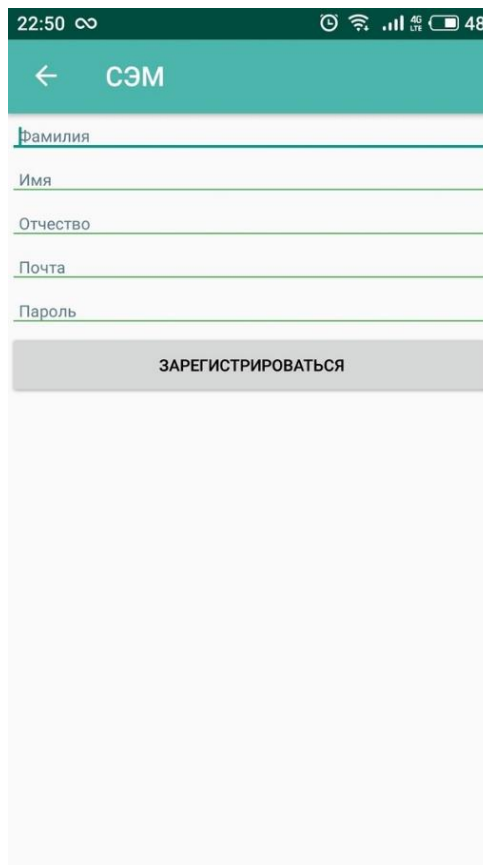


Рисунок 39 – Окно «Регистрации», форма registration

Ниже представлен фрагмент кода файла, отвечающего за логику работы данного окна.

```
private void initialize(Bundle _savedInstanceState) {

    _toolbar = (Toolbar) findViewById(R.id._toolbar);
    setSupportActionBar(_toolbar);
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    getSupportActionBar().setHomeButtonEnabled(true);
    _toolbar.setNavigationOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View _v) {
            onBackPressed();
        }
    });
    sname = (EditText) findViewById(R.id.sname);
    name = (EditText) findViewById(R.id.name);
    otch = (EditText) findViewById(R.id.otch);
    email = (EditText) findViewById(R.id.email);
    password = (EditText) findViewById(R.id.password);
    goreg = (Button) findViewById(R.id.goreg);
    auth = FirebaseAuth.getInstance();

    goreg.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View _view) {
            if ((email.getText().toString().equals("") ||
password.getText().toString().equals("")) ||
(sname.getText().toString().equals("") || name.getText().toString().equals(""))) {
                SketchwareUtil.showMessage(getApplicationContext(), "Заполнены не все
обязательные поля");
            }
        }
    });
}
```

```

        else {
            if (password.getText().toString().length() < 8) {
                SketchwareUtil.showMessage(getApplicationContext(), "Пароль должен
содержать не менее 8 символов");
            }
            else {
                auth.createUserWithEmailAndPassword(email.getText().toString(),
password.getText().toString()).addOnCompleteListener(RegistrationActivity.this,
_auth_create_user_listener);
            }
        }
    }
});

_userProf_child_listener = new ChildEventListener() {
    @Override
    public void onChildAdded(DataSnapshot _param1, String _param2) {
        GenericTypeIndicator<HashMap<String, Object>> _ind = new
GenericTypeIndicator<HashMap<String, Object>>() {};
        final String _childKey = _param1.getKey();
        final HashMap<String, Object> _childValue = _param1.getValue(_ind);
    }
}

```

На рисунке 40 показана форма, на которой отображаются все доступные пользователю чаты.

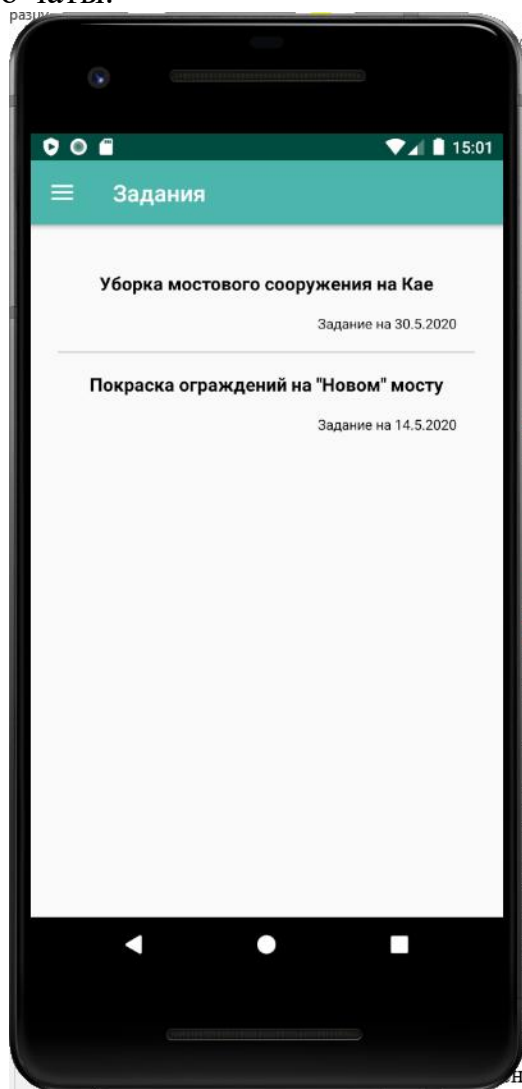


Рисунок 40 – Окно «Чаты/задания», форма chats

Ниже представлен фрагмент кода файла, отвечающего за визуальное составляющее данного окна.

```
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/_drawer"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:openDrawer="start">
    <androidx.coordinatorlayout.widget.CoordinatorLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <com.google.android.material.appbar.AppBarLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:theme="@style/AppTheme.AppBarOverlay">
            <androidx.appcompat.widget.Toolbar
                android:id="@+id/_toolbar"
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                android:background="?attr/colorPrimary"
                app:popupTheme="@style/AppTheme.PopupOverlay"/>
            </com.google.android.material.appbar.AppBarLayout>
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:orientation="vertical"
                app:layout_behavior="@string/appbar_scrolling_view_behavior">
                <LinearLayout
                    android:id="@+id/linear1"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:padding="8dp"
                    android:orientation="vertical">
                    <LinearLayout
                        android:id="@+id/linear2"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:padding="8dp"
                        android:orientation="horizontal">
                        <ImageView
                            android:id="@+id/addchat"
                            android:layout_width="35dp"
                            android:layout_height="35dp"
                            android:src="@drawable/ic_add_black"
                            android:scaleType="fitCenter"
                            android:layout_gravity="center_horizontal|center_vertical"/>
                        <TextView
                            android:id="@+id/textview1"
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:padding="8dp"
                            android:text="Добавить задание"
                            android:textSize="14sp"
                            android:textColor="#000000"/>
                        </LinearLayout>
                    <LinearLayout
                        android:id="@+id/linear3"
                        android:layout_width="wrap_content"
```

```

        android:layout_height="match_parent"
        android:padding="8dp"
        android:orientation="vertical">
        <ListView
            android:id="@+id/listview1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="8dp"
            android:dividerHeight="2dp"
            android:choiceMode="none"/>
        </LinearLayout>
    </LinearLayout>
</androidx.coordinatorlayout.widget.CoordinatorLayout>
<LinearLayout
    android:id="@+id/_nav_view"
    android:layout_width="320dp"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:background="#EEEEEE">
    <include layout="@layout/_drawer_chats"/>
</LinearLayout>
</androidx.drawerlayout.widget.DrawerLayout>

```

Данное окно имеет также боковое меню, описывающееся в другом фрагменте кода.

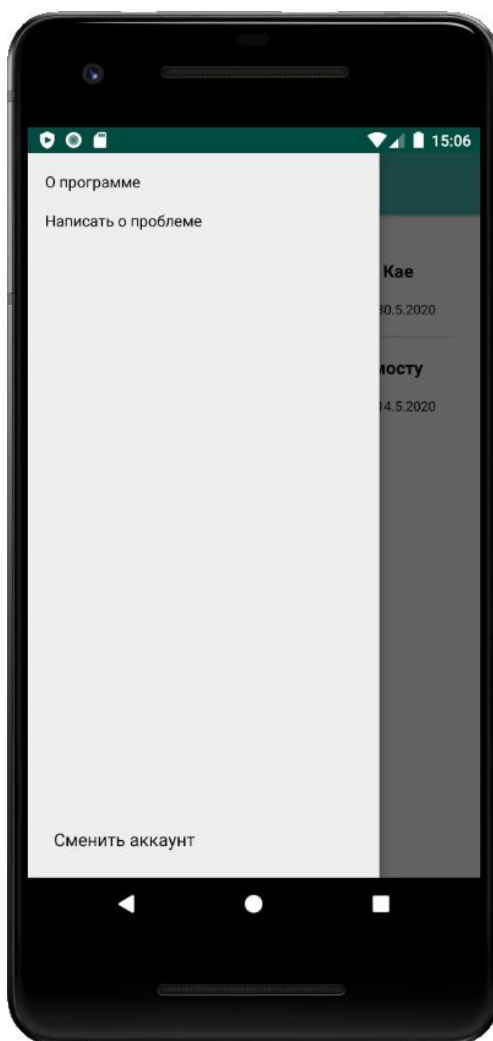


Рисунок 41 – Боковое меню окна «Чаты/задания»

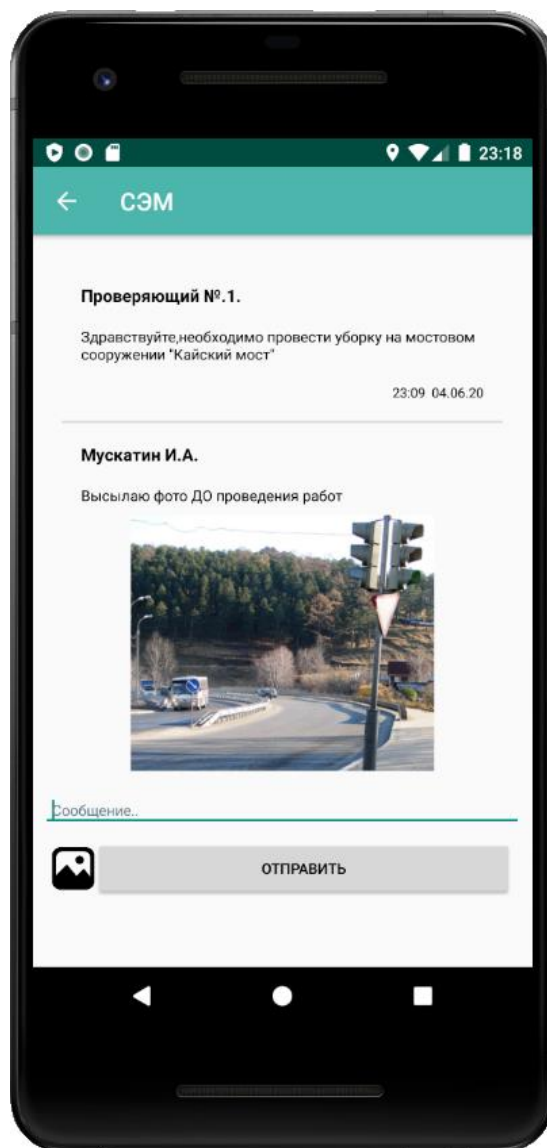


Рисунок 42 – Окно «Личный чат/задание конкретного подрядчика», форма chat

Ниже представлен фрагмент кода файла, отвечающего за логику работы данного окна.

```
private void initialize(Bundle _savedInstanceState) {

    _toolbar = (Toolbar) findViewById(R.id._toolbar);
    setSupportActionBar(_toolbar);
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    getSupportActionBar().setHomeButtonEnabled(true);
    _toolbar.setNavigationOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View _v) {
            onBackPressed();
        }
    });
    linear3 = (LinearLayout) findViewById(R.id.linear3);
    linear2 = (LinearLayout) findViewById(R.id.linear2);
    message = (EditText) findViewById(R.id.message);
    linear5 = (LinearLayout) findViewById(R.id.linear5);
    listview1 = (ListView) findViewById(R.id.listview1);
    adding = (ImageView) findViewById(R.id.addimg);
}
```

```

sendmessage = (Button) findViewById(R.id.sendmessage);
auth = FirebaseAuth.getInstance();
location = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
fp.setType("image/*");
fp.putExtra(Intent.EXTRA_ALLOW_MULTIPLE, true);
_file_cam = FileUtil.createNewPictureFile(getApplicationContext());
Uri _uri_cam = null;
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.N) {
    _uri_cam = FileProvider.getUriForFile(getApplicationContext(),
getApplicationContext().getPackageName() + ".provider", _file_cam);
}
else {
    _uri_cam = Uri.fromFile(_file_cam);
}
cam.putExtra(MediaStore.EXTRA_OUTPUT, _uri_cam);
cam.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
dial = new AlertDialog.Builder(this);

adding.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View _view) {
        dial.setMessage("Загрузить фото");
        dial.setPositiveButton("Сделать снимок", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface _dialog, int _which) {
                startActivityForResult(cam, REQ_CD_CAM);
            }
        });
        dial.setNeutralButton("Загрузить с телефона", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface _dialog, int _which) {
                startActivityForResult(fp, REQ_CD_FP);
            }
        });
        dial.create().show();
    }
});

sendmessage.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View _view) {
        if (message.getText().toString().equals("") && (nimg == 0)) {
        }
        else {
            chatm = new HashMap<>();
            chatm.put("idchat", idchat);
            chatm.put("id", FirebaseAuth.getInstance().getCurrentUser().getUid());
            chatm.put("name", fio);
            if (!message.getText().toString().equals("")) {
                chatm.put("message", message.getText().toString());
            }
            chatm.put("time", new SimpleDateFormat("HH:mm
dd.MM.yy").format(time.getTime()));
            chatm.put("position", pos);
            message.setText("");
            if (nimg == 0) {
                chat.push().updateChildren(chatm);
            }
        }
    }
});

```

```

    }
    else {
        picture.child(pname).putFile(Uri.fromFile(new
File(path))).addOnFailureListener(_picture_failure_listener).addOnProgressListener
(_picture_upload_progress_listener).continueWithTask(new
Continuation<UploadTask.TaskSnapshot, Task<Uri>>() {
            @Override
            public Task<Uri> then(Task<UploadTask.TaskSnapshot> task) throws
Exception {
                return picture.child(pname).getDownloadUrl();
            }}).addOnCompleteListener(_picture_upload_success_listener);
        nimg = 0;
    }
}
});

```

На рисунке 43 показана форма, на которой проверяющий может создавать задания (чаты).

23:31 4G LTE 45

← **Добавление задания**

Работа №7

Подрядчик: Мускатин И.А. ▼

Выберите дату работ:

< Июнь 2020 г. >

пн	вт	ср	чт	пт	сб	вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

ВЫДАТЬ ЗАДАНИЕ

Рисунок 43 – Окно «Создания чата/задания для конкретного подрядчика», форма adddialog



Ниже представлен фрагмент кода файла, отвечающего за логику работы данного окна.

```
private void initialize(Bundle _savedInstanceState) {

    _toolbar = (Toolbar) findViewById(R.id._toolbar);
    setSupportActionBar(_toolbar);
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    getSupportActionBar().setHomeButtonEnabled(true);
    _toolbar.setNavigationOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View _v) {
            onBackPressed();
        }
    });
    editText1 = (EditText) findViewById(R.id.edittext1);
    linear1 = (LinearLayout) findViewById(R.id.linear1);
    textview2 = (TextView) findViewById(R.id.textview2);
    calendarview1 = (CalendarView) findViewById(R.id.calendarview1);
    setquest = (Button) findViewById(R.id.setquest);
    textview1 = (TextView) findViewById(R.id.textview1);
    spinner1 = (Spinner) findViewById(R.id.spinner1);

    calendarview1.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {
        @Override
        public void onSelectedDayChange(CalendarView _param1, int _param2, int
_param3, int _param4) {
            final int _year = _param2;
            final int _month = _param3;
            final int _day = _param4;
            date =
String.valueOf(_day).concat(".").concat(String.valueOf(_month).concat(".").concat(St
ring.valueOf(_year))));
        }
    });

    setquest.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View _view) {
            work = new HashMap<>();
            work.put("staff", user.get("id").toString());
            work.put("datetime", date);
            work.put("name", editText1.getText().toString());
            work.put("observ", profile.get("id").toString());
            dbwork.push().updateChildren(work);
            view.setClass(getApplicationContext(), ChatsActivity.class);
            view.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);
            startActivity(view);
            finish();
        }
    });
}
```

## 6 Тестирование

Таблица 11 –тестирование приложения

№	Тест	Входные данные	Выходные данные
1	Авторизация при не заполненных полях	Введен один логин	Всплывающая подсказка об ошибке (см. рисунок 44)
2	Авторизация при заполненных полях, но несуществующим аккаунтом	Введены ранее не регистрируемые в БД логин и пароль	Всплывающая подсказка о том, что, либо аккаунт был удален, либо его не существовало (см. рисунок 45)
3	Авторизация при заполненных полях, но с ошибочным паролем	Введена корректная почта, но неверный пароль	Всплывающая подсказка о неверном пароле (см. рисунок 46)
4	Регистрация при незаполненных полях	Введены не все обязательные данные в поля регистрации или не введены вовсе	Всплывающая подсказка об ошибке и просьбе заполнить обязательные поля (см. рисунки 47-49)
5	Регистрация при размере введенного пароля меньше 8 символов	Заполненные обязательные поля, поле пароля заполнено 7 символами	Всплывающая подсказка об ошибке ввода некорректного пароля (см. рисунок 50)
6	Успешная регистрация	Введенные пользователем личные данные	Занесение профиля пользователя в БД (см. рисунок 51)
7	Добавление нового задания без указания его названия и даты	Выбран подрядчик	Сообщение о пустых обязательных полях (см. рисунок 52)
8	Добавление нового задания без указания его названия	Выбран подрядчик и дата	Сообщение о пустом поле названия работы (см. рисунок 53)

9	Добавление нового задания без указания его даты (хоть дата и отображается выбранной, но не была нажата пользователем)	Выбран подрядчик и название работы	Сообщение о том, что дата для задания не установлена (см. рисунок 54)
10	Добавление задания	Все поля заполнены	Добавление данных о задании в БД (см. рисунок 55-56)
11	Проверка отправки только текста сообщения	Введенный текст «привет»	Результат работы приложения (см. рисунок 57)
12	Проверка отправки текста сообщения с прикрепленным файлом	Введенный текст «Проверка текста+фотоматериал» и фотография, сделанная на камеру через приложение	Результат работы приложения (см. рисунок 58)
13	Проверка отправки сообщения с прикрепленным файлом	Изображение, загруженное из телефона (см. рисунок 59)	Результат работы приложения (см. рисунок 60)
14	Проверка разграничение доступа	Авторизация под аккаунтом подрядчика (прим. Пупкин)	Отсутствие на форме с заданиями кнопки добавления новых заданий, а также отсутствие других диалогов, помимо своего (см. рисунки 61-62)
15	Проверка на корректное отображение диалогов у обеих сторон	Левый телефон авторизован как подрядчик, правый как проверяющий	Подтверждение корректной работоспособности чата, для обеих сторон (см. рисунок 63)

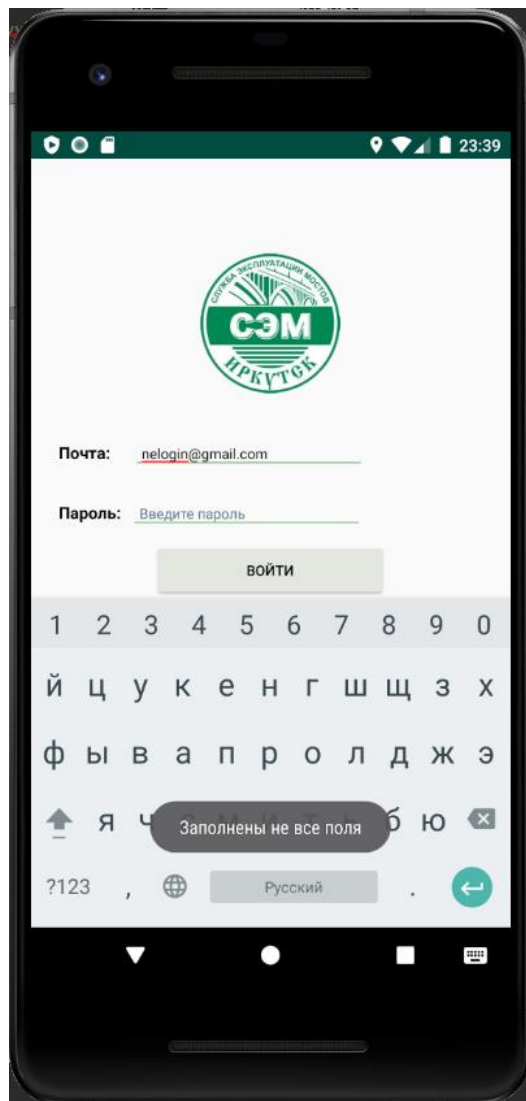


Рисунок 44 – Окно «Авторизации» с подсказкой

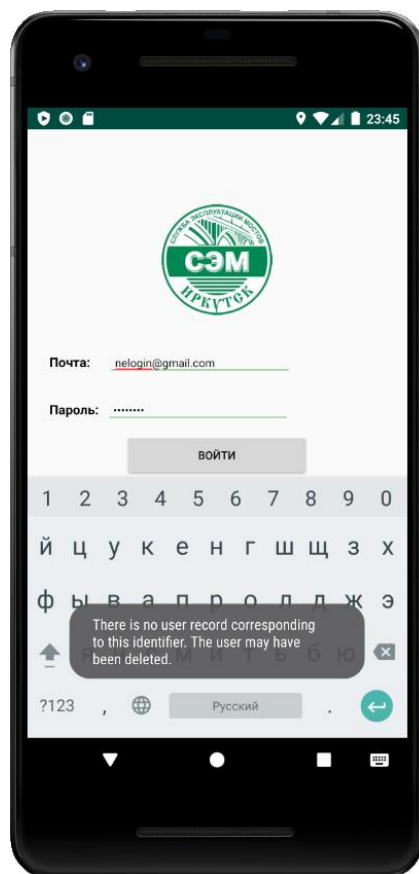


Рисунок 45 – Окно «Авторизации» с сообщением о невозможности входа в систему

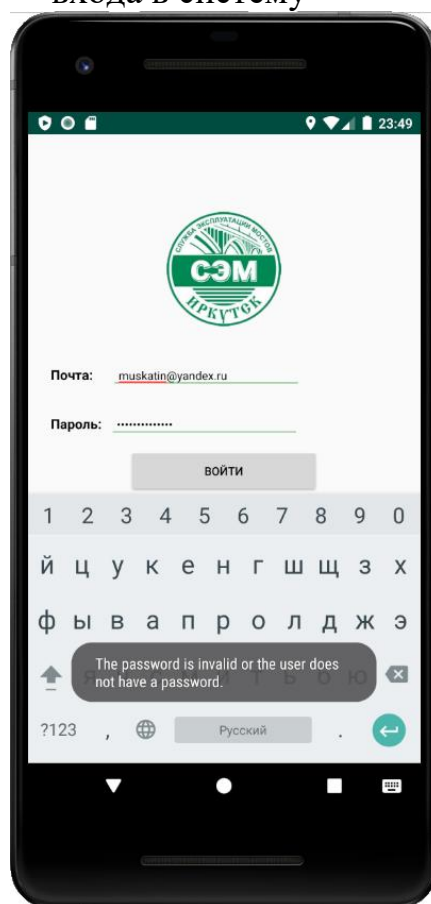


Рисунок 46 – Окно «Авторизации» с сообщением о неверном пароле

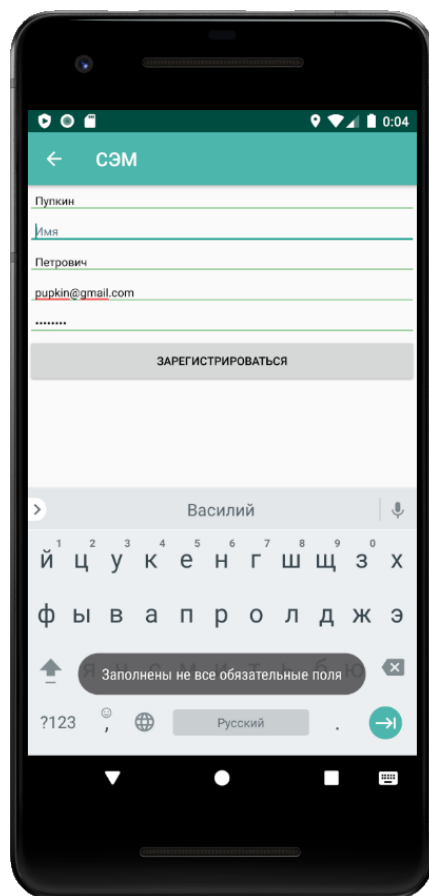


Рисунок 47 – Сообщение о некорректной регистрации

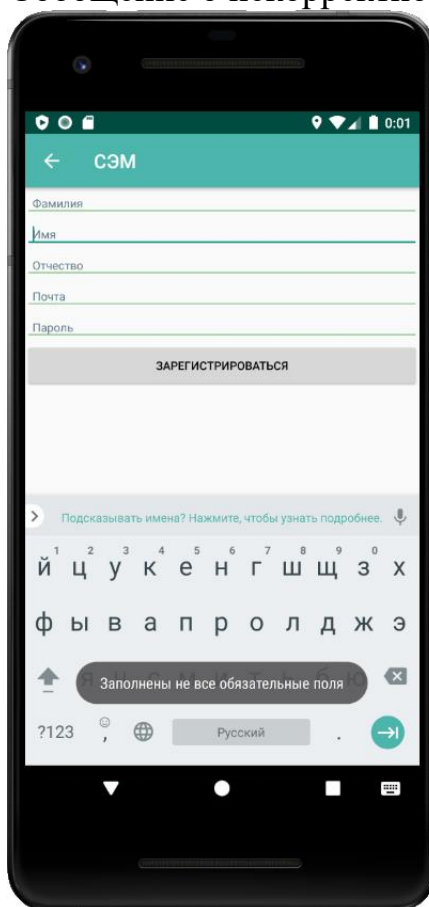


Рисунок 48 – Сообщение о некорректной регистрации

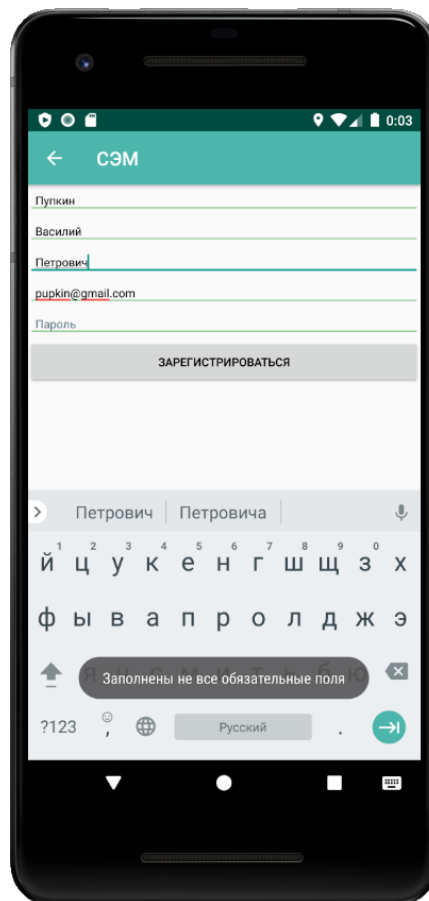


Рисунок 49 – Сообщение о некорректной регистрации

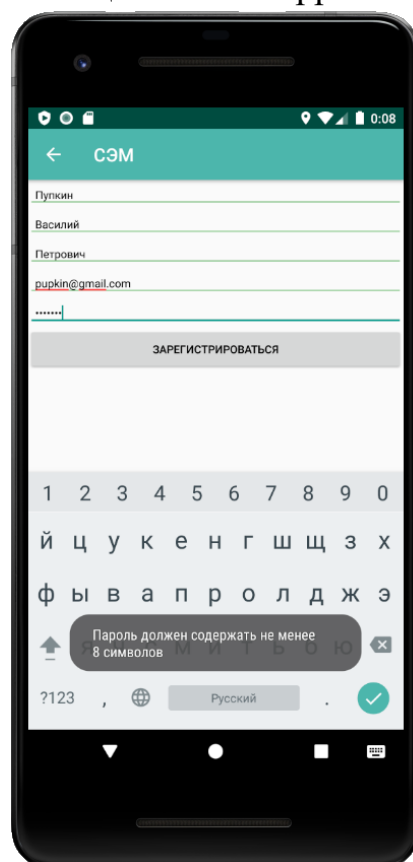


Рисунок 50 – Сообщение о некорректности размера введенного пароля

```
-M9-5MQK5kQ8J246m0xd + x
{
  access: "user"
  email: "pupkin@gmail.com"
  id: "A4dXcZPjVnT0EpP47P4SRqZJZk6"
  name: "Василий"
  otch: "Петрович"
  sname: "Пупкин"
}
```

Рисунок 51 – Запись о новом профиле в БД

0:19 ∞ 43

← Добавление задания

Наименование работ...

Подрядчик: Пупкин В.П. ▼

Выберите дату работ:

< Июнь 2020 г. >

пн	вт	ср	чт	пт	сб	вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Заполнены не все обязательные поля

ВЫДАТЬ ЗАДАНИЕ

Рисунок 52 – Сообщение о некорректном вводе



0:20 43

← Добавление задания

Наименование работ...

Подрядчик: Пупкин В.П.

Выберите дату работ:

< Июнь 2020 г. >

пн	вт	ср	чт	пт	сб	вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

Вы не ввели название задачи

ВЫДАТЬ ЗАДАНИЕ

Рисунок 53 – Сообщение о пустом поле задания

0:20 43

← Добавление задания

hello world

Подрядчик: Пупкин В.П.

Выберите дату работ:

< Июнь 2020 г. >

пн	вт	ср	чт	пт	сб	вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Вы не назначили дату проведения работ

ВЫДАТЬ ЗАДАНИЕ

Рисунок 54 – Реакция на игнорирование выбора даты

1:19 ∞ 35

← Добавление задания

Задание №1

Подрядчик: Пупкин В.П. ▼

Выберите дату работ:

< Июнь 2020 г. >

пн	вт	ср	чт	пт	сб	вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

ВЫДАТЬ ЗАДАНИЕ

Рисунок 55 – Корректное создание задания

```

- -M9-KyX3Sn9Kk0QqirTd + ×
{
  datetime: "11.6.2020"
  id: "-M9-KyX3Sn9Kk0QqirT"
  name: "Задание №1"
  observ: "pHb9JyiQasZIrRRYBCx6aPw5APs"
  staff: "A4dXcZPjVnT0EpP47P4SRqZJZk6"
}

```

Рисунок 56 – Запись о новом задании в БД

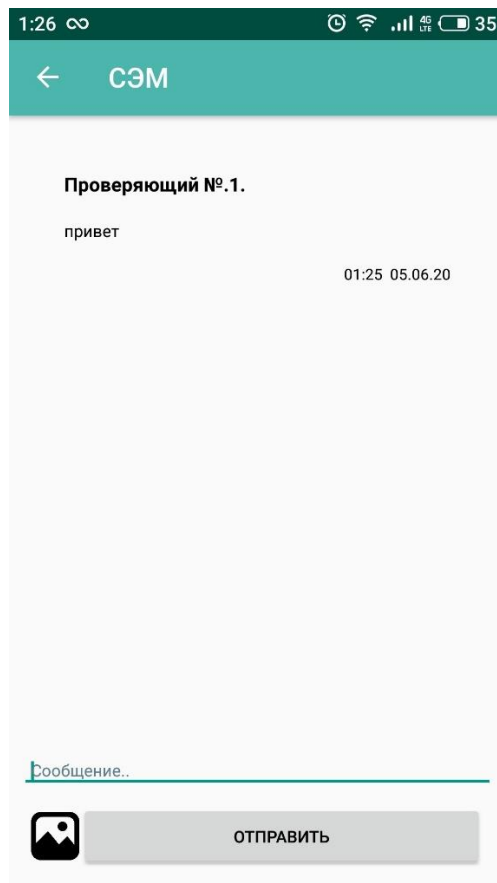


Рисунок 57 – Реакция на отправление только текста

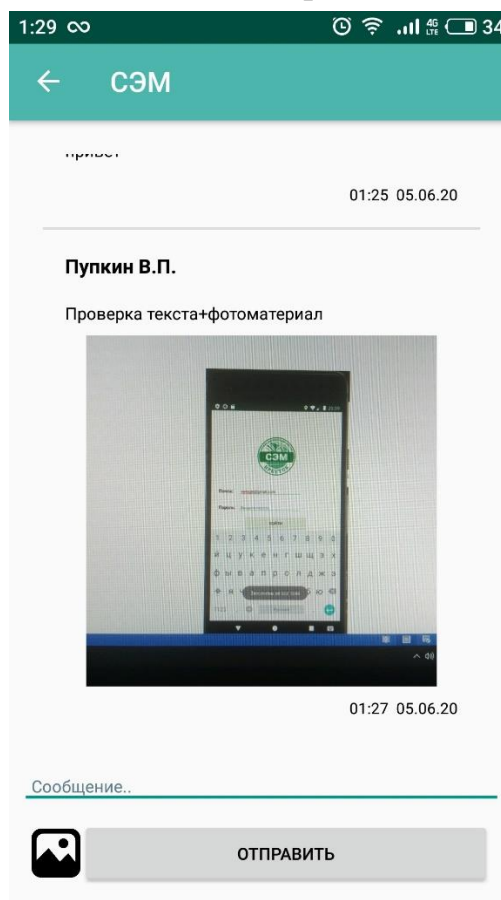


Рисунок 58 – Реакция на отправление текста и фото с камеры

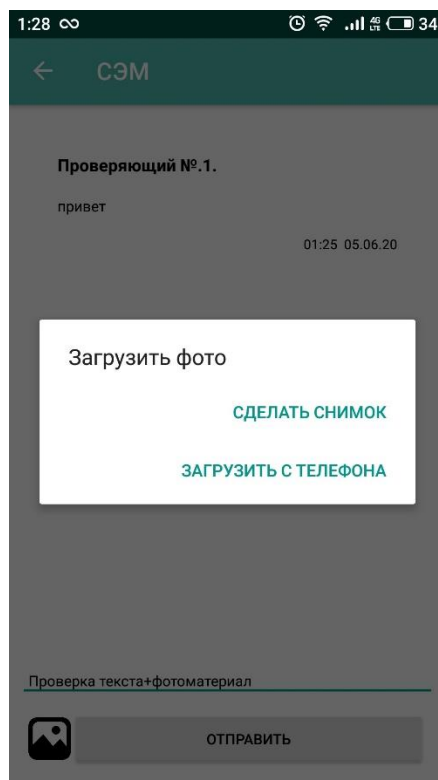


Рисунок 59 – Диалоговое окно выбора варианта прикрепления медиафайла к сообщению

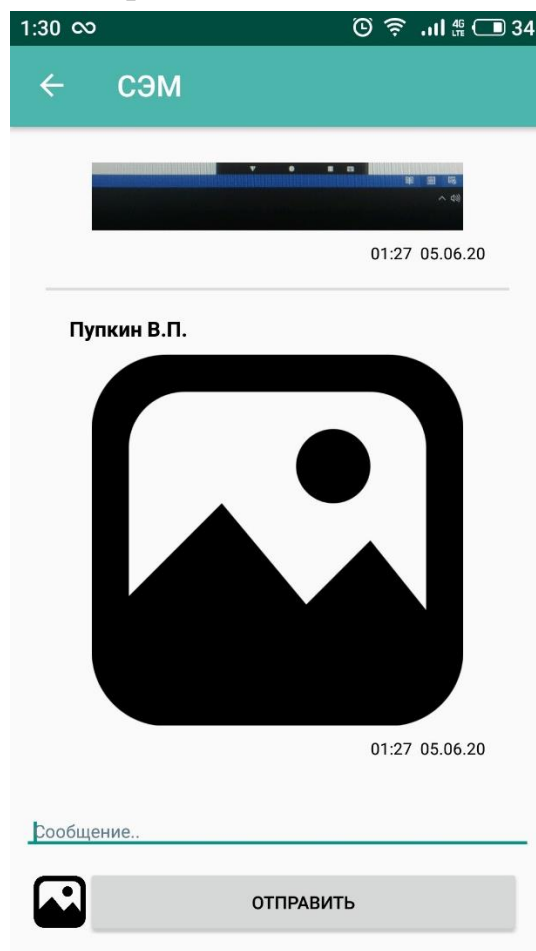


Рисунок 60 – Результат отправки одного изображения в сообщении



Рисунок 61 – Вид окна диалогов для авторизованного пользователя с ролью подрядчик



Рисунок 62 – Вид окна диалогов для авторизованного пользователя с ролью проверяющий

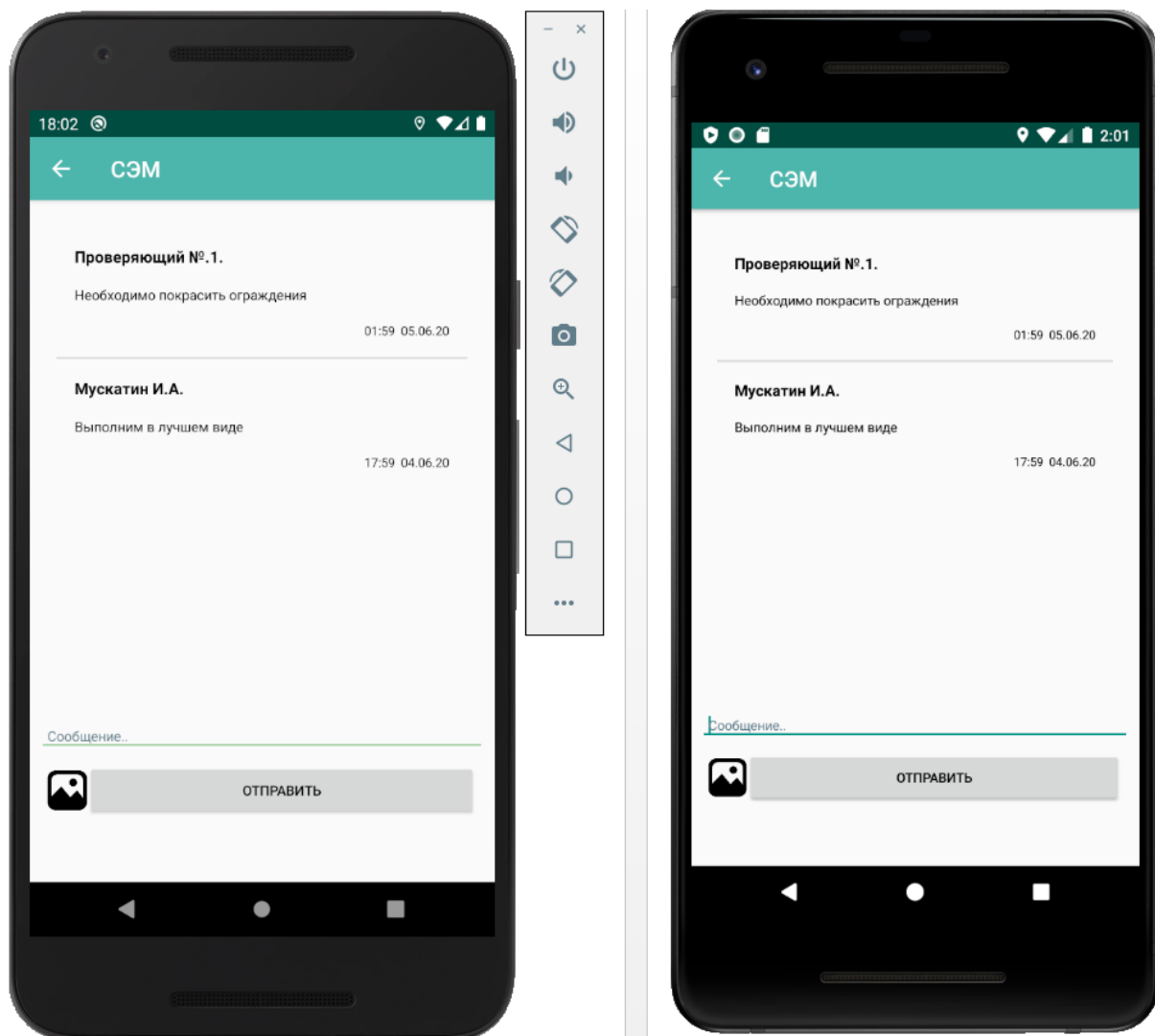


Рисунок 63 – Вид окна диалогов для авторизованного пользователя с ролью подрядчик

## **7 Разработка документации**

### **1 Введение**

#### **1.1 Область применения**

Разрабатываемый программный продукт предназначен для автоматизации сбора данных работы подрядчиков компании «СЭМ», таких как геопозиция сделанных снимков, дата и время выполнения задач и т.п.

Система должна хранить данные о подрядчиках, проверяющих, сообщениях, чатах и фотографиях, позволять создавать новые чаты, разграничивать доступ к некоторым функциям приложения, в зависимости от роли пользователя.

#### **1.2 Краткое описание возможностей**

Мобильное приложение «СЭМ» выполняет следующие функции:

- Регистрация и авторизация пользователя в системе;
- Создание чатов (заданий);
- Отправка сообщений;
- Фильтрация доступных чатов.

#### **1.3 Уровень подготовки пользователя:**

Пользователь должен иметь при себе телефон на операционной системе Android и знать, как включать на нем геолокацию с интернетом.

#### **1.4 Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю:**

- Руководство пользователя (user's guide);

### **2 Назначение и условие применения «СЭМ»**

2.1 Данное средство автоматизации предназначено для автоматизации процессов работы подрядчиков и проверяющих;

2.2 Работа с приложением возможна, если выполнены следующие условия:

Вид ЭВМ и конфигурации технических средств:

- Android: 7.0 и выше.
- База Данных: Firebase;

Работа с «СЭМ» доступна всем пользователям с подтвержденными правами доступа.

### **3 Подготовка к работе**

Чтобы воспользоваться программой предлагаем выполнить следующие действия:

- Установить программу на телефон или планшет.
- Ознакомиться с руководством пользователя, с окнами и возможностями программы.

#### 4 Описание операций

Таблица 12 - Выполняемые функции

Функции	Описание
Регистрация пользователей в системе	В ходе выполнения данной функции система создает профиль пользователя в БД с определением уровня доступа.
Авторизация пользователей в системе	В ходе выполнения данной функции система проверяет данные введенные пользователем для предоставления ему доступа к работе в системе.
Создание чатов (заданий);	В ходе выполнения данной функции пользователю с ролью (уровнем доступа) проверяющего предоставляется возможность создать диалог-задание для подрядчика с назначением выбранного подрядчика и даты выполнения задачи, эти данные сохраняются в БД, для последующей обработки.
Отправка сообщений	В ходе выполнения данной функции пользователю предоставляется возможность написать и отправить собеседнику по чату сообщение, а также включает возможность передачи в сообщении медиа файлов-фотографий.
Фильтрация доступных чатов	В ходе выполнения данной функции пользователю при авторизации в системе демонстрируется только та часть диалогов, в которые он имеет право доступа.

Описание операций технологического процесса обработки данных, необходимых для выполнения функций

**Функция: «Регистрация пользователей в системе»**

Условия, необходимые для выполнения функции:

- Вход в приложение.

**Функция: «Авторизация пользователей в системе»**

Условия, необходимые для выполнения функции:

- Вход в приложение;



- Успешная регистрация в системе;

**Функция: «Создание чатов (заданий)»**

Условия, необходимые для выполнения функции:

- Авторизация в приложении;
- Наличие подрядчиков в системе;
- Наличие у пользователя уровня доступа – проверяющий;
- Выбор даты;
- Написание названия задания.

**Функция: «Отправка сообщений»**

Условия, необходимые для выполнения функции:

- Авторизация в системе;
- Наличие доступного диалога (чата);
- Заполненное поле текста сообщения или прикрепленный документ.

**Функция: «Фильтрация доступных чатов»**

Условия, необходимые для выполнения функции:

- Авторизация в системе;
- Наличие пользователя в чате.

**5 Аварийные ситуации**

Таблица 13 – Аварийные ситуации

Ошибка	Описание ошибки	Действия при возникновении ошибки
Невозможно войти в систему	Сбой работы системы	Для устранения проблем необходимо обратиться к системному администратору
Невозможно войти в систему	Забыли пароль от учетной записи	Для устранения проблемы необходимо обратиться к системному администратору, который сможет помочь вам восстановить пароль
Невозможно войти в систему	Не подходят верные данные авторизации, возможно сбой работы системы или удаление профиля в БД	Необходимо обратиться к системному администратору

Продолжение таблицы 13 на следующей странице

Продолжение таблицы 13

Нет прав доступа	Нет возможности использовать какие-либо функции системы	Для устранения нужно зайти от имени пользователя, которому доступны данные действия или обратиться к системному администратору для увеличения прав доступа данной учетной записи
Нет связи с БД	Нет возможности связаться с базой данных или база данных не найдена	Для устранения проблем необходимо обратиться к системному администратору
Ошибка ввода данных	Данные были введены неправильно, неверный тип данных	Для устранения проблем необходимо ввести данные корректно

## **Заключение**

В рамках данной курсовой работы было разработано мобильное приложение на языке Java, автоматизирующее процесс сбора данных при коммуникации проверяющих и подрядчиков компании «СЭМ».

В ходе работы были получены навыки по формализации системных требований к программному продукту, по архитектурному и интерфейсному проектированию.

Были освоены методологии построения моделей бизнес-процессов и основных компонентов программного продукта.

Были закреплены навыки работы с базами данных, был получен опыт работы с базами данных работающими в реальном времени Firebase.

На данном этапе были формализованы системные требования и спроектирована система, создан графический интерфейс мобильного приложения, реализованы функции регистрации, авторизации, просмотра списков диалогов, отфильтрованных для конкретных пользователей, реализован контроль доступа к функции создания диалогов, реализована функция создания диалога, реализована отправка сообщений с возможностью прикрепления изображений, осуществлено связывание приложения с базой данных.

### **Список использованных источников**

1. Пример использования диаграмм UML для описание предметной области (документ Microsoft word). Составители: Бахвалова З.А.
2. Анализ ПО\_25.11.16 (презентация). Составители: Бахвалова З.А.
3. Курсовая работа «Разработка прикладного программного обеспечения» Руководство и методические указания для студентов направления: 09.03.02 «Информационные системы и технологии». Составитель Бахвалова З.А. – Иркутск, Изд-во ИРНИТУ, 2018, 61 с.