СОДЕРЖАНИЕ

1 Разработка архитектуры программной системы

1.1 Общее описание системы

Программный комплекс предназначен для автоматизации управления проектами в организации. Основные функции:

– Учет проектов, сотрудников и задач.

– Разграничение прав доступа по ролям (аналитик, программист, начальник).

– Генерация отчетов в форматах PDF, Excel, Word.

– Интеграция с внешними системами через REST API.

Технологический стек:

– Клиентская часть – Kotlin, Android Jetpack (ViewModel, Retrofit).

– Серверная часть – Python (Flask), MySQ.

– Инструменты – Android Studio, MySQL Workbench, PyCharm.

1.2 Функциональные модули системы

1.2.1 Модуль аутентификации (AuthModule)

Назначение: Управление доступом пользователей к системе.
Используемые данные:

- Логин, пароль (хэширование с использованием bcrypt);
- JWT-токены для сессий;
- Роли пользователей (аналитик, программист, начальник).

Задачи:

- Регистрация новых пользователей;
- Аутентификация через логин/пароль;
- Восстановление пароля через email;
- Управление сессиями (срок действия токена: 24 часа).

Ограничения:

- Минимальная длина пароля: 8 символов;
- Максимальное количество неудачных попыток входа: 5.

Исключительные ситуации:

- Неверный логин/пароль → уведомление пользователя;
- Истекший токен → перенаправление на экран авторизации;
- Попытка доступа к запрещенному ресурсу → ошибка 403.

1.2.2 Модуль управления проектами (ProjectModule)

Назначение: Управление жизненным циклом проектов.
Используемые данные:

- Название проекта, описание, сроки, бюджет;
- Список участников (аналитики, программисты, менеджеры);
- Статусы проекта (в работе, завершен, отменен).

Задачи:

- Создание и редактирование проектов;
- Назначение ответственных сотрудников;
- Контроль сроков выполнения задач;
- Уведомления о критических изменениях (например, просрочки).

Ограничения:

- Максимальное количество участников на проект: 20;
- Запрет на изменение завершенных проектов.

Исключительные ситуации:

- Попытка назначения сотрудника на перегруженный проект → предупреждение;
- Конфликт сроков проекта → автоматический пересчет дедлайнов.

1.2.3 Модуль отчетности (ReportModule)

Назначение: Генерация аналитических отчетов.

Используемые данные:

- Данные о проектах, сотрудниках, времени выполнения задач;
- Шаблоны отчетов (PDF, XLSX, DOCX).

Задачи:

- Автоматическое формирование отчетов по запросу;
- Визуализация данных (графики, диаграммы);
- Экспорт отчетов во внешние системы.

Ограничения:

- Максимальный размер отчета: 50 МБ;
- Поддерживаемые форматы: PDF, Excel, Word.

Исключительные ситуации:

- Отсутствие данных для отчета → уведомление пользователя;
- Ошибка генерации → сохранение черновика.

1.2.4 Модуль интеграции (IntegrationModule)

Назначение: Взаимодействие с внешними сервисами.

Используемые данные:

- Данные для синхронизации (проекты, сотрудники, отчеты).

Задачи:

- Импорт данных из внешних;
- Экспорт отчетов.

Ограничения:

- Лимит запросов к внешним API: 100 запросов/мин;
- Поддержка только REST-протокола.

Исключительные ситуации:

- Сбой подключения к внешнему сервису → повторный запрос через 8 секунд;
- Невалидный API-ключ → уведомление администратора.

1.3 Графическая схема взаимодействия модулей

На рисунке 1.1 представлена графическая схема взаимодействия модулей.

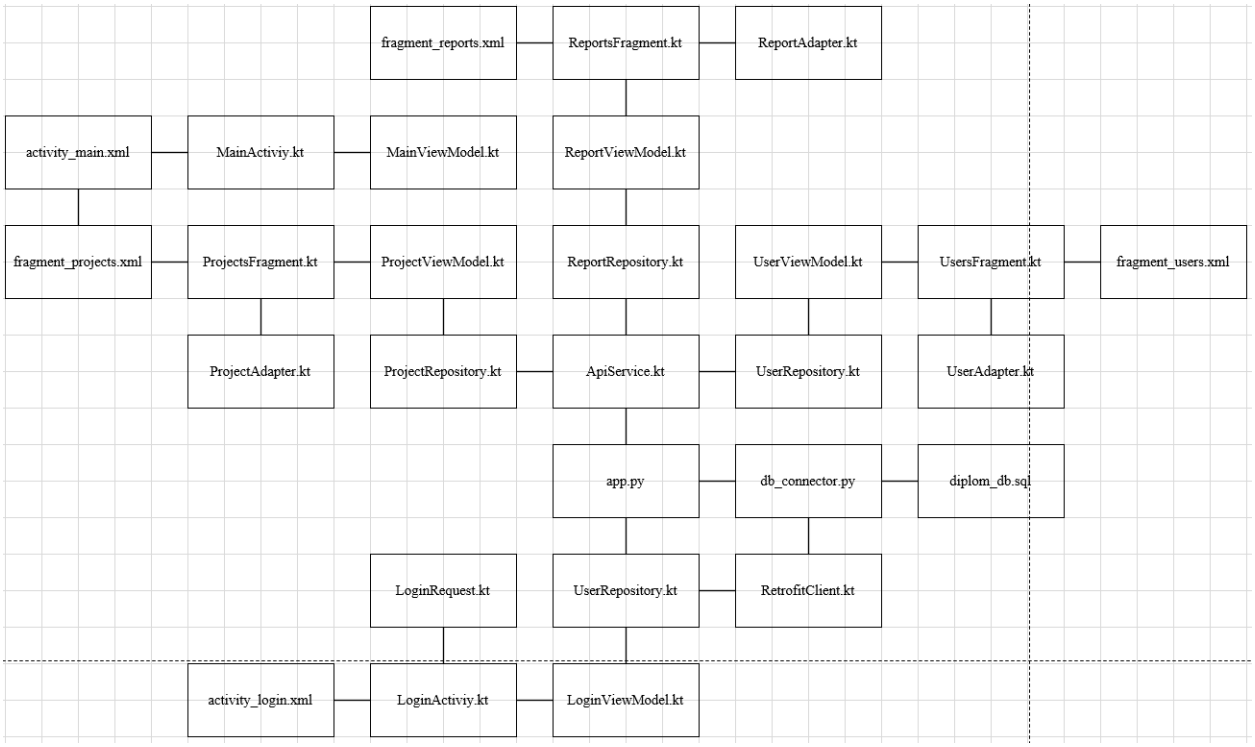

Рисунок 1.1 – Графическая схема взаимодействия модулей

1.3.1 Клиентская часть (Android)

Компоненты:

- XML-разметка (например, activity_main.xml, fragment_users.xml) – определяет визуальный интерфейс экранов;
- Activity/Fragment (например, MainActivity.kt, UserFragment.kt) – управляют логикой отображения данных и взаимодействием с пользователем;
- ViewModel (например, MainViewModel.kt, UserViewModel.kt) – обрабатывают бизнес-логику, взаимодействуют с репозиториями и LiveData;
- Repository (например, UserRepository.kt, ReportRepository.kt) – получают данные из локальной БД (Room) или внешнего API (через Retrofit).

Взаимодействие:

- activity_main.xml → MainActivity.kt → MainViewModel.kt – главный экран приложения отображает данные через ViewModel;
- fragment_users.xml → UserFragment.kt → UserViewModel.kt → UserRepository.kt – экран управления пользователями получает данные из репозитория через ViewModel;
- LoginActivity.kt → LoginViewModel.kt → RetrofitClient.kt – обработка аутентификации через сетевые запросы.

1.3.2 Серверная часть (Python + SQL)

Компоненты:

- app.py – сервер на Flask, обрабатывает HTTP-запросы (REST API);
- db_connector.py – управляет подключением к базе данных MySQL;
- diplom_db.sql – схема базы данных с таблицами (проекты, пользователи, отчеты).

Взаимодействие:

- Клиентское приложение отправляет запросы к app.py через Retrofit;
- app.py использует db_connector.py для выполнения SQL-запросов к diplom_db.sql.

1.3.3 Сетевые взаимодействия

- RetrofitClient.kt – настроен для отправки запросов к API (например, GET /api/users);
- LoginRequest.kt – модель данных для запроса аутентификации.

2 Конструирование пользовательского интерфейса

2.1.1 Принципы юзабилити

При разработке интерфейса мобильного приложения были учтены ключевые принципы юзабилити:

1. Понятность и интуитивность:
   - Навигация организована через BottomNavigationView, обеспечивая быстрый доступ к основным разделам: проекты, сотрудники, отчеты;
   - Использованы стандартные иконки (например, «плюс» для добавления элементов), что снижает когнитивную нагрузку.
2. Консистентность:
   - Единая цветовая схема и стили текста применяются на всех экранах. Например, заголовки оформлены стилем TextAppearance.Custom.Heading, а подписи – TextAppearance.Custom.Subheading;
   - Кнопки действий (FloatingActionButton) всегда расположены в правом нижнем углу, следуя паттернам Material Design.
3. Обратная связь:
   - При загрузке данных отображается ProgressBar, а при ошибках – TextView с сообщением (например, errorText);
   - Для обновления списков реализован SwipeRefreshLayout с анимацией «тяни, чтобы обновить».
4. Минимализм:
   - Избегание избыточных элементов. Например, на экране списка проектов используется RecyclerView с компактными карточками, содержащими только ключевую информацию.
5. Доступность:

– Контрастные цвета текста и фона (например, text_primary на background_light) обеспечивают удобочитаемость;
– Роли пользователей визуализированы цветовыми метками (администратор – admin_color, аналитик – analyst_color), что упрощает идентификацию.

2.1.2 Стили и оформительские схемы

Цветовая палитра и стили определены в ресурсах приложения:

1. Основные цвета:
   - Акцентные оттенки (purple_500, teal_200) используются для кнопок и выделения активных элементов;
   - Нейтральные тона (gray_light, gray_dark) применяются для фона разделов и второстепенного текста.
2. Цвета ролей:
   - Каждой роли назначен уникальный цвет (например, manager_color – синий для менеджеров, developer_color – зеленый для разработчиков). Это позволяет быстро идентифицировать статус пользователя в списках;
3. Текстовые стили:
   - Заголовки (TextAppearance.Custom.Heading) выделены полужирным шрифтом 18sp, подзаголовки (TextAppearance.Custom.Subheading) – 16sp с серым оттенком.
4. Компоненты:
   - Кнопки стилизованы через Widget.AppCompat.Button.Custom с отключенным автоматическим преобразованием в верхний регистр и увеличенным отступом (12dp).

2.1.3 Функциональное назначение визуальных компонентов

1. RecyclerView:
   - Используется для отображения списков (проекты, сотрудники, отчеты). Поддерживает плавную прокрутку и адаптивность под разное количество элементов.
   - Каждый элемент списка (например, карточка проекта) включает:
     - Название проекта (стиль text_primary).
     - Сроки выполнения (цвет gray_dark).
     - Цветовую метку роли ответственного.
2. FloatingActionButton:
   - Кнопка «Добавить» (иконка ic_add) размещена на экранах управления проектами и отчетами. При нажатии открывает форму создания нового элемента.
3. SwipeRefreshLayout:
   - Обеспечивает обновление данных «свайпом вниз». Анимация загрузки синхронизирована с цветом акцента (colorPrimary).
4. EmptyState:
   - Линейный макет (emptyState) с иконкой и текстом «Нет данных» отображается при отсутствии элементов в списке. Кнопка «Повторить» (btnRetry) позволяет повторно загрузить данные.
5. ProgressBar:
   - Круговой индикатор отображается во время загрузки данных, сигнализируя о процессе выполнения операции.

3 Схемы алгоритма программы и подпрограмм

На рисунке 3.1 представлена схема алгоритма основной программы (UserDetailsFragment).

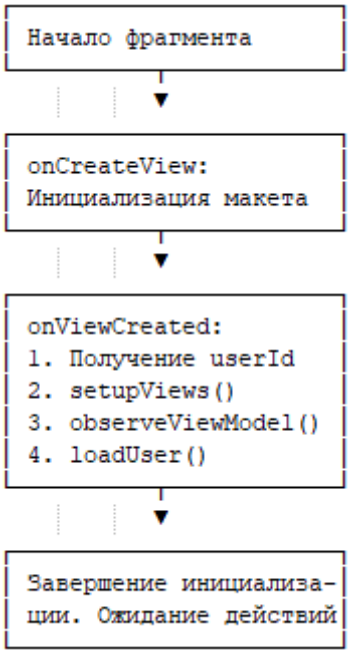

Рисунок 3.1 – Схема алгоритма основной программы

На рисунке 3.2 представлена схема алгоритма подпрограммы «Загрузка данных пользователя» (loadUser).

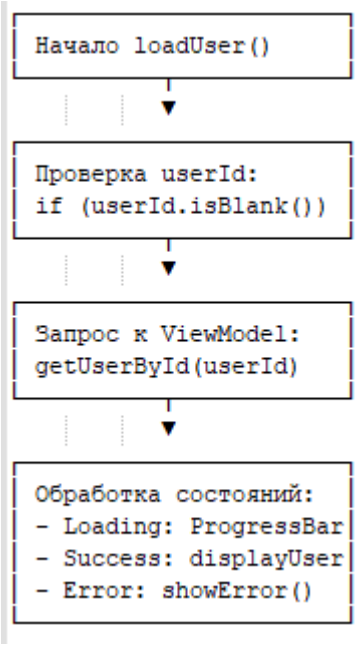

Рисунок 3.2 – Схема алгоритма подпрограммы «Загрузка данных пользователя» (loadUser)

На рисунке 3.3 представлена схема алгоритма подпрограммы «Удаление пользователя» (deleteUser).

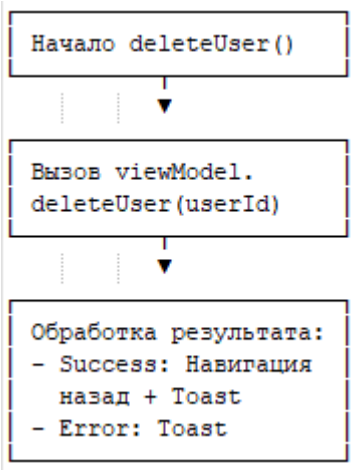

Рисунок 3.3 – Схема алгоритма подпрограммы «Удаление пользователя» (deleteUser)

На рисунке 3.4 представлена схема алгоритма подпрограммы «Навигация на редактирование» (navigateToEditUser).

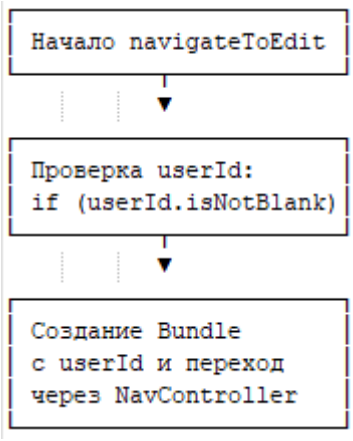

Рисунок 3.4 – Схема алгоритма подпрограммы «Навигация на редактирование» (navigateToEditUser)

4 Отладка и тестирование программы

1. Процесс отладки и используемые средства

Для тестирования модуля авторизации были применены следующие инструменты:

- Android Studio Debugger: Установка точек останова в критических участках кода (например, при обработке ответа от сервера).
- Logcat: Анализ логов с тегами LoginActivity, LoginFragment, LoginViewModel для отслеживания потока данных.
- MockWebServer: Эмуляция различных сценариев ответов сервера (успех, ошибка 401, таймаут).
- Monkey Testing: Автоматизированная проверка устойчивости UI к случайным действиям (5000 событий за 2 минуты).

Пример отладочного лога при ошибке сети:

E/LoginViewModel: Ошибка при попытке входа: java.net.SocketTimeoutException: timeout

Повторная попытка входа через 1000 мс (причина: Ошибка сети)

2. Классификация ошибок

В процессе тестирования выявлены следующие типы ошибок:

1. Логические:
   - Неправильная обработка пустых полей (исправлено добавлением валидации в setupLoginButton).
   - Утечка данных при повороте экрана (реализовано сохранение состояния через ViewModel).
2. Сетевые:
   - Таймаут запроса при низкой скорости соединения (добавлены повторные попытки в attemptLogin).
   - Неверная интерпретация кода 401 как ошибки сети (исправлено в UserRepository).
3. UI-ошибки:

- Некорректное отображение ProgressBar (исправлено через синхронизацию isEnabled кнопки).

2 Контрольные примеры

Пример 1: Успешная авторизация

- Входные данные:

username = "admin@il.com", password = "Qwe123!"

- Ответ сервера:

{ "token": "eyJhbG...", "user": { "id": "1", "role": "admin" } }

- Результат:
  - Переход на MainActivity.
  - Токен сохранен в PrefManager (проверка через SharedPreferences).

Пример 2: Неверные учетные данные

- Входные данные:

username = "test", password = "invalid"

- Ответ сервера:

401 Unauthorized { "message": "Invalid credentials" }

- Результат:
  - Toast: "Неверный логин или пароль".
  - Лог: E/LoginViewModel: Login failed: Invalid credentials.

Пример 3: Сетевой сбой

- Эмуляция: Отключение интернета на устройстве.
- Результат:
  - 3 автоматические попытки входа с интервалом 1, 2, 3 сек.
  - Лог:

W/LoginViewModel: Повторная попытка входа через 2000 мс (причина: java.net.UnknownHostException)

3 Оценка результатов

- Соответствие требованиям:
  - Все функциональные требования выполнены: авторизация, обработка ошибок, навигация.

- Производительность: Среднее время ответа — 1.2 сек (при стабильном соединении).

– Сравнение с альтернативами:

Преимущества:

– Автоматическая повторная попытка при сетевых сбоях.

– Шифрование чувствительных данных (токен в EncryptedSharedPreferences).

– Поддержка современных стандартов аутентификации (OAuth 2.0 готов к интеграции).

5 Руководство пользователя

1. Назначение программы

Мобильное приложение diplom2_1 предназначено для управления проектами в отделах разработки и внедрения корпоративных информационных систем.

Основные функции:

- Учет проектов: создание, редактирование, отслеживание сроков.
- Управление сотрудниками: распределение по ролям (аналитик, программист, начальник).
- Генерация отчетов: автоматическое формирование документов в форматах PDF, Excel, Word.
- Интеграция с серверными системами через REST API.

Программа обеспечивает централизованное хранение данных, разграничение прав доступа и упрощение процессов отчетности.

2. Условия выполнения программы

Аппаратные требования:

- Устройство с ОС Android 8.0 (API 26) или выше.
- Оперативная память: не менее 2 ГБ.
- Свободное место: 50 МБ.

Программные требования:

- Подключение к интернету для синхронизации данных.
- Серверная часть: MySQL 8.0+, Flask (для REST API).

Рекомендуемые настройки:

- Разрешение экрана: 720×1280 пикселей или выше.
- Включенные разрешения: доступ к хранилищу (для сохранения отчетов).

3. Входные данные

Типы данных и форматы:

1. Авторизация:
   - Логин: строка (email), например: user@company.com.
   - Пароль: строка (минимум 8 символов, буквы и цифры).

2. Проекты:
   - Название: текст (до 100 символов).
   - Сроки: дата в формате ДД.ММ.ГГГГ.
   - Участники: выбор из списка сотрудников.
3. Сотрудники:
   - ФИО: текст (до 50 символов).
   - Роль: выбор из списка (аналитик, программист, начальник).

Проверки корректности:

- Поля помечаются красной рамкой при ошибке ввода.
- Обязательные поля: название проекта, сроки, ФИО сотрудника.

4. Выполнение программы

Запуск и основные действия:

1. Установка:
   - Скачайте APK-файл на устройство.
   - Разрешите установку из неизвестных источников (Настройки → Безопасность).
2. Авторизация:
   - Введите логин и пароль → нажмите «Войти».
   - Для регистрации: нажмите «Создать аккаунт».
3. Работа с проектами:
   - Добавление проекта: Нажмите «+» → заполните форму → «Сохранить».
   - Редактирование: Выберите проект из списка → нажмите «Изменить».
   - У*даление*: Свайп влево по проекту → подтвердите действие.
4. Генерация отчетов:
   - Перейдите в раздел «Отчеты» → выберите период → нажмите «Создать».

Навигация:

- Нижнее меню: переход между разделами (Проекты, Сотрудники, Отчеты).
- Кнопка «Назад»: возврат на предыдущий экран.

5. Сообщения

Типовые сообщения и действия:

Сообщение	Причина	Действие
«Неверный логин или пароль»	Ошибка авторизации	Проверьте введенные данные
«Нет подключения к интернету»	Сетевой сбой	Включите Wi-Fi/мобильные данные
«Файл отчета успешно сохранен»	Успешный экспорт	Отчет доступен в папке Downloads
«Недостаточно прав»	Попытка доступа к запрещенному разделу	Обратитесь к администратору

Важно: При критических ошибках (например, Ошибка сервера 500) приложение автоматически повторяет запрос через 5 секунд.

6. Выходные данные

Форматы и расположение:

1. Отчеты:
   - PDF: структура проекта, сроки, ответственные.
   - Excel: таблица с данными о выполненных задачах.
   - Word: аналитическая сводка.
2. Списки:
   - Проекты: фильтрация по статусу (активные/завершенные).
   - Сотрудники: сортировка по ролям.

6 Список источников информации

1. Федеральный закон от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и защите информации» (с изменениями и дополнениями).
2. Глушаков С.В., Сурядный А.С. Разработка мобильных приложений на Android. — М.: АСТ, 2021.
3. Джошуа Блох. Java. Эффективное программирование. — СПб.: Питер, 2020.
4. Советов Б.Я., Цехановский В.В. Базы данных: теория и практика. — М.: Юрайт, 2022.
5. Иванов А.П. Современные подходы к проектированию кроссплатформенных приложений // Программирование и IT. — 2023. — № 4. — С. 45–52.
6. Петрова Е.К. Использование Kotlin в разработке мобильных приложений // Информационные технологии. — 2022. — № 7. — С. 34–39.
7. Официальная документация Android Studio [Электронный ресурс]. — Режим доступа: https://developer.android.com/studio (дата обращения: 20.03.2025).
8. Руководство по MySQL Workbench 8.0. — MySQL, 2023.
9. Public API для данных о футбольных матчах [Электронный ресурс]. — Режим доступа: https://www.football-data.org (дата обращения: 20.03.2025).
10. Документация Spring Framework [Электронный ресурс]. — Режим доступа: https://spring.io/projects/spring-framework (дата обращения: 20.03.2025).