**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ**

**ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**КУРСОВОЙ ПРОЕКТ**

### НА ТЕМУ

|  |  |
| --- | --- |
| РАЗРАБОТКА ПОДСИСТЕМЫ УЧЁТА | |
| МАТЕРИАЛОВ И СПЕЦТЕХНИКИ ДЛЯ | |
| АВТОДОРОГ | |
| Л109. 24КП01. 016 ПЗ | |

(Обозначение документа)

|  |
| --- |
| МДК.02.01 Технология разработки |
| программного обеспечения |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИСПП-11 | | 05.12.2024 | И.М.Пономарев |
|  | (Группа) | (Подпись) | (Дата) | (И.О. Фамилия) |
| Преподаватель |  |  | 05.12.2024 | Ю.С. Маломан |
|  |  | (Подпись) | (Дата) | (И.О. Фамилия) |

Архангельск 2024

**СОДЕРЖАНИЕ**

Перечень сокращений и обозначений 3

Введение 4

1 Анализ и разработка требований 6

1.1 Назначение и область применения 6

1.2 Постановка задачи 6

1.3 Описание алгоритма функционирования системы 6

1.4 Выбор состава программных и технических средств 7

2 Проектирование программного обеспечения 9

2.1 Проектирование интерфейса пользователя 9

2.2 Разработка архитектуры программного обеспечения 10

2.3 Проектирование базы данных 11

3 Разработка и интеграция модулей программного обеспечения 12

3.1 Разработка программных модулей 12

3.2 Реализация интерфейса пользователя 17

3.3 Разграничение прав доступа пользователей 19

3.4 Экспорт и импорт данных 20

4 Тестирование и отладка программного обеспечения 21

4.1 Структурное тестирование 21

4.2 Функциональное тестирование 22

5 Инструкция по эксплуатации программного обеспечения 24

5.1 Установка программного обеспечения 24

5.2 Инструкция по работе 25

Заключение 29

Список использованных источников 30

**ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ**

В настоящем курсовом проекте применяют следующие сокращения и обозначения:

3НФ – третья нормальная форма

БД – база данных

ИС – информационная система

ОЗУ – оперативное запоминающее устройство

ОС – операционная система

ПК – персональный компьютер

ПО – программное обеспечение

СУБД – система управления базами данных

ER - entity-relationship model, модель «сущность — связь»

IDE - Integrated Development Environment, интегрированная среда разработки

MS – Microsoft

DVD – формат оптических носителей последнего поколения

CD – оптический носитель информации в виде пластикового диска с отверстием в центре, процесс записи и считывания информации которого осуществляется при помощи лазера

**ВВЕДЕНИЕ**

Актуальность курсового проекта заключается в том, что применение баз данных в разработке подсистемы учёта материалов и спецтехники для автодорог позволяет значительно повысить эффективность управления ресурсами и улучшить планирование. В условиях современного строительства и эксплуатации автодорог, где объемы материалов и техники могут быть значительными, важно иметь систему, которая обеспечит точный учёт, оперативный доступ к информации и возможность анализа данных.

На многих предприятиях по-прежнему используются устаревшие методы учёта, такие как табличные редакторы или общие бухгалтерские программы. Хотя эти инструменты могут предоставлять базовые функции для создания расчётов и ведения учёта, они не способны обеспечить необходимую гибкость и актуальность информации для всех участников процесса. В результате, подрячики и другие заинтересованные стороны часто сталкиваются с проблемами, связанными с недостаточной прозрачностью данных, задержками в получении информации и трудностями в анализе текущего состояния ресурсов.

Целью курсового проектирования является разработка многопользовательской клиент-серверной веб-приложение для учета материалов и спецтехники. Эта система должна обеспечить удобный интерфейс для пользователей, а также надежное хранение и обработку данных.

Для достижения поставленной цели требуется решить следующие задачи:

* выполнить сбор требований целевой аудитории,
* проанализировать информационные источники по предметной области,
* спроектировать архитектуру приложения,
* спроектировать диаграмму вариантов использования приложения,
* выбрать состав программных и технических средств для реализации мобильного приложения,
* спроектировать БД,
* создать БД в MySQL Workbench,
* реализовать разграничение прав доступа пользователей,
* реализовать защиту данных,
* разработать интерфейс веб-приложение,
* разработать веб-приложение,
* реализовать экспорт данных в виде файлов .xlsx,
* реализовать работу веб-приложения по средством контроллеров ASP.NET,
* выполнить структурное тестирование ПО,
* выполнить функциональное тестирование ПО,
* разработать программную и эксплуатационную документацию.

В результате выполнения поставленных задач будет разработано веб-приложение для учёта ресурсов для автодорог.

**1 Анализ и разработка требований**

* 1. **Назначение и область применения**

ИС предназначена для автоматизации процессов управления ресурсами в сфере строительства и эксплуатации автодорог. Основное назначение данной ИС заключается в обеспечении точного учёта, мониторинга данных о проектах, что позволяет повысить эффективность работы организации, улучшить планирование и сократить затраты.

Разрабатываемая ИС ориентирована на проектную организацию, занимающуюся разработкой проектной документации для строительства и ремонта автодорог.

* 1. **Постановка задачи**

Необходимо разработать веб-приложение, которое предоставит доступ к следующей функциональности:

* авторизации пользователей,
* регистрации новых пользователей,
* просмотру и редактировании информации о проектах, их задачах и требуемых ресурсов для выполнения задач,
* просмотру и редактировании информации о имеющихся пользователей в ИС,
* генерации сметы по проекту в файле .xslx,
* публикации проектов для подрячиков.
  1. **Описание алгоритма функционирования системы**

При запуске приложения отображается страница с опубликованными проектами, в верхней панели приложения находится кнопка аутентификации, после нажатия на кнопку открывается модальное окно аутентификации с возможностью авторизоваться, как администратор и как проектировщик.

Неавторизованный пользователь может открыть проект, нажав на него, где будут указаны задачи, которые можно аналогично открыть и просмотреть подробную информацию с указанием требуемых ресурсов.

Проектировщик может к тому, что может делать неавторизованный пользователь, редактировать информацию о проектах и их составляющих (при указании для задачи ресурсов можно указать существующий или указать новый), генерировать смету по проекту в файл .xlsx.

Администратор может к тому, что может делать проектировщик, редактировать информацию о ресурсах и пользователях.

На рисунке 1 изображена диаграмма вариантов использования приложения различными категориями пользователей.

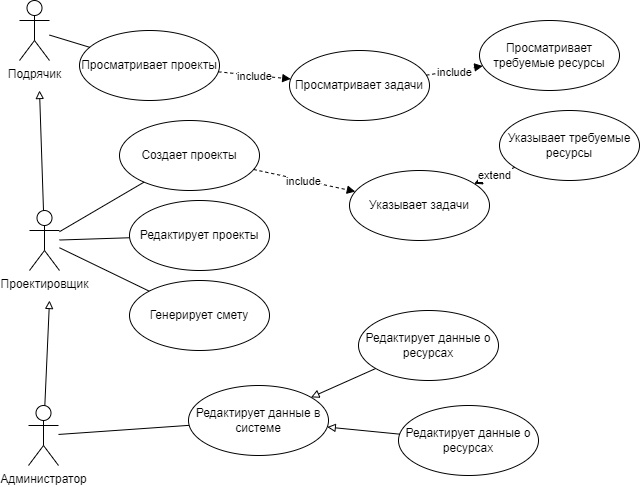


Рисунок 1 – Диаграмма вариантов использования

* 1. **Выбор состава программных и технических средств**

Согласно цели проекта требуется создать веб-приложение для учета материалов и спецтехники.

Работа с ИС будет осуществляться на мобильных устройствах и персональных компьютерах, которые имеют браузер.

В качестве СУБД выбрана MySQL версии не ниже 8.0. MySQL была выбрана благодаря своей высокой производительности, надежности и широкому распространению. Она поддерживает многопользовательский доступ, что является важным для нашей системы, а также предоставляет мощные инструменты для работы с данными, включая возможности для создания сложных запросов и оптимизации производительности. Кроме того, MySQL имеет активное сообщество и обширную документацию, что упрощает процесс разработки и устранения возможных проблем. и индексации, поддерживает различные типы данных, включая JSON.

Приложение будет написано на языке программирования C#, который был выбран благодаря своей сильной типизации, поддержке объектно-ориентированного программирования и богатой стандартной библиотеке. C# обеспечивает высокую надежность и стабильность за счет обнаружения ошибок на этапе компиляции, а также позволяет создавать модульные и легко поддерживаемые приложения.

Для разработки приложения будет использоваться интегрированная среда разработки Visual Studio 2022. Эта IDE была выбрана благодаря своему мощному функционалу, удобному интерфейсу и отличной поддержке языка C#. Visual Studio 2022 предлагает инструменты для отладки, тестирования и профилирования, а также интеграцию с системами контроля версий и доступ к множеству расширений, что делает процесс разработки более эффективным и продуктивным.

Для функционирования системы на стороне сервера достаточны следующие программные и технические средства:

* Windows Server 2016 или более поздняя версия, либо Linux (например, Ubuntu 20.04 LTS)
* сервер БД: MySQL не ниже 6.0,
* процессор минимум 2 ядра (рекомендуется 4 и более),
* свободная оперативная память объемом 1 ГБ,
* ПО для конфигурирования, управления и администрирования сервера БД: MySQL Workbench версия 8.0,
* места на диске минимум 20 ГБ.

Для функционирования системы на стороне клиента достаточны следующие программные и технические средства:

* любая ОС, поддерживающая современные веб-браузеры,
* процессор с частотой 1,6 ГГц,
* оперативная память в объеме 1 ГБ,
* постоянное интернет-подключение.

**2 Проектирование программного обеспечения**

**2.1 Разработка архитектуры программного обеспечения**

Архитектура приложения построена на основе клиент-серверной модели и включает в себя серверную часть, веб-приложение и БД. Все компоненты будут развернуты в контейнерах Docker.

Для серверной части будет создан API, позволяющий клиенту взаимодействовать с БД. Она будет использовать СУБД MySQL [1], которая будет хранить информацию о пользователях, проектах, задачах, материалах, спецтехнике и рабочих.

Веб-приложение будет написано на фреймворке ASP.NET. Взаимодействие с сервером будет происходить при помощи HTTP-запросов к RESTful API, ответы будут приходить в формате JSON. Для отправки запросов будет использоваться веб-клиент ASP.NET [2]. Архитектура системы отображена на диаграмме развертывания на рисунке 2

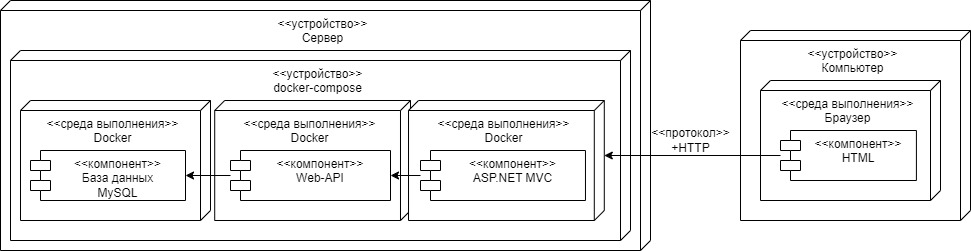


Рисунок 3 – Диаграмма развертывания

**2.2 Проектирование базы данных**

Требуется разработать БД для системы подбора персонала. Система будет использоваться заказчиками в виде физических или юридических лиц [3].

Модели БД созданы при помощи MySQL Workbench. На рисунке 3 в виде ERD показана часть физической модели предметной области, связанной с поиском экспертов для созданной задачи.

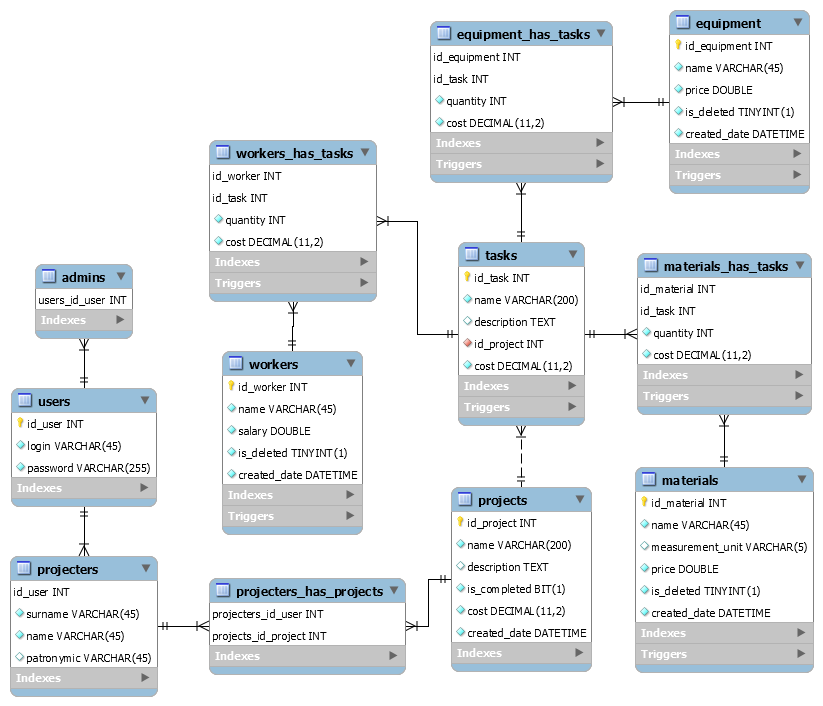


Рисунок 3 – MySQL Workbench. Физическая модель БД

**2.3 Проектирование интерфейса пользователя**

В рамках разработки веб-приложения создан интерфейс пользователя в виде мокапов при помощи инструмента Figma. Эти визуальные представления позволяют наглядно увидеть структуру приложения, его основные элементы и функциональность.

Мокапы главной страницы с проектами, страницы с материалами, страницы одного из проектов интерфейса для роли Администратора веб-приложения представлены на рисунке 4.

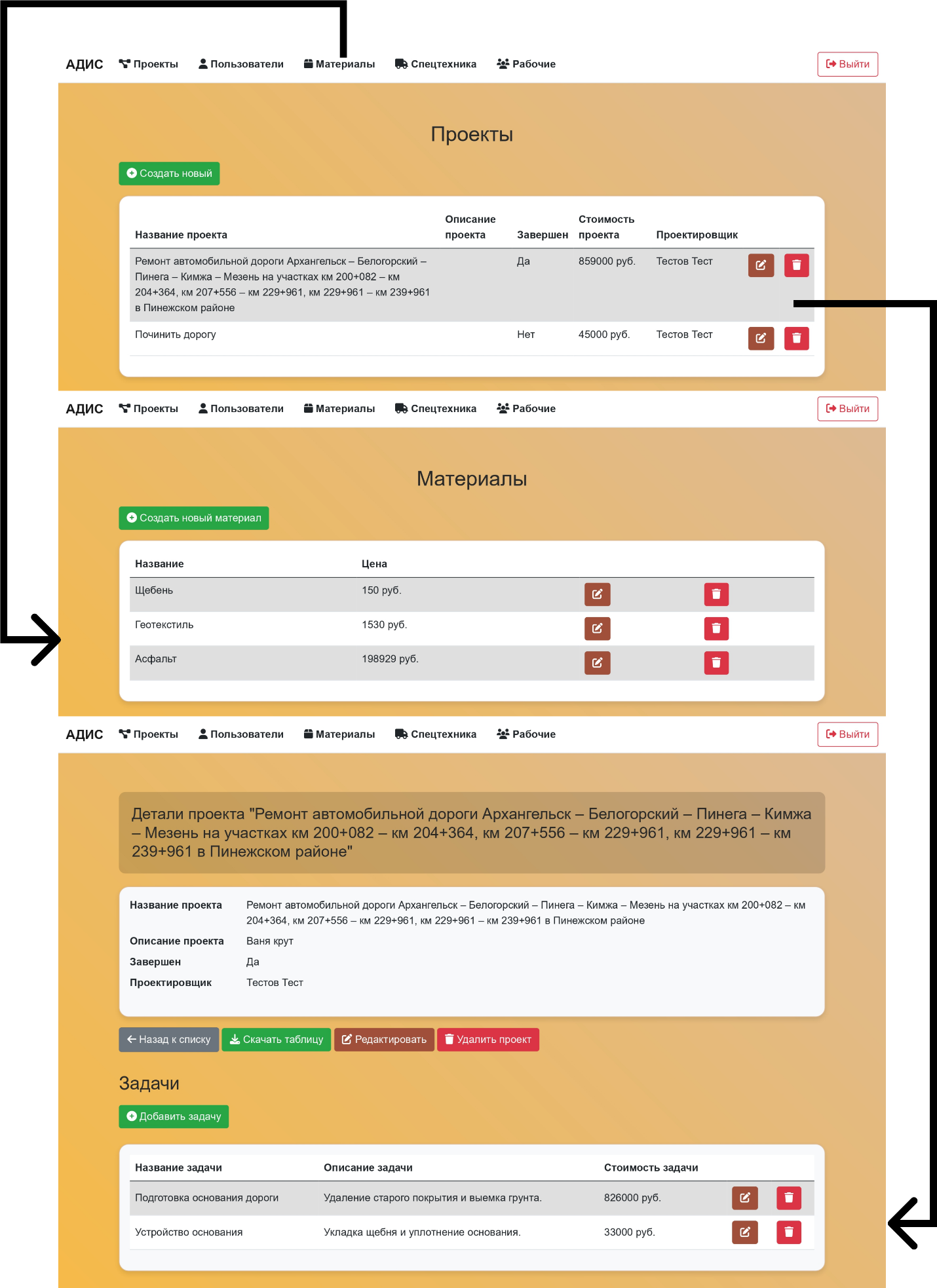


Рисунок 4 – Figma. Мокапы интерфейса пользователя мобильного приложения

**3 Разработка и интеграция модулей программного обеспечения**

**3.1 Разработка программных модулей**

Для курсового проекта разработано веб-приложение на С# с использованием фреймворка ASP.NET в Visual Studio [3].

Взаимодействие мобильного приложения с сервером будет происходить при помощи HTTP-запросов к API, а ответы будут возвращаться в формате JSON. Для реализации HTTP-запросов используется сетевой клиент, который позволяет отправлять запросы и получать ответы от сервера. Код метода для получения списка проектов путем отправки GET-запроса на сервер представлен листингом 1.

Листинг 1 – Код метода для отправки GET-запроса на сервер

|  |
| --- |
| // Получение страницы с проектами  // GET: Projects  public async Task<IActionResult> Index()  {  // Проверяем, аутентифицирован ли пользователь  if (User.Identity.IsAuthenticated)  {  // Если пользователь имеет роль "Admin"  if (User.IsInRole("Admin"))  // Получаем список всех проектов и возвращаем представление с этими данными  return View(await \_httpClientService.GetHttpClient().GetFromJsonAsync<List<Project>>("Projects"));  // Если пользователь не администратор, получаем проекты, связанные с конкретным пользователем  return View(await \_httpClientService.GetHttpClient().GetFromJsonAsync<List<Project>>($"Projects?idProjecter={Int32.Parse(User.FindFirst(ClaimTypes.NameIdentifier).Value)}"));  }  else  {  // Если пользователь не аутентифицирован, получаем только завершенные проекты  return View(await \_httpClientService.GetHttpClient().GetFromJsonAsync<List<Project>>("Projects?isCompleted=true"));  }  } |

Код представления в API серверной части для поиска материалов по имени представлен листингом 2.

Листинг 2 – Код представления для поиска материалов по имени в API

|  |
| --- |
| // GET: api/Materials  [HttpGet]  public async Task<ActionResult<IEnumerable<Material>>> GetMaterials([FromQuery] string? name, [FromQuery] bool? isDeleted)  {  var query = \_context.Materials.AsQueryable();  // Фильтрация по имени, если оно указано  if (!string.IsNullOrEmpty(name))  {  query = query.Where(m => m.Name == name);  }  // Фильтрация по isDeleted, если оно указано  if (isDeleted.HasValue)  {  query = query.Where(m => m.IsDeleted == isDeleted.Value);  }  // Получение списка материалов  return await query.ToListAsync();  } |

**3.2 Реализация интерфейса пользователя**

Интерфейс разработан с использованием постраничной навигации и различных элементов управления, что упрощает взаимодействие пользователя с приложением. В данном разделе представлено представление, которое позволяет пользователю вводить информацию о новой спецтехнике.

На странице реализована форма, которая включает поля для ввода названия, цены и количества спецтехники. Для улучшения пользовательского опыта добавлено автозаполнение для поля "Название", которое срабатывает при вводе текста. Также предусмотрены механизмы валидации, которые отображают ошибки, если введенные данные не соответствуют требованиям.

Код представления для добавления спецтехники представлен листингом 3.

Листинг 3 – Код виджета для SearchExpertItem

|  |
| --- |
| @model AutodorInfoSystem.Models.Equipment  @{  ViewData["Title"] = "Добавление спецтехники"; // Устанавливаем заголовок страницы  }  <h1 class="mb-4">Добавление спецтехники</h1> <!-- Заголовок страницы -->  <div class="card p-4"> <!-- Контейнер для формы с отступами -->  <div class="row"> <!-- Строка для размещения элементов формы -->  <div class="col"> <!-- Колонка для формы -->  <form asp-action="Create"> <!-- Указываем действие контроллера для обработки формы -->  <div asp-validation-summary="ModelOnly" class="text-danger"></div> <!-- Отображение ошибок валидации модели -->    <div class="form-group"> <!-- Группа для поля "Название" -->  <label asp-for="Name" class="control-label">Название</label> <!-- Метка для поля "Название" -->  <input asp-for="Name" class="form-control" id="equipmentName" onkeyup="fetchSimilarNames()" /> <!-- Поле ввода для названия с автозаполнением -->  <span asp-validation-for="Name" class="text-danger"></span> <!-- Отображение ошибок валидации для поля "Название" -->  <ul id="suggestions" class="list-group mt-2" style="display:none;"></ul> <!-- Список для автозаполнения, скрыт по умолчанию -->  </div>    <div class="form-group"> <!-- Группа для поля "Цена" -->  <label asp-for="Price" class="control-label">Цена</label> <!-- Метка для поля "Цена" -->  <input asp-for="Price" class="form-control" id="equipmentPrice" /> <!-- Поле ввода для цены -->  <span asp-validation-for="Price" class="text-danger"></span> <!-- Отображение ошибок валидации для поля "Цена" -->  </div>    <div class="form-group"> <!-- Группа для поля "Количество" -->  <label asp-for="Quantity" class="control-label">Количество</label> <!-- Метка для поля "Количество" -->  <input asp-for="Quantity" class="form-control" /> <!-- Поле ввода для количества -->  <span asp-validation-for="Quantity" class="text-danger"></span> <!-- Отображение ошибок валидации для поля "Количество" -->  </div>    <input type="hidden" name="IdTask" value="@ViewBag.IdTask" /> <!-- Скрытое поле для передачи идентификатора задачи -->    <div class="form-group"> <!-- Группа для кнопки отправки формы -->  <button class="btn btn-primary" type="submit"> <!-- Кнопка для отправки формы -->  <i class="fas fa-check"></i> <!-- Иконка для кнопки -->  Создать  </button>  </div>    <div class="mt-3"> <!-- Контейнер для кнопки "Назад" -->  <a asp-controller="Tasks" asp-action="Details" asp-route-id="@ViewBag.IdTask" class="btn btn-outline-primary"> <!-- Ссылка для возврата на страницу деталей задачи -->  <i class="fas fa-arrow-left"></i> <!-- Иконка для ссылки -->  Назад  </a>  </div>  </form>  </div>  </div>  </div>  @section Scripts { <!-- Раздел для подключения скриптов -->  <script src="~/js/create-and-edit-equipment.js"></script> <!-- Подключение скрипта для автозаполнения и других функций -->  } |

**3.3 Разграничение прав доступа пользователей**

JWT В мобильном приложении разработано разграничение прав доступа пользователей. Для этого в приложении реализованы авторизация и регистрация. Пользователь с ролью заказчика под названием «Ищу эксперта», может производить поиск экспертов, создавать задачи, а также заполнять информацию в профиле про компанию или свою деятельность. Пользователь с ролью «Эксперт», заполняет в профиле информацию про образование и опыт. Пример разграничения прав доступа пользователей можно увидеть на примере заполнения информации на странице «Редактирование профиля» на рисунке 7.

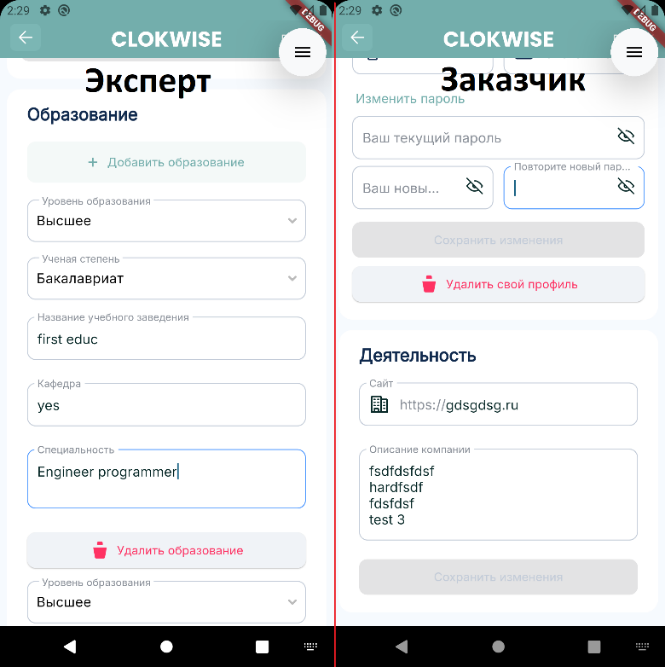


Рисунок 7 – Clokwise. Вид страницы «Редактирование профиля»

**3.4 Экспорт и импорт данных**

Форматы файлов

В мобильном приложении реализован экспорт и импорт данных при помощи библиотек flutter\_downloader и file\_picker.

Пример экспорта данных каких? можно увидеть на листинге 4.

Листинг 4 – Код экспорта данных

|  |
| --- |
| final data = dio.FormData.fromMap({  'file': dio.MultipartFile.fromBytes(  image.bytes,  filename: 'image.${image.fileTypeInfo.extension}',  contentType: MediaType('image', image.fileTypeInfo.extension),  ),  });  final response = await \_mainNetworkClient.client.post(  '$\_path/upload-attachment/$chatId/',  data: data,  ); |

Пример импорта данных можно увидеть на листинге 5.

Листинг 5 – Код импорта данных

|  |
| --- |
| final taskId = await FlutterDownloader.enqueue(  url: '${AppUrls.path}/$fileId/',  headers: {'Authorization': 'Bearer $authToken'},  savedDir: directory.path,  saveInPublicStorage: true,  showNotification: true,  openFileFromNotification: true, // click on notification  // to open downloaded file (for Android)  );  FlutterDownloader.registerCallback(downloadCallback);  final tasks = await FlutterDownloader.loadTasks(); |

1. **Тестирование и отладка программного обеспечения**

**4.1 Структурное тестирование**

Во время курсового проектирования проведено структурное тестирование для search\_repository [5]. Для него использована библиотека mockito для создания имитации поведение зависимостей для репозитория и библиотека flutter\_test. Код unit-теста для поиска экспертов представлен листингом 3.

Листинг 6 – Код unit-теста для search\_repository

|  |
| --- |
| void main() {  late MockSearchRepository mockSearchRepository;  late MockSecureStorageManager mockSecureStorageManager;  late MockMainNetworkClient mockMainNetworkClient;  TestWidgetsFlutterBinding.ensureInitialized();  setUp(() {  mockSecureStorageManager = MockSecureStorageManager();  mockMainNetworkClient = MockMainNetworkClient();  mockSearchRepository = MockSearchRepository();  });  group('SearchRepository', () {  test('postSearchResult returns list of experts on success', () async {  final searchRequest = SearchRequest(industry: 1, experience: 0);  final result = await mockSearchRepository.postSearchResultMock(searchRequest);  expect(result, isA<List<ExpertResultModel>>());  expect(result.length, 2);  expect(result[0].id, 1);  expect(result[0].firstName, 'Expert 1');  expect(result[1].id, 2);  expect(result[1].firstName, 'Expert 2');  });  test('postSearchResult throws exception on error', () async { |
| final searchRequest = SearchRequest(prompt: 'invalid', industry: 1, experience: 0);  try {  // Используем метод мока для выбрасывания исключения  await mockSearchRepository.throwExceptionOnPostSearchResult(searchRequest);  } catch (e) {  // Выводим информацию об исключении  print('Caught an exception: $e');  // Проверяем, что исключение действительно произошло  expect(e, isA<Exception>());  }  });  });  } |

На рисунке 8 изображена консоль с результатами тестирования, где отображается информация о перехваченных исключениях и успешном тестировании.

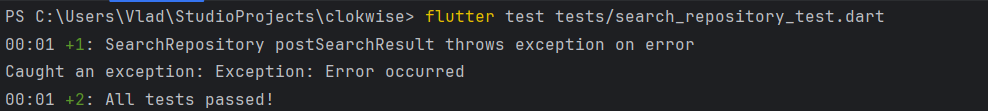


Рисунок 8 – Android Studio. Результаты unit-тестирования

**4.2 Функциональное тестирование**

Во время курсового проектирования проведено функциональное тестирование при помощи виджет-тестирования из библиотеки integration\_test.

Листинг 7 – Код интеграционного тестирования для страницы с авторизацией

|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:flutter\_test/flutter\_test.dart';  import 'package:integration\_test/integration\_test.dart';  import 'package:clokwise/main.dart' as app;  void main() { |
| // Инициализация интеграционного тестирования.  IntegrationTestWidgetsFlutterBinding.ensureInitialized();  /// Тест для проверки функциональности входа в систему.  ///  /// Этот тест выполняет следующие шаги:  /// 1. Запускает приложение.  /// 2. Находит элементы интерфейса для ввода имени пользователя и пароля.  /// 3. Вводит тестовые данные (имя пользователя и пароль).  /// 4. Нажимает кнопку входа.  /// 5. Проверяет, что после успешного входа отображается имя пользователя.  testWidgets("Login Test", (WidgetTester tester) async {  // Запускаем основное приложение.  app.main();  // Дожидаемся завершения анимаций и обновления интерфейса.  await tester.pumpAndSettle();  // Находим кнопку входа по ключу.  final menuButton = find.byKey(ValueKey('enterButton'));  // Проверяем, что кнопка входа отображается на экране.  expect(menuButton, findsOneWidget);  // Находим поле для ввода имени пользователя и пароля по ключам.  final usernameField = find.byKey(ValueKey('emailTextField'));  final passwordField = find.byKey(ValueKey('passwordTextField'));  // Вводим тестовый адрес электронной почты.  await tester.enterText(usernameField, 'budrefurta@gufum.com');  await tester.pumpAndSettle(); // Обновляем интерфейс.  // Вводим тестовый пароль.  await tester.enterText(passwordField, 'test123456');  await tester.pumpAndSettle(); // Обновляем интерфейс.  // Находим кнопку входа по ключу и нажимаем на нее.  final enterButton = find.byKey(ValueKey('enterButton'));  await tester.tap(enterButton);  await tester.pumpAndSettle(); // Дожидаемся завершения анимаций.  // Проверяем, что после успешного входа отображается имя пользователя.  expect(find.text('Михаилл'), findsOneWidget);  });  } |

1. **Инструкция по эксплуатации программного обеспечения**
   1. **Установка программного обеспечения**

Для функционирования системы на стороне сервера достаточны следующие программные и технические средства:

* операционная система – Linux (Ubuntu, Debian + версии),
* Процессор - 2 ГГц;
* Оперативная память - 2 ГБ;
* Свободное место на диске – 10 ГБ;
* Дополнительные компоненты: Python 3.6 (или выше), PostgreSQL, Django REST Framework, Nginx (или Apache).

Добавление объектов БД происходит через миграции от Django REST Framework.

Для установки серверной части требуется перейти в терминале в корневую папку API и в терминале использовать команду python manage.py runserver.

Для функционирования мобильного приложения достаточны следующие минимальные программные и технические средства::

* Операционная система - Android 11 или iOS 18;
* Процессор - 2 ГГц;
* Оперативная память - 2 ГБ;
* Свободное место в хранилище – 200 МБ;
* Дополнительное - постоянное интернет-подключение.

Для установки мобильного приложения требуется выбрать собранный apk файл с приложением в проводнике и установить его.

В мобильном приложении используются следующие данные для авторизации:

* Электронная почта – [budrefurta@gufum.com](mailto:budrefurta@gufum.com);
* Пароль – test123456.
  1. **Инструкция по работе**

При запуске приложения, пользователя встречает начальное окно с общей информацией о приложении. Для авторизации требуется нажать на иконку логина в левом верхнем углу страницы, ввести данные для авторизации и выбрать под какой ролью она происходит в появившемся модальном окне, отображенном на рисунке 9.

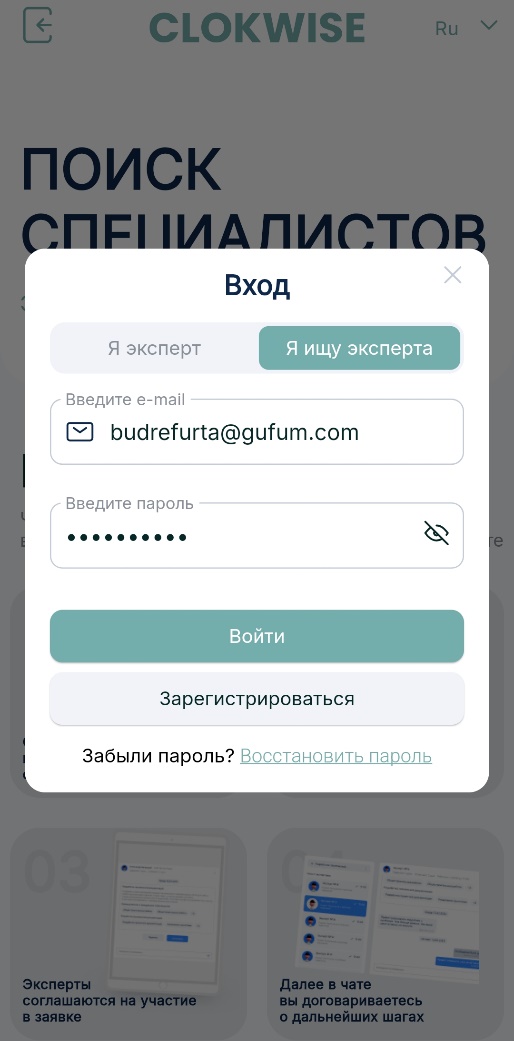


Рисунок 9 – Clokwise. Вид модального окна «Авторизация»

После авторизации пользователь перенаправляется на экран своего профиля. Пользователю в роли «Ищу эксперта» доступны экраны чатов, поиск экспертов и созданные запросы, к которым можно получить доступ через навигационную панель внизу экрана.

Для поиска эксперта требуется перейти на страницу «Поиск экспертов» при помощи навигационной панели, нажав на кнопку с иконкой лупы. После этого требуется выбрать требуемые опыт, отрасль, подотрасль, функцию и подфункцию, а также, если нужно, ввести слово для фильтрации. Затем требуется нажать на кнопку «Найти эксперта». Пример заполненной страницы для поиска экспертов изображен на рисунке 10.

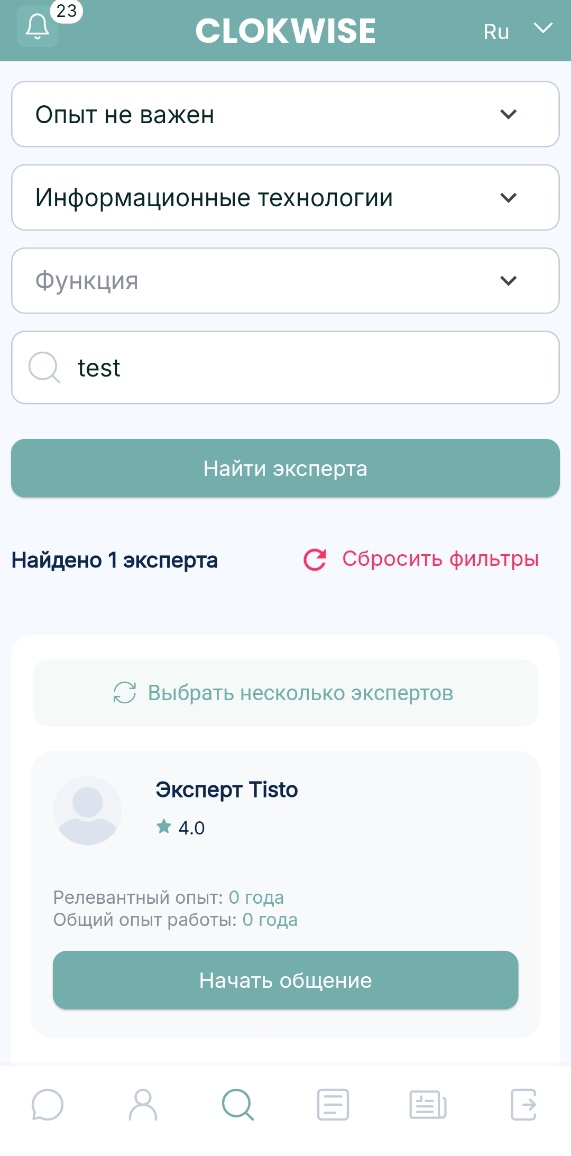


Рисунок 10 – Clokwise. Вид страницы «Поиск эксперта»

Для создания задачи требуется перейти на страницу «Запросы», где можно посмотреть свои созданные задачи. Для создания задачи требуется нажать на кнопку «Создать новый запрос», после чего заполнить данными запрос.

Пример страниц «Создание нового запроса» с заполненными данными и «Запросы» можно увидеть на рисунке 11.

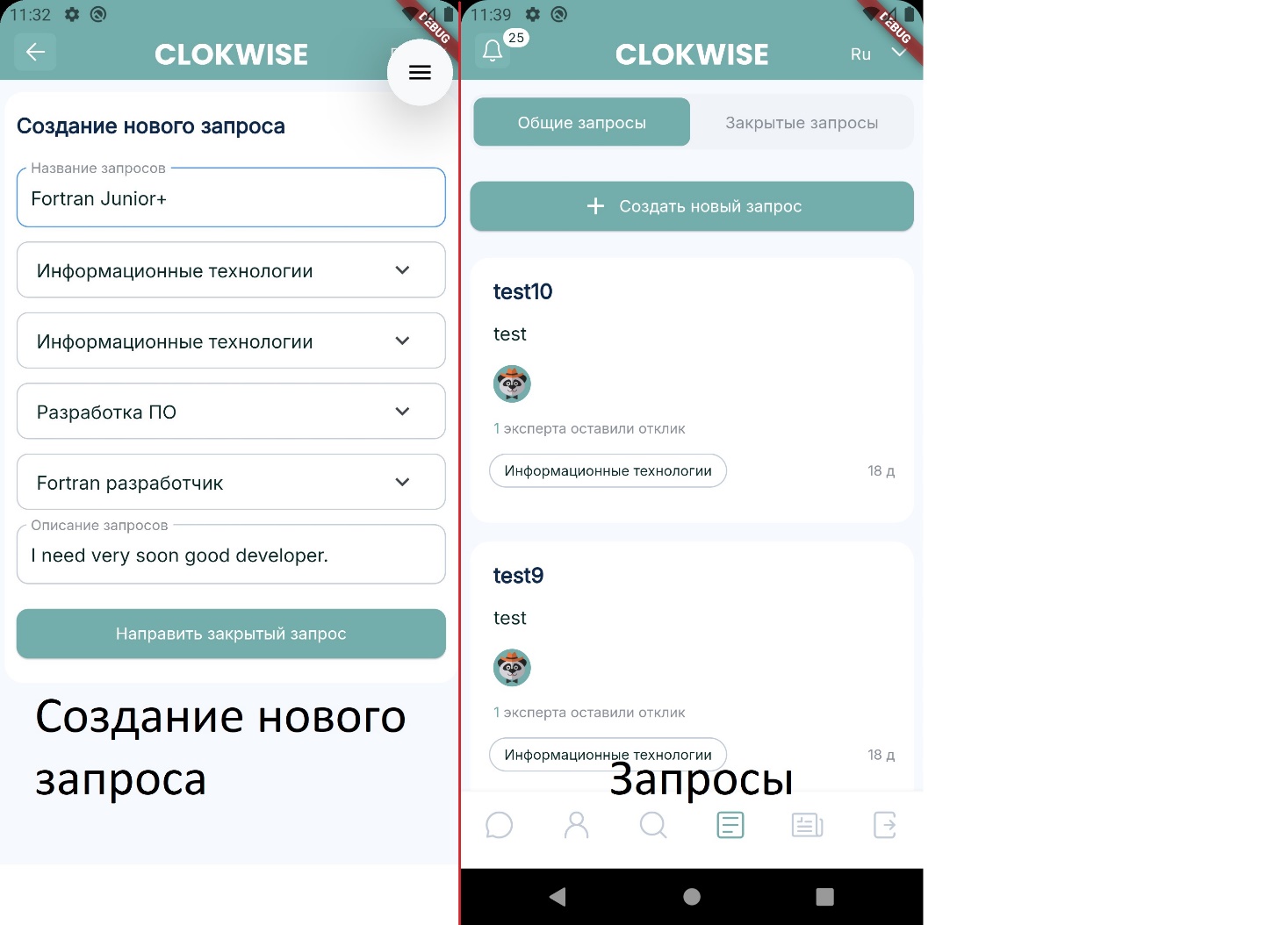


Рисунок 11 – Clokwise. Вид страниц «Создание нового запроса» и «Запрос»

Для перехода на страницу с чатами требуется нажать на навигационной панели на кнопку с иконкой облачка сообщения. На странице с чатами доступны чат с поддержкой, чаты с заказчиками. Чаты сортируются по последнему сообщению. Чат, у которого присутствует галочка и время, означает, что чат прочитан. При нажатии на чат откроется чат с выбранным пользователем.

Пример страниц со списков чатов и с чатом пользователя можно увидеть на рисунке 12.

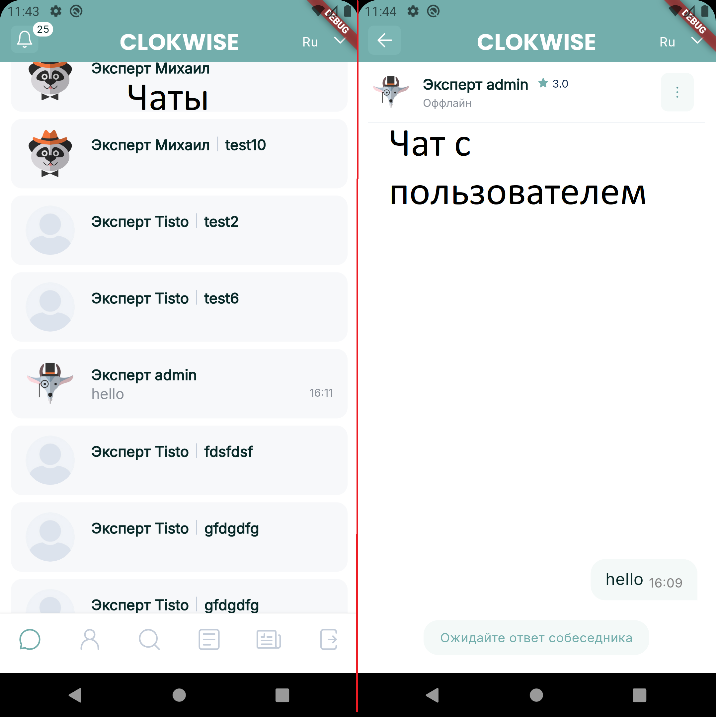


Рисунок 12 – Clokwise. Вид страниц «Чаты» и «Чат с пользователем»

Пользователю с ролью «эксперт» доступны только экраны чатов и своего профиля. Для смены роли требуется зайти на страницу профиля и в верхней части страницы выбрать роль. Переключатель ролей можно увидеть на странице «Профиль», изображенной на рисунке 13.

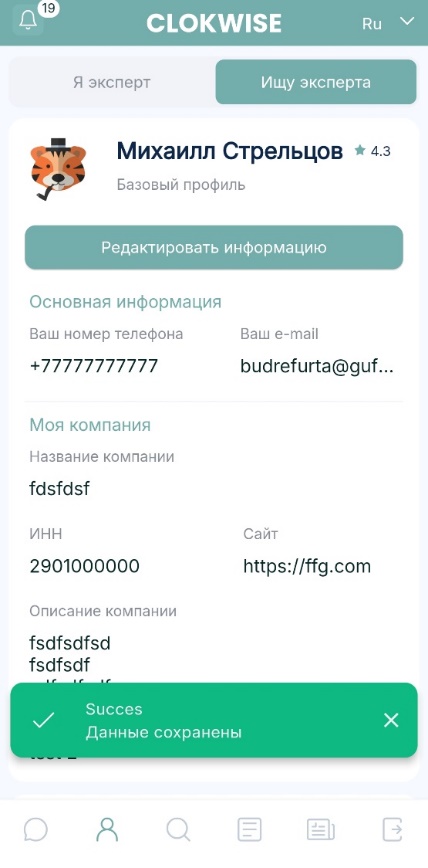


Рисунок 13 – Clokwise. Вид страницы «Профиль»**ЗАКЛЮЧЕНИЕ**

В ходе курсового проектирования достигнута поставленная цель: спроектировано мобильное приложение «Подбор персонала для выполнения задач». Данная обучающая программа поможет в освоении новых знаний или закрепления уже имеющегося в области сварочных работ. Кроме того, решены все поставленные задачи:

* выполнен сбор требований целевой аудитории,
* проанализированы информационные источники по предметной области,
* спроектирована архитектура приложения,
* спроектирована диаграмма вариантов использования,
* выбран состав программных и технических средств, используемых для реализации информационно-поисковой системы,
* спроектирована диаграмма вариантов использования приложения,
* спроектирована физическая схему БД,
* разработано API для мобильного приложения,
* реализовано разграничение прав доступа пользователей,
* реализована защита данных,
* разработано мобильное приложение,
* реализован экспорт данных в виде файлов и изображений,
* реализован импорт данных в виде файлов и изображений,
* реализована работа с сервером при помощи REST API,
* выполнено структурное тестирование ПО,
* выполнено функциональное тестирование ПО,
* разработать программную и эксплуатационную документацию.

В результате выполнения поставленных задач разработано мобильное приложения для подбора персонала, отвечающее современным тенденциям и требованиям заказчика.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Мартишин, С. А. Базы данных. Практическое применение СУБД SQL- и NoSQL-типа для проектирования информационных систем : учебное пособие / С. А. Мартишин, В. Л. Симонов, М. В. Храпченко. — Москва : ФОРУМ : ИНФРА-М, 2023. — 368 с. - URL: https://znanium.com/catalog/product/1912454 (дата обращения: \_\_.11.2024). – Режим доступа: по подписке. — Текст : электронный.
2. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Сидорова-Виснадул ; под ред. Л. Г. Гагариной. – Москва : ФОРУМ : ИНФРА-М, 2023. – 400 с. – URL: https://znanium.com/catalog/product/1895679 (дата обращения: \_\_.11.2024). – Режим доступа: по подписке. – Текст : электронный.
3. Дадян, Э. Г. Данные: хранение и обработка : учебник. – Москва : ИНФРА-М, 2020. – 205 с. – URL: https://znanium.com/catalog/product/1045133 (дата обращения: \_\_.11.2024). – Режим доступа: по подписке. – Текст : электронный.
4. Гивакс, Д. Д. Паттерны проектирования API. – Санкт-Петербург : Питер, 2023. – 512 с. – URL: https://ibooks.ru/bookshelf/390212/reading (дата обращения: \_\_.11.2024). – Режим доступа: для зарегистрир. пользователей. – Текст: электронный.
5. Бек, К. Экстремальное программирование: разработка через тестирование. – Санкт-Петербург : Питер, 2021. – 224 с. – URL: https://ibooks.ru/bookshelf/376974/reading (дата обращения: \_\_.11.2024). – Режим доступа: для зарегистрир. пользователей. – Текст: электронный.