Учреждения образования «БЕЛОРУССКИЙ

`

ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 1-98 01 03 Программное обеспечение информационной безопасности мобильных систем

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к дипломному проекту:**

Мобильное приложения для назначения задач TaskMate

Дипломник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Тихон А. А.

Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бурмакова А. В., асс.

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Блинова Е. А., и.о. зав кафедры,

доцент

Консультант \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Соболевский А. С., ст. преп.

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Николайчук А. Н., преподаватель-стажер

Дипломный проект защищен с оценкой

Председатель ГЭК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дюбков В. К., к.т.н., доцент

Минск 2023

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

`

Учреждение образования «БЕЛОРУССКИЙ

ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 1-98 01 03 Программное обеспечение информационной безопасности мобильных систем

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_ Е.А. Блинова

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.

**ЗАДАНИЕ**

**на дипломный проект**

Тихон Алексей Александрович

(фамилия, имя, отчество)

1. Тема проекта:Мобильное приложение для назначения задач TaskMate

2. Тема утверждена приказом по университету от 21.02.2024 г. № 90-С

3. Срок сдачи студентом законченного проекта: 26.06.2024 г.

4. Исходные данные к проекту (требования к системе):

3.1 Назначение: *мобильное приложение предназначено для решения проблем при ведении проектов. Возможность хранить информацию о текущих проектах, поставленных перед пользователями задачах и сроках их выполнения позволяет повысить производительность, эффективность работы и снизить количество ошибок.*

4.2 Основные функциональные возможности: *создание и управление организацией, создание проектов, добавление участников в проект, создание задач и подзадач, назначение ответственных задач, изменение статуса задачи и проекта, возможность просмотра задач и проектов в календаре, отображение информации о проекте в формате инфографики.*

4.3 Пользовательские роли: *Создатель организации (удаление участников из организации, изменение информации об организации). Создатель проекта (добавление участников в проект, удаление участников из проекта, удаление проекта, завершение проекта, фиксация статуса задачи). Участник проекта (добавление задач, добавление подзадач, предложение изменений статуса задачи).*

4.4 Целевая аудитория: *компании и частные лица, которые желают повысить эффективность работы и снизить количество ошибок.*

4.5 Программная платформа: *NodeJS 20.11.1, Express 3.18.2, Microsoft SQL Server 16.0; клиентская часть: Flutter 2.19.3, Dart 2.2.1.*

5. Содержание расчетно-пояснительной записки:

1. *Реферат*

`

1. *Содержание*

[Привлеките внимание читателя с помощью яркой цитаты из документа или используйте это место, чтобы выделить ключевой момент. Чтобы поместить это текстовое поле в любой части страницы, просто перетащите его.]

1. *Введение*
2. *Раздел 1: обзора аналогов и постановка задачи*
3. *Раздел 2: проектирование программного продукта*
4. *Раздел 3: программная реализация продукта*
5. *Раздел 4: анализ информационной безопасности приложения*
6. *Раздел 5: тестирование приложения*
7. *Раздел 6: руководство пользователя*
8. *Раздел 7: экономический раздел*
9. *Заключение*
10. *Список использованных источников*
11. *Приложения и графическая часть*

6. Перечень графического материала (с точным указанием обязательных чертежей):

1. *Диаграмма вариантов использования мобильного приложения*
2. *Диаграмма вариантов использования веб-приложения*
3. *Логическая схема глобальной базы данных*
4. *Логическая схема локальной базы данных*
5. *Архитектура приложения*
6. *Блок-схема выгрузки задач и подзадач*
7. *Результаты расчетов экономического раздела*

7. Консультанты по проекту с указанием относящихся к ним разделов проекта

|  |  |
| --- | --- |
| Раздел | Консультант |
| *Экономический раздел* | *А. С. Соболевский* |

8. Дата выдачи задания:

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Бурмакова

(подпись)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Тихон

(подпись)

9. Календарный план

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование этапов дипломного проекта | Срок выполнения этапов проекта | Примечание |
| *1* | *Постановка задачи и анализ аналогичных решений* | 20.02.2024 |  |
| *2* | *Проектирование приложения* | 02.03.2024 |  |
| *3* | *Реализация приложения* | 26.03.2024 |  |
| *4* | *Тестирование приложения* | 02.06.2024 |  |
| *5* | *Анализ информационной безопасности* | 10.06.2024 |  |
| *6* | *Выполнение расчетов экономического раздела* | 17.06.2024 |  |
| *7* | *Оформление пояснительной записки* | 19.06.2024 |  |
| *8* | *Рецензирование дипломного проекта* | 21.06.2024 |  |

Дипломник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Руководитель проекта\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (подпись)

`

**Реферат**

Листов

*ФИО*

*Тихон А.А.*

*Бурмакова А.В.*

*Блинова Е.А.*

*Подп.*

*Дата*

Разраб.

Пров.

Утв.

Лит.

Лист

*БГТУ 00.00.ПЗ*

*74201055, 2024*

*Реферат*

Н. контр.

*Николайчук А.Н.*

1

1

Пояснительная записка дипломного проекта содержит 71 страницу, 34 иллюстрации, 5 таблиц, 16 использованных источников, 8 приложений.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ для НАЗНАЧЕНИЯ ЗАДАЧ, nodejs, EXPRESS, javascript, FLUTTER, DART, MICROSOFT SQL SERVER

Объект разработки – мобильное приложение для назначения задач.

Цель дипломного проекта заключается в создании мобильного приложения, которое позволит пользователю создавать команды для выполнения проектов, отслеживать прогресс и распределять задачи среди участников команды путем создания задач и подзадач.

Пояснительная записка состоит из введения, семи разделов, заключения, списка ипользованных источников и приложений.

Во введении представлена общая информация о рарабатываемом приложении, описана цель дипломного проекта и задачи для ее достижения.

В первом разделе представлены результаты аналитического обзора аналогичных решений, рассмотрены их достоинства и недостатки, описана предментная область веб-прилодения.

Во втором разделе произведено обоснование используемых технологий и подробно описан процесс проектирования программного средства.

В третьем разделе представлена программная реализация приложений. Рассмотрены процессы реализации серверной и клиентской частей приложения, а также процесс создания архитектуры базы данных.

В четвертом разделе рассмотрены средства для обеспечения информационной безопасности веб-приложения, а также описаны возможные уязвимости.

В пятом разделе рассмотрен процесс тестирования веб-приложения. Рассмотрены сценарии, в которых могут возникнуть ошибки.

В шестом разделе представлено руководство по использованию приложения. Рассмотрены основные сценарии использования приложения.

В седьмом разделе представлен расчет экономических параметров и себестоимости программного средства, разработанного в рамках дипломного проекта.

В заключении представлены итоги дипломного проекта и задачи, которые были решены в ходе разработки программного средства.

В результате разработки создано веб-приложение для учета и планирования бюджета, с возможностью вести учет в различных валютах.

**Abstract**

[Привлеките внимание читателя с помощью яркой цитаты из документа или используйте это место, чтобы выделить ключевой момент. Чтобы поместить это текстовое поле в любой части страницы, просто перетащите его.]

Листов

*ФИО*

*Тихон А.А.*

*Бурмакова А.В.*

*Блинова Е.А.*

*Подп.*

*Дата*

Разраб.

Пров.

Утв.

Лит.

Лист

*БГТУ 00.00.ПЗ*

*74201055, 2024*

*Abstract*

Н. контр.

*Николайчук А.Н.*

1

1

The explanatory note of the diploma project contains 71 pages, 34 illustrations, 5 tables, 16 references, 8 applications.

WEB APPLICATION FOR ACCOUNTING AND BUDGET, NODEJS, POSTGRESQL, DOCKER, JAVASCRIPT, TYPESCRIPT, REACTJS, TYPEORM, GRAPHQL, APOLLO CLIENT

The object of development is a web application for accounting and budget planning.

The goal of the graduation project is to create a web application that will allow the user to track their budget by creating transactions in order to further plan income and expenses.

The explanatory note consists of an introduction, seven sections, a conclusion, a list of references and applications.

The introduction provides general information about the developed web application, describes the purpose of the graduation project and the tasks to achieve it.

The first section presents the results of an analytical review of similar solutions, discusses their advantages and disadvantages, and describes the subject area of the web application.

The second section substantiates the technologies used and describes in detail the software design process.

The third section presents the software implementation of applications. The processes of implementing the server and client parts of a web application, as well as the process of creating a database architecture, are considered.

The fourth section discusses the means for ensuring the information security of a web application, and also describes possible vulnerabilities.

The fifth section discusses the process of testing a web application. Scenarios in which errors may occur are considered.

The sixth section provides guidance on how to use the application. The main scenarios for using the application are considered.

The seventh section presents the calculation of economic parameters and the cost of software developed as part of the graduation project.

In conclusion, the results of the graduation project and the tasks that were solved during the development of the software are presented.

As a result of the development, a web application for accounting and budget planning was created, with the ability to keep records in various currencies.

[Привлеките внимание читателя с помощью яркой цитаты из документа или используйте это место, чтобы выделить ключевой момент. Чтобы поместить это текстовое поле в любой части страницы, просто перетащите его.]

Листов

*ФИО*

*Тихон А.А.*

*Бурмакова А.В.*

*Блинова Е.А.*

*Подп.*

*Дата*

Разраб.

Пров.

Утв.

Лит.

Лист

*БГТУ 00.00.ПЗ*

*74201055, 2024*

*Содержание*

Н. контр.

*Николайчук А.Н.*

1

2

**Оглавление**

[Введение 7](#_Toc167717039)

[1 Обзор аналогов и постановка задачи 8](#_Toc167717040)

[1.1 Обзор приложения «Trello» 8](#_Toc167717041)

[1.2 Обзор приложения «Todoist» 9](#_Toc167717042)

[1.3 Обзор приложения «TickTick» 10](#_Toc167717043)

[1.4 Постановка задач 11](#_Toc167717044)

[2 Проектирование программного продукта 12](#_Toc167717045)

[2.1 Архитектура приложения 12](#_Toc167717046)

[2.2 Проектирование глобальной базы данных 12](#_Toc167717047)

[2.3 Проектирование локальной базы данных 13](#_Toc167717048)

[2.4 Проектирование сервисов доступа к данным 14](#_Toc167717049)

[2.5 Проектирование мобильного приложения 15](#_Toc167717050)

[2.6 Проектирование веб приложения 18](#_Toc167717051)

[2.7 Выводы по главе «Проектирование программного продукта» 20](#_Toc167717052)

[3 Программная реализация продукта 21](#_Toc167717053)

[3.1 Технические средства разработки 21](#_Toc167717054)

[3.2 Разработка глобальной базы данных 22](#_Toc167717055)

[3.3 Разработка локальной базы данных 22](#_Toc167717056)

[3.4 Разработка сервера 22](#_Toc167717057)

[3.5 Разработка мобильного приложения 23](#_Toc167717058)

[3.6 Выводы по главе «Програмная реализация продукта» 28](#_Toc167717059)

[4 Обеспечение безопасности информационной системы 29](#_Toc167717060)

[4.1 Вывод по главе «Обеспечение безопасности информационной системы» 29](#_Toc167717061)

[5 Тестирование приложения 30](#_Toc167717062)

[5.1 Тестирование валидации данных 30](#_Toc167717063)

[5.2 Выводы по разделу 32](#_Toc167717064)

[6 Руководство пользователя 33](#_Toc167717065)

[6.1 Руководство пользователя мобильного приложения 33](#_Toc167717066)

[6.2 Руководство пользователя веб приложения 40](#_Toc167717067)

[7 Технико-экономическое обоснование проекта 44](#_Toc167717068)

[7.1 Общая характеристика разрабатываемого программного средства 44](#_Toc167717069)

[7.2 Исходные данные для проведения расчётов 44](#_Toc167717070)

[7.3 Объем программного средства 44](#_Toc167717071)

[7.4 Основная заработная плата 45](#_Toc167717072)

[7.5 Дополнительная заработная плата 46](#_Toc167717073)

[7.6 Отчисления в Фонд социальной защиты населения 46](#_Toc167717074)

[7.7 Расходы на материалы 47](#_Toc167717075)

[7.8 Расходы на специальное оборудование и платные услуги 47](#_Toc167717076)

[7.9 Прочие прямые затраты 47](#_Toc167717077)

[7.10 Накладные расходы 48](#_Toc167717078)

[7.11 Сумма расходов на разработку программного средства 48](#_Toc167717079)

[7.12 Расходы на сопровождение и адаптацию 48](#_Toc167717080)

[7.13 Полная себестоимость 49](#_Toc167717081)

[7.14 Определение цены, оценка эффективности 49](#_Toc167717082)

[7.15 Вывод по разделу 51](#_Toc167717083)

[Заключение 53](#_Toc167717084)

[Список использованных источников 54](#_Toc167717085)

[ПРИЛОЖЕНИЕ А 55](#_Toc167717086)

[ПРИЛОЖЕНИЕ Б 56](#_Toc167717087)

[ПРИЛОЖЕНИЕ В 57](#_Toc167717088)

[ПРИЛОЖЕНИЕ Г 58](#_Toc167717089)

Введение

`

Листов

*ФИО*

*Тихон А.А.*

*Бурмакова А.В.*

*Блинова Е.А.*

*Подп.*

*Дата*

Разраб.

Пров.

Утв.

Лит.

Лист

*БГТУ 00.00.ПЗ*

*74201055, 2024*

*Введение*

Н. контр.

*Николайчук А.Н.*

1

1

Для успешного функционирования бизнеса необходимо четко следить за рабочим процессом. Рабочий процесс можно описать с помощью трех вопросов: что, как и когда необходимо сделать. В наше время задачи становятся объемнее и сложнее, удержать все в голове становится попросту невозможно. Для решения данной проблемы существуют различные системы управления рабочими процессами. Такие системы могут применяться абсолютно в любой сфере: финансы, логистика, техническое обслуживание.

Использование систем управления рабочими процессами позволяет повысить производительность и эффективность работы, а также снизить количество ошибок. Также одним из важных преимуществ использования систем управления рабочим процессом является обеспечение прозрачности выполняемых работ. Прозрачность позволяет более точно планировать и распределять ресурсы, обеспечивать согласованность внутри коллектива.

Для успешного функционирования системы управления рабочими процессами необходимо обеспечить представление задач в удобном и простом для понимания формате. Все проекты можно разделить на три категории: выполненные, выполняющиеся и ожидающие выполнения в будущем. Также необходима возможность разбивать проект на подзадачи, которые будут распределяться между участниками команды.

Разработка мобильной версии системы управления рабочими процессами связана с возможностью возникновения ситуации, в которой у сотрудника не будет доступа к компьютеру. Однако мобильные устройства всегда находятся под рукой и обеспечивают простой и быстрый доступ к приложениям.

Также необходимо предусмотреть невозможность пользователя выйти в интернет. В такой ситуации пользователь должен обладать возможностью просмотреть текущие проекты и задачи, которые он должен выполнить. То есть необходимо обеспечить автономность приложения в условиях отсутствия доступа в сеть интернет.

Подводя итог и обобщив всю вышеизложенную информацию, определим цель выполнения дипломного проекта как разработку системы управления рабочим процессом, которая позволяет создавать проекты, разбивать их на подзадачи и распределять между участниками проекта. При этом необходимо разработать мобильную версию и обеспечить её автономность с помощью синхронизации внутренней базы данных с глобальной базой данных

1 Обзор аналогов и постановка задачи

`

Листов

*ФИО*

*Тихон А.А.*

*Бурмакова А.В.*

*Блинова Е.А.*

*Подп.*

*Дата*

Разраб.

Пров.

Утв.

Лит.

Лист

*БГТУ 01.00.ПЗ*

*74201055, 2024*

*Обзор аналогов и постановка*

*задачи*

Н. контр.

*Николайчук А.Н.*

1

4

Обзор аналогов – важный этап при разработке программного обеспечения, который позволяет определиться с выбором подхода и инструментов для разработки приложения. Также обзор аналогов помогает определить базовый функционал приложения, выявить все недостатки и преимущества конкурентных приложений, определить функционал, необходимый для разрабатываемого приложения. Выявления недостатков позволяет избежать их при разработке и проектировании собственного приложения.

## Обзор приложения «Trello»

В качестве аналога было выбрано мобильное приложение Trello (рис 1.1).



Рисун ок 1.1 – Мобильное приложение Trello

Trello – облачная программа для управления проектами небольших групп, разработанная Fog Creek Software.

Trello позволяет создавать карточки, которые представляют из себя задачу, изменять статус задачу: задача может находиться в одном из трех состояний: «Сделать», «В работе», «Готово». Карточки, в зависимости от состояния, распределяются в три колонки.

Trello позволяет создавать множество проектов, называемых «досками». Каждая из досок также имеет свое состояние.

Также Trello поддерживает функцию «Чек -листы». По сути, данная функция позволяет разбивать задачи на подзадачи для более удобного оформления задачи.

Trello – одна из самых популярных на данный момент систем управления проектами.

## 1.2 Обзор приложения «Todoist»

В качестве аналога было выбрано мобильное приложение Trello (рис 1.1).

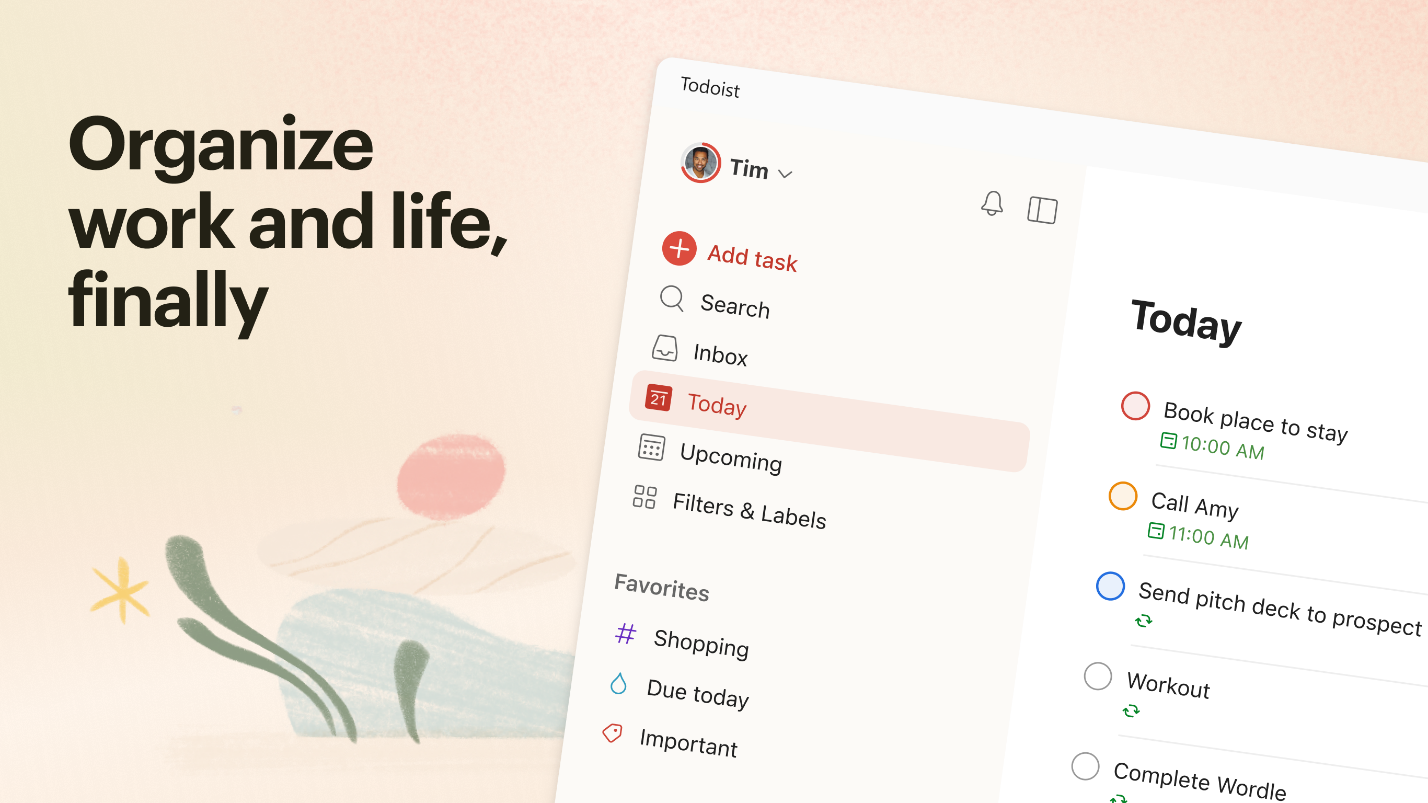


Рисунок 1.2 – приложение Todoist

[**Todoist** –это функциональный таск-менеджер, который позволяет удобно работать над задачами и быстро отслеживать прогресс](https://udalenking.ru/todoist/).

Todoist обладает следующими преимуществами:

– простой и интуитивный интерфейс;

– кроссплатформенность;

– возможность быстрого добавления задач;

– поддержка общих проектов для совместной работы.

Также приложение обладает функцией синхронизации между устройствами, что позволяет получить доступ к приложению в любом месте.

Приложение распространяется на бесплатной основе, но с ограничениями: напоминания, резервное копирование и роли пользователей, используемые для более детального управления командой доступны только в платной версии.

## 1.3 Обзор приложения «TickTick»

Последним приложением-аналогом для обзора является приложение TickTick.

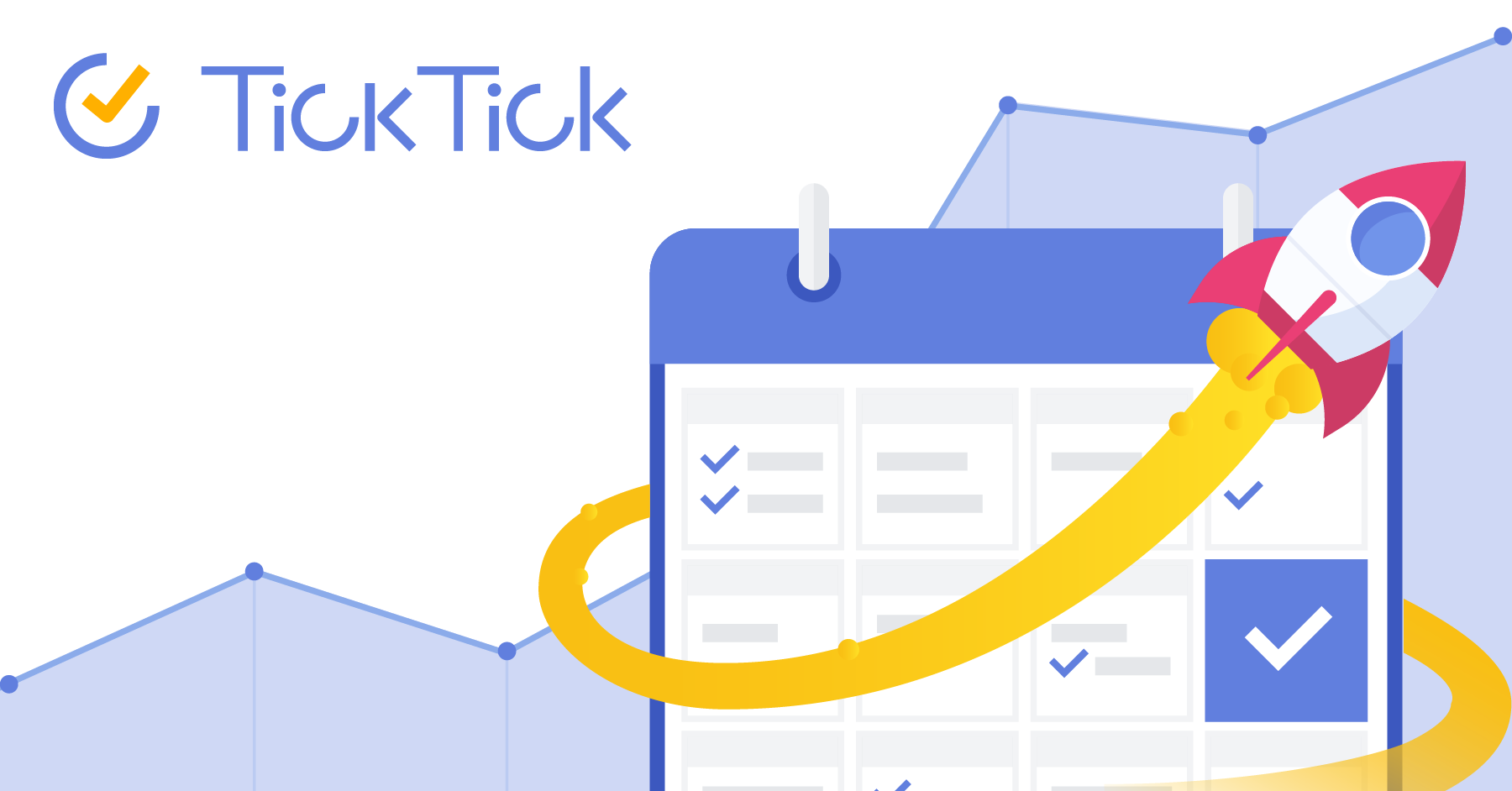


Рисунок 1.3 – приложение TickTick

Приложение в первую очередь полезно для индивидуальных пользователей и для отслеживания повседневных задач, а для команд, работающих совместно это приложение может быть недостаточно функциональным.

Особенности приложения TickTick:

– интуитивный интерфейс;

– интеллектуальные напоминания;

– приоритеты задач;

– кроссплатформенность;

– pomodoro-таймер.

Приложение распространяется на бесплатной основе, тем не менее бесплатная версия имеет некоторые ограничения: максимальное количество задач – 99, максимальное количество подзадач – 19.

Также приложение предоставляет функцию привычек, которая позволяет отслеживать и формировать здоровые привычки у пользователя.

## 1.4 Постановка задач

Делая выводы из вышеперечисленного, можно сделать заключение и сформировать задачу: разработать приложение, обладающее следующими основными функциями:

– возможность создавать организации для объединения пользователей;

– возможность создавать команды для выполнения проекта;

– возможность вступать в организацию;

– возможность удалять участников организации;

– возможность создавать проекты;

– возможность создавать команды для выполнения проекта;

– возможность добавлять участников в проект;

– возможность создания задач;

– возможность создания подзадач;

– возможность изменения статуса проекта;

– возможность удаления задач, задач;

– возможность удаления участников проекта;

– возможность завершения проекта.

При реализации всех вышеперечисленных требований приложение сможет обеспечить пользователей необходимым уровнем функционала.

# 

2 Проектирование программного продукта

`

Листов

*ФИО*

*Тихон А.А.*

*Бурмакова А.В.*

*Блинова Е.А.*

*Подп.*

*Дата*

Разраб.

Пров.

Утв.

Лит.

Лист

*БГТУ 02.00.ПЗ*

*74201055, 2024*

*Проектирование программного*

*продукта*

Н. контр.

*Николайчук А.Н.*

1

9

## 2.1 Архитектура приложения

Архитектура программного средства – это его строение как оно видно (или должно быть видно) извне его, т.е. представление программного средства как системы, состоящей из некоторой совокупности взаимодействующих подсистем. В качестве таких подсистем выступают обычно отдельные программы.

Разработка архитектуры является первым этапом, на котором реализуется принцип выделения относительно независимых компонентов. Архитектура программного средства схематично представлена на рисунке 2.1.

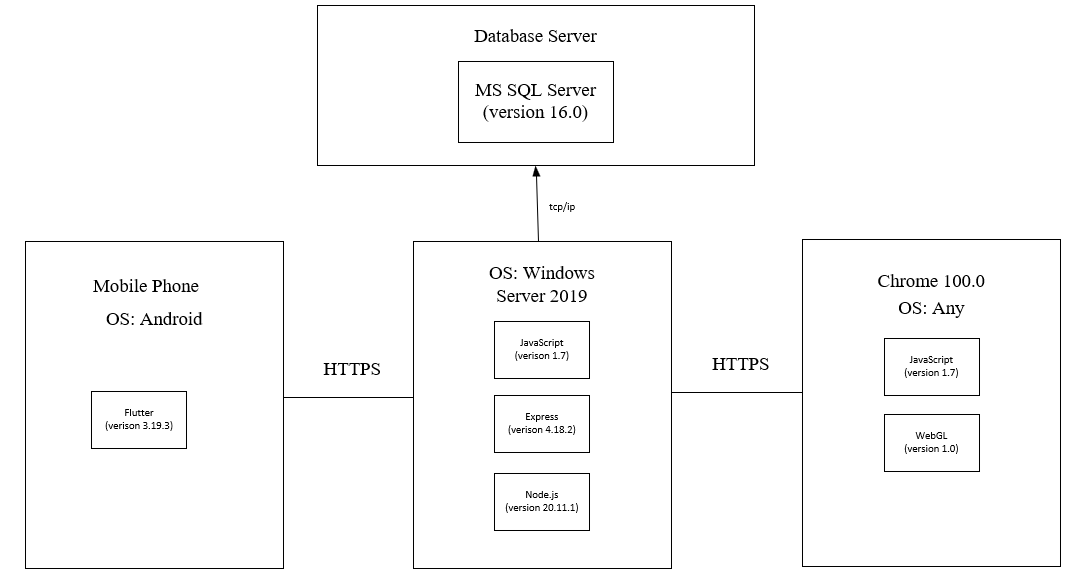


Рисунок 2.1 – Архитектура программного средства

Если рассматривать рисунок 1.1, то можно заметить, что приложение состоит из трех основных компонентов. Серверная часть приложения реализована на Node.js, платформе для разработки серверных приложений на JavaScript. Node.js предоставляет эффективное и масштабируемое окружение для обработки веб-запросов и выполнения бизнес-логики.

## 2.2 Проектирование глобальной базы данных

Глобальная база данных состоит из 7 таблиц: «AppUser», «Organisation»,

«OrganisationMember», «ProjectMember», «Project», «SubTask», «SubTaskExecutor».

Схема глобальной базы данных представлена на рисунке 2.1.

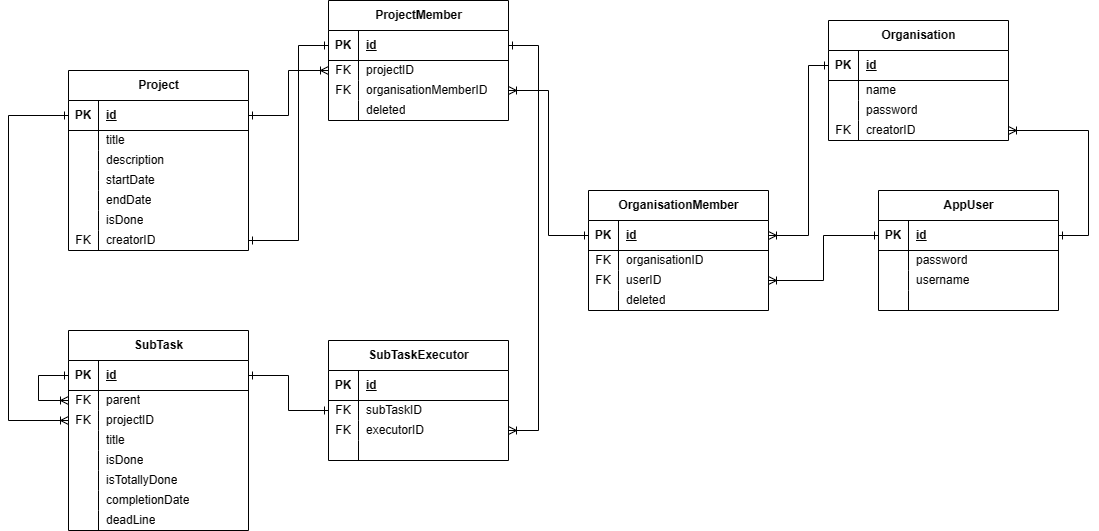


Рисунок 2.2 – Схема глобальной базы данных

Таблица «AppUser» предназначена для хранения имен пользователей, а также их паролей.

Таблица «Organisation» хранит информацию об организациях, в которые могут вступать пользователи.

Таблица «OrganisationMember» используется для хранения информации об участниках организации.

Таблица «Project» хранит данные о проектах: название, описание, дата начала, дата конца, состояние проекта, а также данные о создателе проекта.

Таблица «ProjectMember» хранит информацию об участниках проекта, а также используется для идентификации исполнителя задачи.

Таблица «SubTaskExecutor» используется для хранения данных об исполнителе задачи.

Таблица «SubTask» хранит в себе данные о задачах, которые должны быть выполнены. В данной таблице столбец parent указывает на родительскую задачу.

Таблица «SubTaskExecutor» используется для хранения данных об исполнителе задачи.

## 2.3 Проектирование локальной базы данных

Локальная база данных используется для обеспечения автономности. Пользователь, даже не имя доступа в интернет может просмотреть текущие проекты и его задачи, создать новый проект, изменить состояние текущего проекта: добавить новую задачу или подзадачу, внести изменение в состояние текущих задач. Локальная база данных имеет аналогичные глобальной структуру и связи, однако с целью реализации обратной синхронизации были добавлены некоторые столбцы в таблицы «AppUser», «Project», «ProjectMember», «SubTask», «SubTaskExecutor». Данные столбцы можно увидеть в схеме локальной базы данных, которая продемонсторирована на рисунке 2.2.

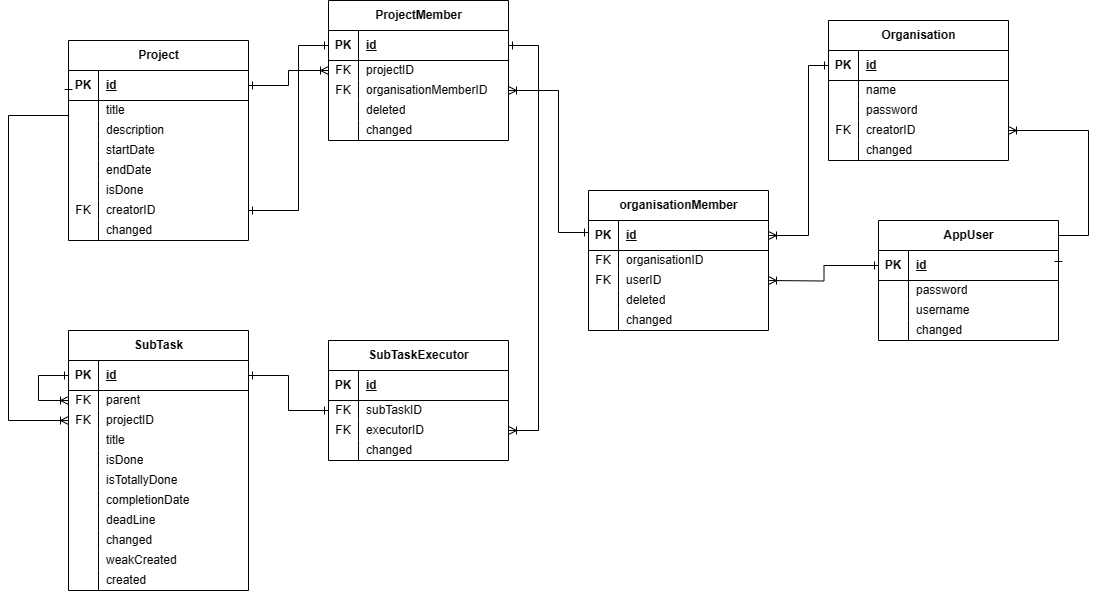


Рисунок 2.3 – Схема локальной базы данных

Столбцы changed в таблицах «AppUser», «Project», «ProjectMember», «SubTaskExecutor» и «SubTask» используются для идентификации записей, которые были внесены в локальную базу данных, однако на данный момент они не существуют в глобальной базе данных.

## 2.4 Проектирование сервисов доступа к данным

Для синхронизации двух и более клиентов необходимо серверное приложение. Серверная часть приложения должна предоставлять WEB API для доступа к каждой сущности в базе данных. Передаваемые или получаемые данные от сервера должны передаваться в формате JSON.

Серверное приложение должно быть построено на основе архитектурного стиля REST и будет использоваться для взаимодействия компонентов распределённого приложения в сети.

Сервер содержит в себе реализованные сервисы для взаимодействия с каждой сущностью базы данных.

При обращении к серверу по заданному маршруту и определенному HTTP методу сервер будет определять с какой сущностью базы данных необходимо взаимодействовать и какие действия над данными необходимо будет произвести.

При помощи HTTP метода GET сервер будет возвращать данные клиенту, ссылаясь на аргументы в запросе, как на условие для выборки данных. Метод POST в свою очередь позволяет передать данные на сервер для последующей обработки. Метод PUT позволяет обновить соответствующие данные, переданные с клиента. Метод DELETE удаляет из базы данных заданную запись.

## 2.5 Проектирование мобильного приложения

Приложение должно обеспечивать пользователю осуществлять планирование и выполнение проектов.

На данном этапе был разработан макет приложения. Макет экранов входа, регистрации и экрана со списком проектов представлены на рисунке 2.3.

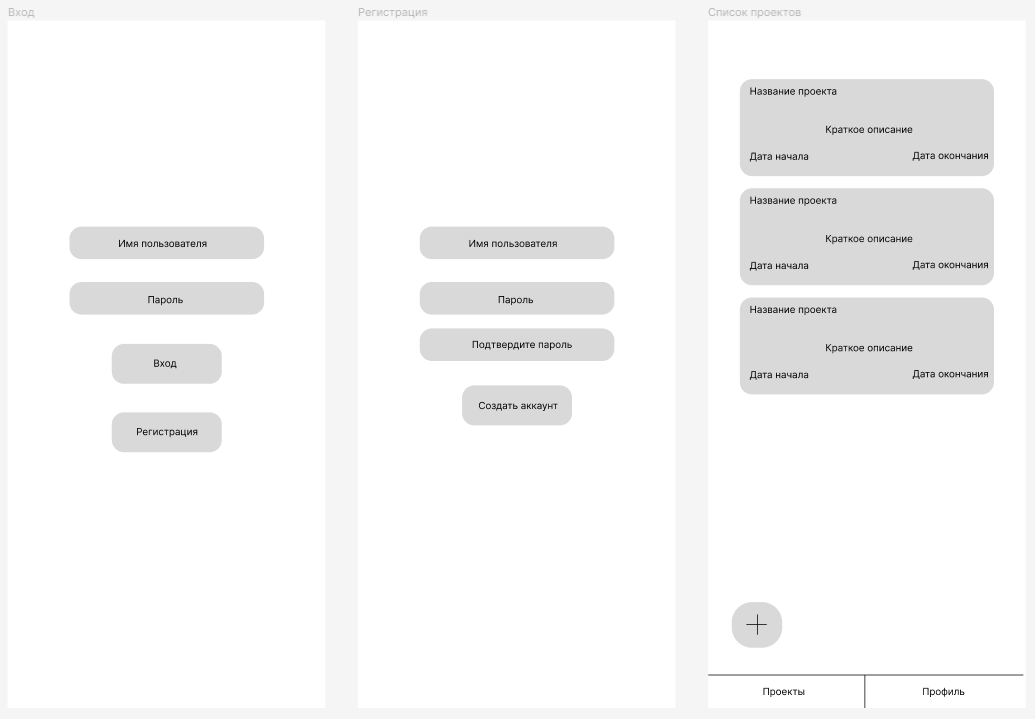


Рисунок 2.4 – Экраны вход регистрация и список проектов

На экране входа пользователь вводит данные и нажимает на кнопку входа, или нажимает на кнопку регистрации для того, чтобы перейти на экран создания аккаунта, на котором необходимо ввести имя пользователя и пароль для входа. После входа пользователь переходит на основной экран приложения. На экране проекты пользователь видит все свои текущие проекты и также может создать новый проект, нажав на соответствующую кнопку. Для того чтобы открыть страницу проекта необходимо нажать на соответствующий проект.

Макет экранов Подробности о проекте Создание нового проекта и создание подзадача представлены на рисунке 2.6.

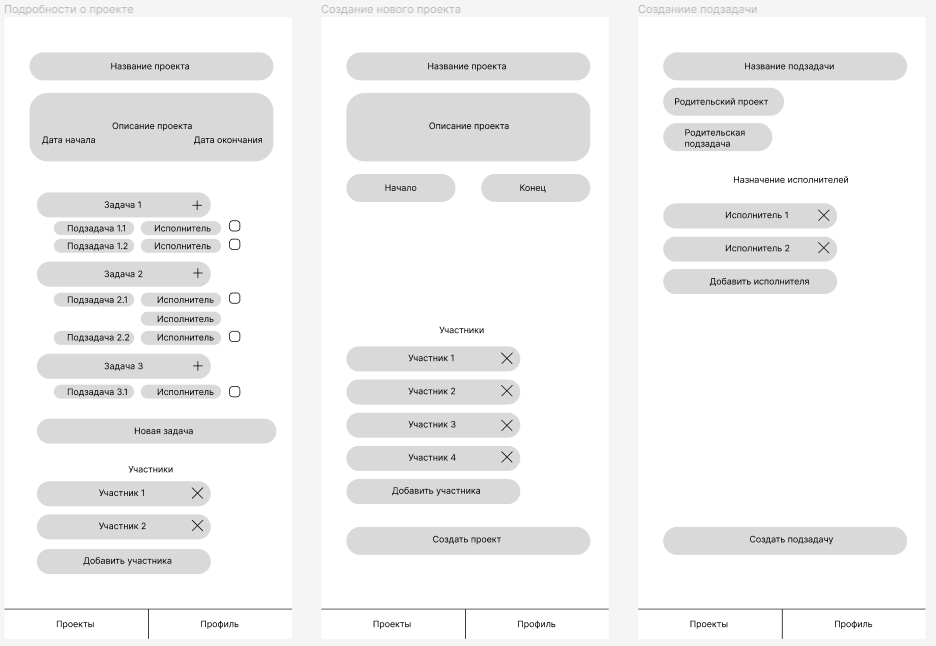


Рисунок 2.5 – Экраны вход регистрация и список проектов

Макет экранов создание новой организации добавление участника и управление организацией представлены на рисунке 2.6.



Рисунок 2.6 – Экраны вход регистрация и список проектов

При создании организации пользователю необходимо ввести название и пароль для организации. Страница управления организацией предоставляет возможность изменить название организации или удалить участников из организации.

Макет экранов Профиль и вступление в организацию представлены на рисунке 2.7.

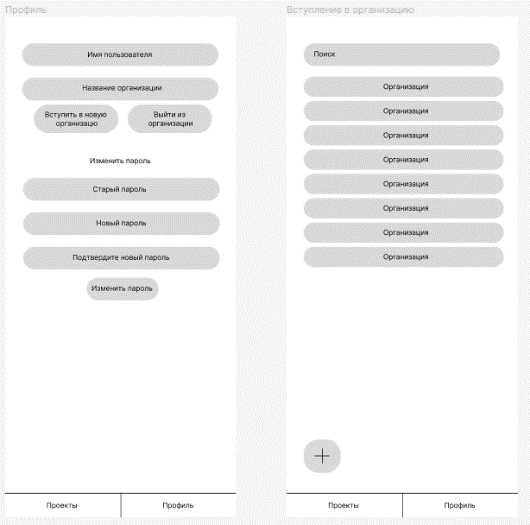


Рисунок 2.7 – Экраны профиль и вступление в организацию

Также на этапе проектирования мобильного приложения была разработана диаграмма вариантов использования. Диаграмма представлена на рисунке 2.8.

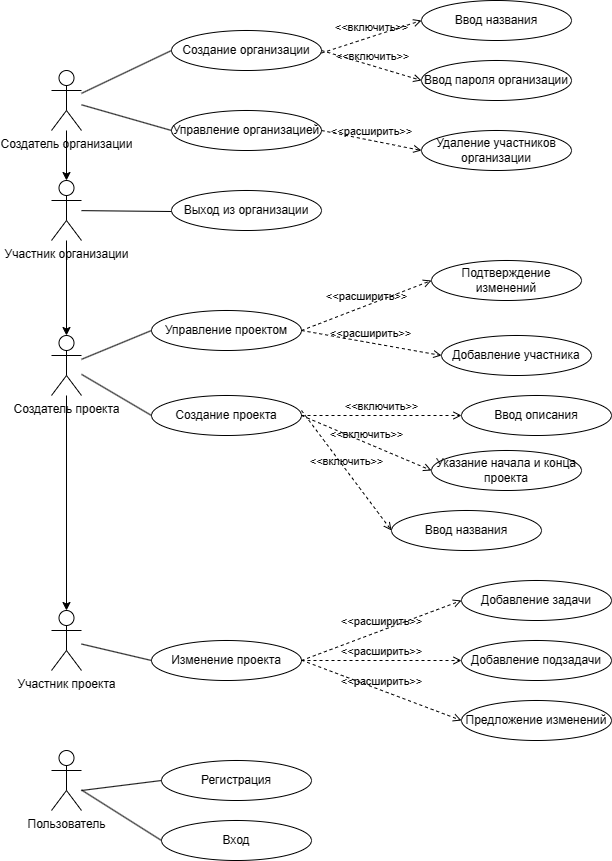


Рисунок 2.8 – Диаграмма вариантов использования

На диаграмме представлены возможности для пользователей приложения.

Согласно диаграмме, в приложении должны быть реализованы следующие функции:

– регистрация пользователя;

– авторизация пользователя;

– создание организации;

– создание проекта;

– вступление в организацию;

– добавление подзадачи;

– добавление подзадачи;

– изменение состояния подзадачи.

## 2.6 Проектирование веб приложения

На этапе проектирования веб приложения было принято решение о разработке веб приложения для реализации функций админа: удаление подзадачи, удаление задачи, удаление проекта и так далее.

Был разработан макет для веб приложения, который, который состоит из трех экранов: экран входа, список проектов, экран проекта.

Экран входа представлен на рисунке 2.9



Рисунок 2.9 – Экран входа веб приложения

Пользователю необходимо ввести имя пользователя и пароль, затем нажать на кнопку вход, после чего пользователь будет перенес на страницу со списком проектов.

Экран со список проектов представлен на рисунке 2.10.

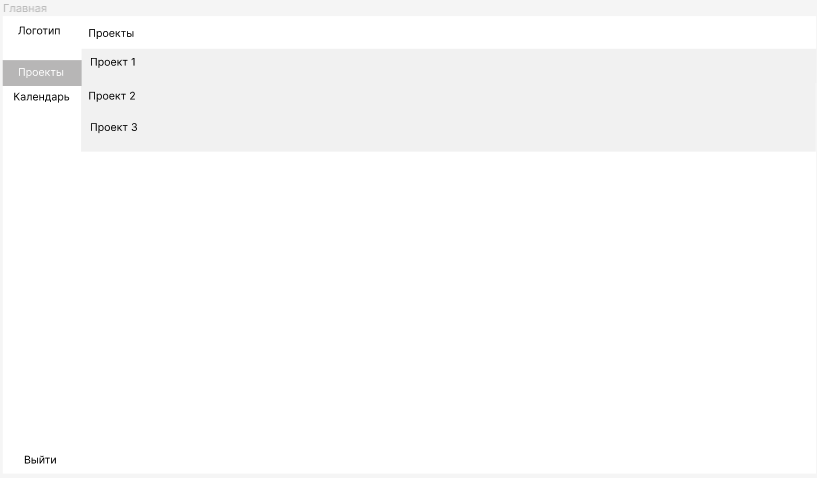


Рисунок 2.10 – Экран со списком проектов

На данном экране пользователь видит свои проекты, а также статистику, которая представляет из себя процент выполненных задач и прошедшее время.

Экран с информацией о конкретном проекте на рисунке 2.11.

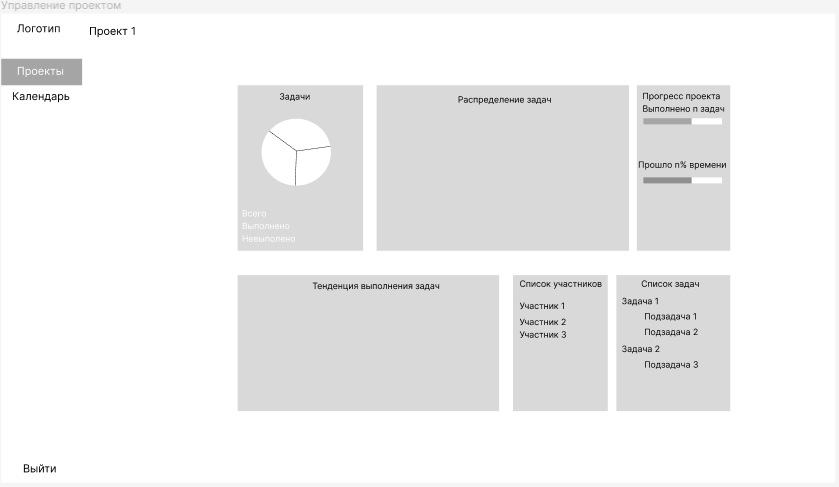


Рисунок 2.11 – Экран управления проектом

Экран управления проектом представляет из себя страницу, на которой выводятся название проекта, его описание и дата начала и завершения проекта. Помимо этого, на данной странице пользователь видит список всех участников проекта, задач и подзадач. Для управления проектом на данной страницу реализованы возможности удалять подзадачи, удалять участников из проекта и удалять задачи вместе со всеми соответствующими подзадачами.

Для веб приложения также была разработана диаграмма вариантов использования. Данная диаграмма отображает функционал веб приложения, доступный создателям проекта.

Диаграмма вариантов использования веб приложения представлена на рисунке 2.11.

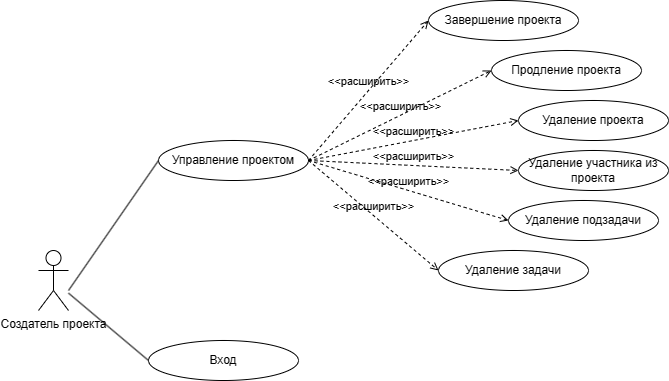


Рисунок 2.12 – Диаграмма вариантов использования веб приложения

Диаграмма, представленная на рисунке 1.11, дает понимание основных функций веб приложения.

## 2.7 Выводы по главе «Проектирование программного продукта»

В данной главе было рассмотрено проектирование каждого отдельного компонента приложения.

Были созданы макеты для мобильного и веб приложений с целью получения представления о внешнем виде экранов. Также были разработаны диаграммы вариантов использования для определения основного функционала, доступного каждой из групп пользователей.

При проектировании баз данных были разработаны таблицы, столбцы и связи между таблицами для локальной и глобальной баз данных.

В ходе проектирования сервера был выбран формат передачи данных, а также основные методы, которые будут использованы для передачи.

3 Программная реализация продукта

`

Листов

*ФИО*

*Тихон А.А.*

*Бурмакова А.В.*

*Блинова Е.А.*

*Подп.*

*Дата*

Разраб.

Пров.

Утв.

Лит.

Лист

*БГТУ 03.00.ПЗ*

*74201055, 2024*

*Программная реализация*

*продукта*

Н. контр.

*Николайчук А.Н.*

1

8

Согласное индивидуальному заданию необходимо выполнить следующие задачи:

– Выбор технических средств разработки;

– Разработка глобальной и локальный баз данных;

– Разработки сервера;

– Разработка мобильного приложения.

## 3.1 Технические средства разработки

Для разработки программного продукта были использованы следующие средства: Microsoft SQL Server, Flutter, SQLite, Node.js и Express.

Microsoft SQL Server – система управления реляционными базами данных, разработанная корпорацией Microsoft. Microsoft SQL Server была использована для разработки глобальной базы данных.

Flutter – комплект средств разработки и фреймворк для создания мобильных приложений под Android и iOS, веб приложений, а также настольных приложений с использованием языка программирования Dart. Flutter был использован при разработке мобильного и веб приложений.

Flutter был выбран за счет своей удобности и простоты использования.

SQLite – это встраиваемая кроссплатформенная БД, которая поддерживает достаточно полный набор команд SQL и доступна в исходных кодах.

Node.js – среда выполнения кода на JavaScript, которая построена на основе движка JavaScript Chrome V8, который позволяет транслировать вызовы на языке JavaScript в машинный код. Node.js прежде всего предназначен для создания серверных приложений на языке JavaScript.

Express – это минималистичный и гибкий веб-фреймворк для приложений Node.js, предоставляющий обширный набор функций для мобильных и веб-приложений.

В качестве среды разработки мобильного и веб приложений была выбрана Android Studio – интегрированная среда разработки для работы с платформой Android.

Для разработки сервера же был выбран Visual Studio Code – текстовый редактор, разработанный Microsoft.

## 3.2 Разработка глобальной базы данных

Как уже упоминалось, для разработки глобальной базы данных использовалась СУБД Microsoft SQL Server.

Скрипт для создания глобальный базы данных представлен в приложении А.

Все CRUD операции базы данных проводятся с помощью хранимых процедур, которые представляют из себя объект базы данных, который содержит в себе набор инструкций, которые выполняются как единое целое. Хранимые процедуры позволяют упростить комплексные операции и вынести их в единый объект. Также процедуры позволяют ограничить доступ к данным в таблицах и тем самым уменьшить вероятность преднамеренных или неосознанных нежелательных действий в отношении эти данных. Кроме этого, они выполняются быстрее чем обычные SQL-инструкции.

Некоторые из хранимых процедур представлены на рисунке 3.1.

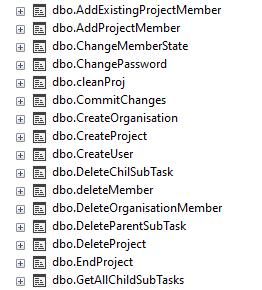


Рисунок 3.1 – Некоторые из хранимых процедур глобальной базы данных

Хранимые процедуры позволяют повысить производительность, а также обеспечить определенный уровень безопасности базы данных.

## 3.3 Разработка локальной базы данных

Локальная база данных полностью дублирует структуру глобальной базы данных. Для взаимодействия с локальной базой данных была использована библиотека sqflite – библиотека для взаимодействия с sqlite – базами данных.

Скрипт для создания локальной базы данных представлен в приложении Б.

## 3.4 Разработка сервера

Для разработки сервера был выбран программная платформа Node.js и фреймворк Express. Совокупность данных средств позволяет с легкостью разработать сервер для обработки http-запросов.

Для обработки запросов используются определенный ряд встроенных функций, таких, как например функция get.

Использование функции get представлено в листинге 3.1.

app.get("/organisation/getAllOrganisations",function(req,response){

    const request = new sql.Request();

    request.execute("GetAllOrganisations",(err,result)=>{

        response.end(JSON.stringify(result.recordset));

    })

})

Листинг 3.1 – Пример обработки GET-запроса

В качестве параметров для этой функции передаются маршрут и обработчик, который будет вызываться, если запрос к серверу соответствует данному маршруту. Таким же образом выполняется обработка POST – запросов. Пример обработки POST – запроса представлен в листинге 3.1.

app.post("/project/addSubTask",function(req,response){

    console.log("proejcts call");

    request = new sql.Request();

    request.input('userID',req.query.userID);

    request.execute('GetAllUserProjects',(err,result)=>{

        response.statusCode=200

        response.end(JSON.stringify(result.recordset));

    })

})

Листинг 3.2 – Пример обработки POST – запроса

## 3.5 Разработка мобильного приложения

При разработке мобильного приложения была использована среда разработки Android Studio. Структура проекта представлена на рисунке 2.1.

Проект состоит из пяти основных директорий: CustomModels, manyUsageTemplates, Models, Pages, Utility.

В директории Models находятся классы, который используются для взаимодействия с базами данных. Каждый файл содержит класс, чьи поля совпадают со столбцами соответствующей таблицы.

В директории CustomModels находятся классы, которые используются для выполнения сложных запросов, в которых объединяется множество таблиц, к базам данных.

В директории manyUsageTemplates находятся виджеты, многократно используемые в проекте, например текстовое поле или кнопки.

В директории Utility расположены два файла – DatabaseHandler и Utility. Файл Utility содержит в себе класс, который обеспечивает доступ свойствам, которые должны быть доступны из любого места программы, например ip-адрес и порт сервера. Файл DatabaseHandler, как следует из названия, предназначен для взаимодействия с локальной базой данных. В данном файле определены функции для синхронизации и получения данных из локальной базы данных, происходит начальная инициализация и создание базы данных.

В директории Pages находятся файлы, которые представляют из себя страницы приложения.

Каждый класс, который представляет из себя страницу имеет метод build, который возвращает интерфейс, который будет отображен на экране пользователя.

Структура проекта представлена на рисунке 3.2

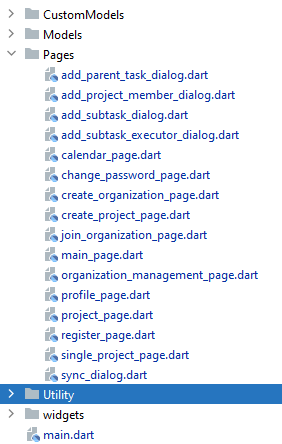


Рисунок 3.2 – Структура проекта мобильного приложения

Для взаимодействия с сервером была использована библиотека http версии 0.12.6. Данная библиотека позволяет отправлять http запросы используя фреймворк Flutter. Данная библиотека предоставляет класс http, который в свою очередь обладает методами для взаимодействия с сервером по протоколу HTTP: get, post, put, delete.

Пример использования библиотеки http представлен в листинге 3.2.

final String url = "http://${Utility.*url*}/project/addMembers";  
final response = await http.post(Uri.*parse*(url),body: jsonEncode(<String, String>{  
 'organisationID': member.organisationID.toString(),  
 'projectID': project.id.toString(),  
}));  
if (response.statusCode == 200) {  
 InitializeProject();  
} else {  
 print("Ошибка");  
}

Листинг 3.3 – Пример использования библотеки http

При отсутствии подключения данные для вывода пользователю извлекаются из локальной базы данных. Для взаимодействия с локальной базы данных необходимо получить объект класса Database – класса, который предоставляет методы для взаимодействия с базой данных. Для выполнения запросов к базе данных используется метод rawQuery, в качестве параметров которого передаются запрос и его параметры.

Пример получения данных из внутренней базы данных представлен в листинге 3.3.

Future<List <CustomProject>> getProjectsFromLocal() async{  
 final db= await database;  
 List<CustomProject> projectsBuffer = [];  
 final List<Map<String,dynamic>> maps = await db.rawQuery("select Project.id,Project.title,Project.creatorID,"  
 "Project.decription as Description,Project.endDate,Project.isDone,Project.startDate From Project "  
 "inner join ProjectMember on Project.id = ProjectMember.projectID "  
 "inner join OrganisationMember on OrganisationMember.id = ProjectMember.organisationMemberID "  
 "where OrganisationMember.userID=? and OrganisationMember.deleted=0",[Utility.*user*.id]);  
 maps.toList().forEach((element) {  
 int valueBuffer = element.putIfAbsent("isDone", () => null);  
 bool booleanBuffer;  
 if(valueBuffer==1){booleanBuffer=true;}else{  
 booleanBuffer=false;  
 } projectsBuffer.add(Project(id,title,description,startDate,endDate,creatorID));  
 });  
 return projectsBuffer;  
}

Листинг 3.4 – Получение списка проектов из локальный базы данных

Для реализации синхронизации был разработан ряд методов в классе DatabaseHandler и ряд отдельных обработчиков маршрутов на сервере. При входе пользователя в приложение производится синхронизация – используя идентификатор пользователя загружается вся информация, связанная с данным пользователем: пароль, организация, проекты пользователя, список задач и участников проекта. Помимо этого, синхронизацию можно провести нажав на кнопку, расположенную на странице профиля. Также, при восстановлении подключения пользователю будет предложено провести синхронизацию при помощи диалогового окна.

Список методов продемонстрирован в таблице 3.1.

Таблица 3.1 – Список методов синхронизации

|  |  |
| --- | --- |
| getAllData() | Главный метод, вызывающий дочерние |
| getOrg() | Получение информации об организации |
| getProjMembers() | Получение списка участников проектов |
| getProjects() | Получение списка проектов |
| getSubTasks() | Получение списка задач и подзадач |
| getSubTasksExecutors() | Получение списка исполнителей подзадач |

Также в приложении была реализована обратная синхронизация: в случае если у пользователя отсутствует доступ в сеть, он может внести некоторые изменения, такие как создание проекта, создание задачи или подзадачи, изменение состояния задачи, а после подключения к сети происходит выгрузка внесенных изменений в глобальную базу данных.

Для того чтобы определить какие из записей базы данных были изменены, в локальной базе данных существуют дополнительные столбцы, упомянутые в разделе «Проектирование локальной базы данных».

При запуске приложения вызывается метод uploadData, представленный в листинге 2.6.

Future<void> uploadData()async{  
 await uploadAppUserData();  
 await uploadProjectsData();  
 uploadUnboundedParentSubTasks();  
 uploadWeakCreatedChildSubTasks();  
 uploadSimplyChangedChildSubTasks();  
}

Листинг 3.5 – Метод uploadData

Метод uploadData в свою очередь вызывает ряд методов, осуществляющих обратную синхронизацию.

Выгрузка задач, созданных в рамках существующего проекта, происходит по следующему алгоритму: сперва выбираются задачи, значение поля created которых установлено в 1, но при этом значение changed проекта, к которому относится данная задача, установлено в null. Выбранные задачи загружаются на сервер, в ответе сервера мы получаем id созданной задачи, данной значение, необходимое для сопоставления локальной записи в базе данных с записью в глобальной базе данных, записывается в map. После загрузки всех задач мы переходим к выгрузке подзадач.

У каждой подзадачи, выбранной из базы данных, мы проверяем значения полей created и weakcreated. В том случае, если значение поля created установлено в 1, нам необходимо выгрузить на сервер подзадачу, созданную в рамках локально созданной задачи. Для этого мы вызываем метод uploadUnboundedChildSubTasks и используем map, для того чтобы найти id соответствующей задачи в глобальной базе данных. В противном случае мы проверяем значение поля weakcreated. В случае если оно установлено в 1, мы должны загрузить подзадачу, которая создана в рамках глобальной задачи. В такой ситуации все довольно просто: у нас уже имеется вся необходимая информация и мы просто вызываем метод uploadWeakCreatedChildSubTasks, после чего переходим к следующей подзадаче.

Алгоритм выгрузки задач и подзадач, созданных в рамках существующих проектов представлен на рисунке 3.3.



Рисунок 3.3 – Алгоритм выгрузки задач и подзадач

Представленная выше блок-схема описывает алгоритм выгрузки задач и подзадач из локальной базы данных пользователя в глобальную.

## 3.6 Выводы по главе «Програмная реализация продукта»

В данной главе была рассмотрена разработка каждого из компонентов программного продукта.

Сперва был рассмотрен спектр технических средств. Затем была рассмотрена разработка локальной и глобальной баз данных

Третий этап – разработка веб сервера с использованием Node.js

На четвертом этапе была рассмотрена разработка мобильного приложения и веб приложения.

4 Обеспечение безопасности информационной системы

`

Листов

*ФИО*

*Тихон А.А.*

*Бурмакова А.В.*

*Блинова Е.А.*

*Подп.*

*Дата*

Разраб.

Пров.

Утв.

Лит.

Лист

*БГТУ 04.00.ПЗ*

*74201055, 2024*

*Обеспечение безопасности*

*информационной системы*

Н. контр.

*Николайчук А.Н.*

1

1

Для обеспечения безопасности данных и предотвращения несанкционированного доступа к аккаунтам пользователей необходимо обеспечить хеширование паролей.

Хэширование – это процесс преобразования текстового пароля в необратимый хэш. Хэш – это набор случайных символов, который генерируется на основе текстового пароля с помощью специального алгоритма хэширования. Хэширование пароля позволяет защитить пароль от чтения злоумышленниками, так как обратить хэш в исходный пароль невозможно.

Для этого была использована библиотека crypto, предоставляющая набор хеширующих функций для языка Dart. В состав библиотеки входят алгоритмы SHA-1, SHA-224, SHA-256, MD6.

Для данного приложения был выбран алгоритм MD5 – 128-битный алгоритм хеширования, разработанный профессором Рональдом Ливестом из Массачусетского технологического института.

При этом хеширование должно осуществляться на стороне мобильного приложения, так как в ходе передачи пароля на сервер, он может быть перехвачен злоумышленниками.

Использование библиотеки crypto представлено в листинге 4.1.

string password = loginFieldController.text;  
string hashedPassword = md6.convert(utf8.encode(password)).toString()

Листинг 4.1 – Использование библиотеки crypto

Сперва необходимо прочитать введенный пользователем текст, затем используя метод convert, в который мы передаем текст, который необходимо хэшировать. Данный метод вернет введенный пароль в виде дайджеста, который затем передается серверу для записи в базу данных.

## 4.1 Вывод по главе «Обеспечение безопасности информационной системы»

В данной главе было рассмотрено обеспечение безопасности информационной системы проекта.

Так как в данном дипломном проекте присутствует авторизация пользователя, то было решено использовать хеширование паролей в качестве обеспечения информационной безопасности системы.

5 Тестирование приложения

`

Листов

*ФИО*

*Тихон А.А.*

*Бурмакова А.В.*

*Бурмакова А.В.*

*Блинова Е.А.*

*Подп.*

*Дата*

Разраб.

Пров.

Утв.

Лит.

Лист

*БГТУ 05.00.ПЗ*

*74201055, 2024*

*Тестирование приложения*

Н. контр.

*Николайчук А.Н.*

1

3

Тестирование приложений является неотъемлемой частью разработки программного обеспечения. Оно позволяет обнаруживать ошибки, проверять функциональность и качество приложения перед его выпуском. В данном разделе рассматриваются основные аспекты тестирования приложения, включая его цели, подходы, методы и инструменты, которые помогут в создании надежного и стабильного приложения.

Цель тестирования заключается в обнаружении ошибок, проверке функциональности и качества приложения, а также обеспечении надежности и стабильности его работы. Тестирование помогает убедиться, что приложение соответствует требованиям и ожиданиям пользователей, исполняет свои функции без сбоев, а также обеспечивает безопасность и защиту данных.

## 5.1 Тестирование валидации данных

При использовании мобильного приложения пользователю необходимо много взаимодействовать с данными. При создании проекта, задачи, организации пользователю необходимо ввести название проекта, назначить исполнителя задачи, выбрать пароль для организации. В том случае если этого не будет сделано, в качестве последствий возникнут проблемы хранения этих данных, нарушения зависимостей в базе данных. Для того чтобы предотвратить возникновение таких ситуаций необходимо проводить валидацию. Использование валидации позволяет остановить пользователя еще на этапе ввода данных.

Рассмотрим валидацию данных на примере создания проекта. При создании проекта пользователю необходимо ввести название и описание проекта, а также выбрать даты начала и завершения проекта. Добавление участников является опциональным и может быть выполнено после создания проекта.

На рисунке 5.1 представлена форма, которую пользователю необходимо заполнить для создания проекта.

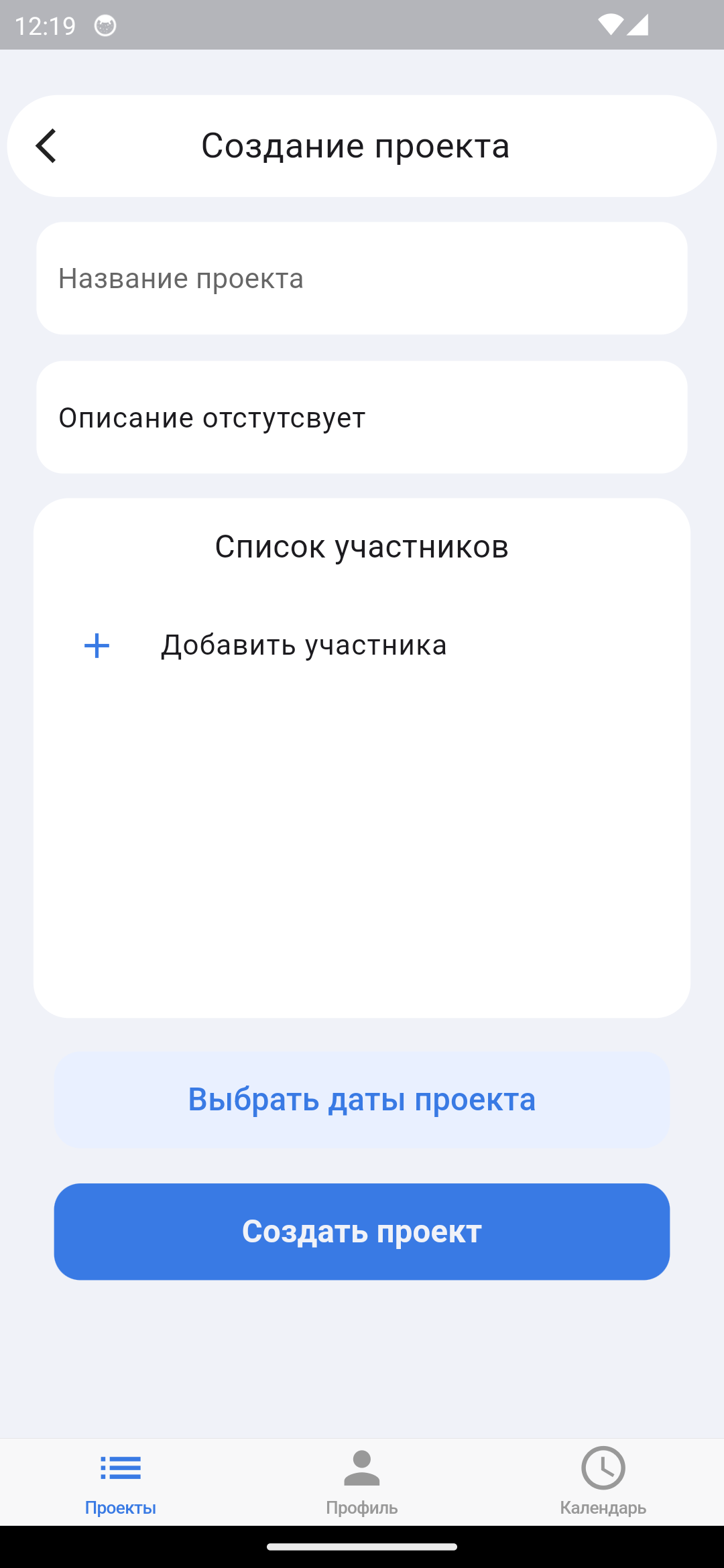


Рисунок 5.1 – Форма для создания проекта

В том случае, если пользователь попытается создать проект, не выбрав названия для проекта, проект не будет создан и пользователю будет выведено соответствующее сообщение, представленное на рисунке 5.2

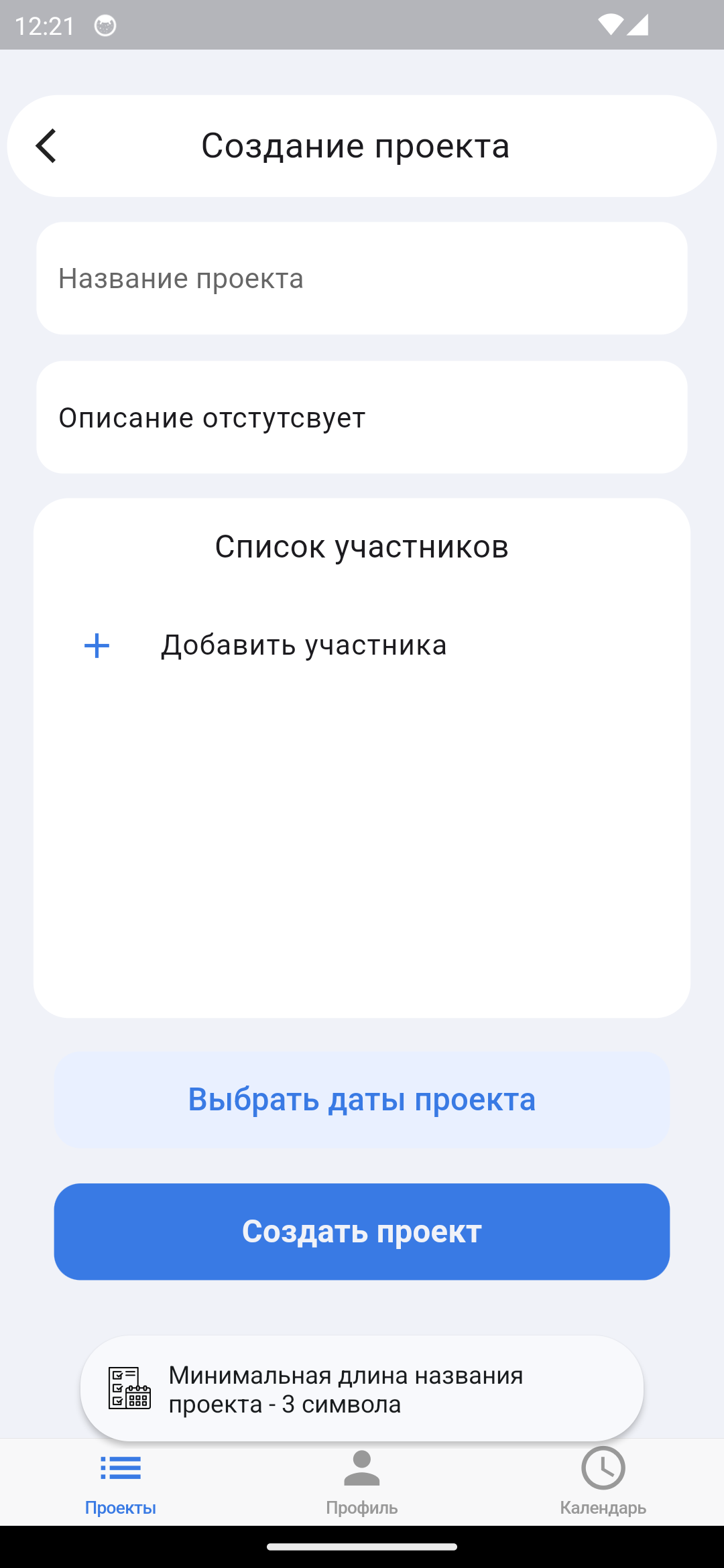


Рисунок 5.2 – Сообщение об ошибке при создании проекта

Аналогичная ситуация произойдет при попытке создать проект, не выбрав даты начала и завершения проекта.

Единственное отличие – текст выводимого сообщения, представленный на рисунке 5.3.

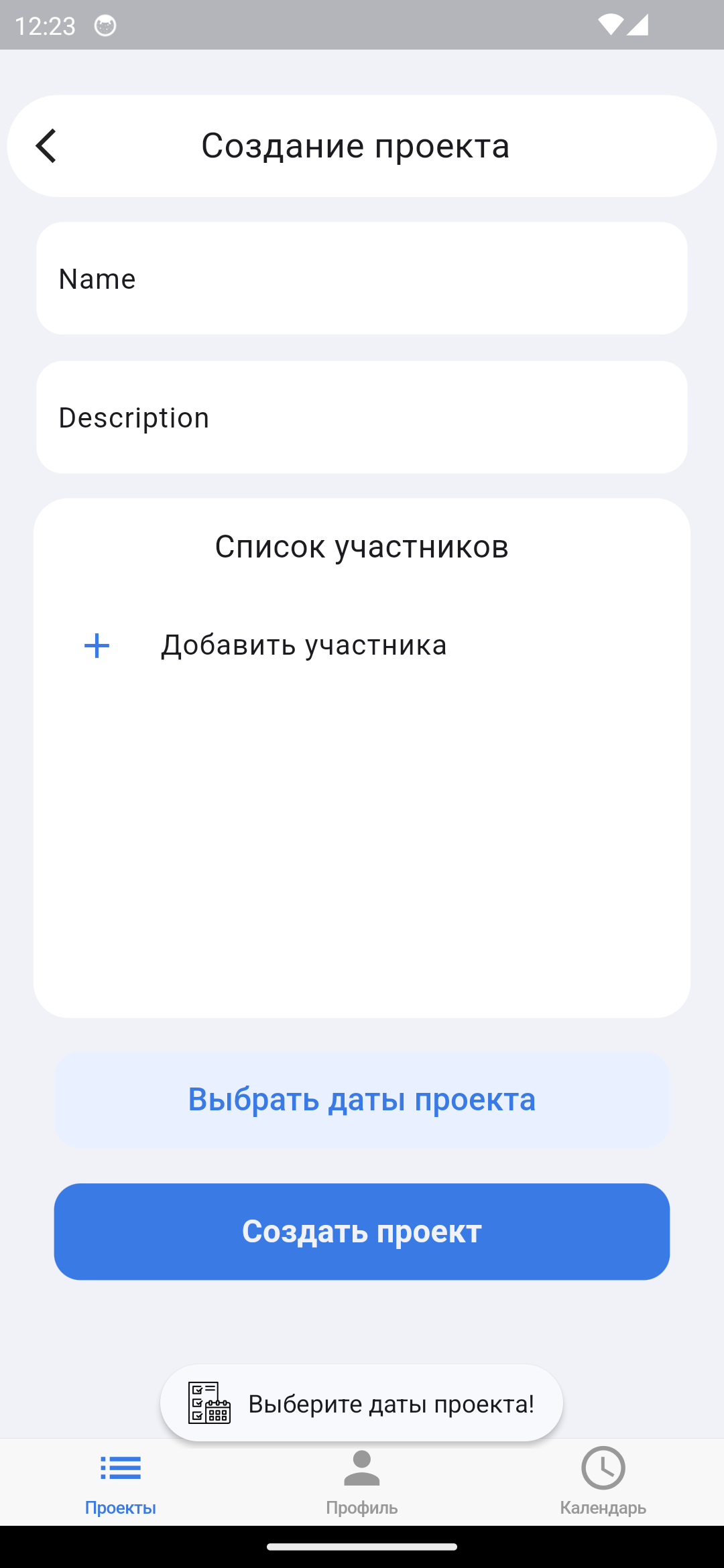


Рисунок 5.3 – Сообщение об ошибке при создании проекта

Процесс валидации является важным процессом и представляет собой проверку соответствия разрабатываемого или изменяемого продукта или системы установленным требованиям и ожиданиям.

5.2 Выводы по разделу

В данном разделе были описаны общие понятия процесса тестирования приложений, а также были рассмотрены сценарии создания кошельков и транзакций, при которых пользователь предоставляет недостаточно данных или эти данные указаны в неверном формате. Были рассмотрены процессы валидации входящих данных в этих сценариях.

6 Руководство пользователя

`

Листов

*ФИО*

*Тихон А.А.*

*Бурмакова А.В.*

*Бурмакова А.В.*

*Блинова Е.А.*

*Подп.*

*Дата*

Разраб.

Пров.

Утв.

Лит.

Лист

*БГТУ 06.00.ПЗ*

*74201055, 2024*

*Руководство пользователя*

Н. контр.

*Николайчук А.Н.*

1

12

## 6.1 Руководство пользователя мобильного приложения

При входе в приложение пользователь попадает на экран входа (рис.6.1).

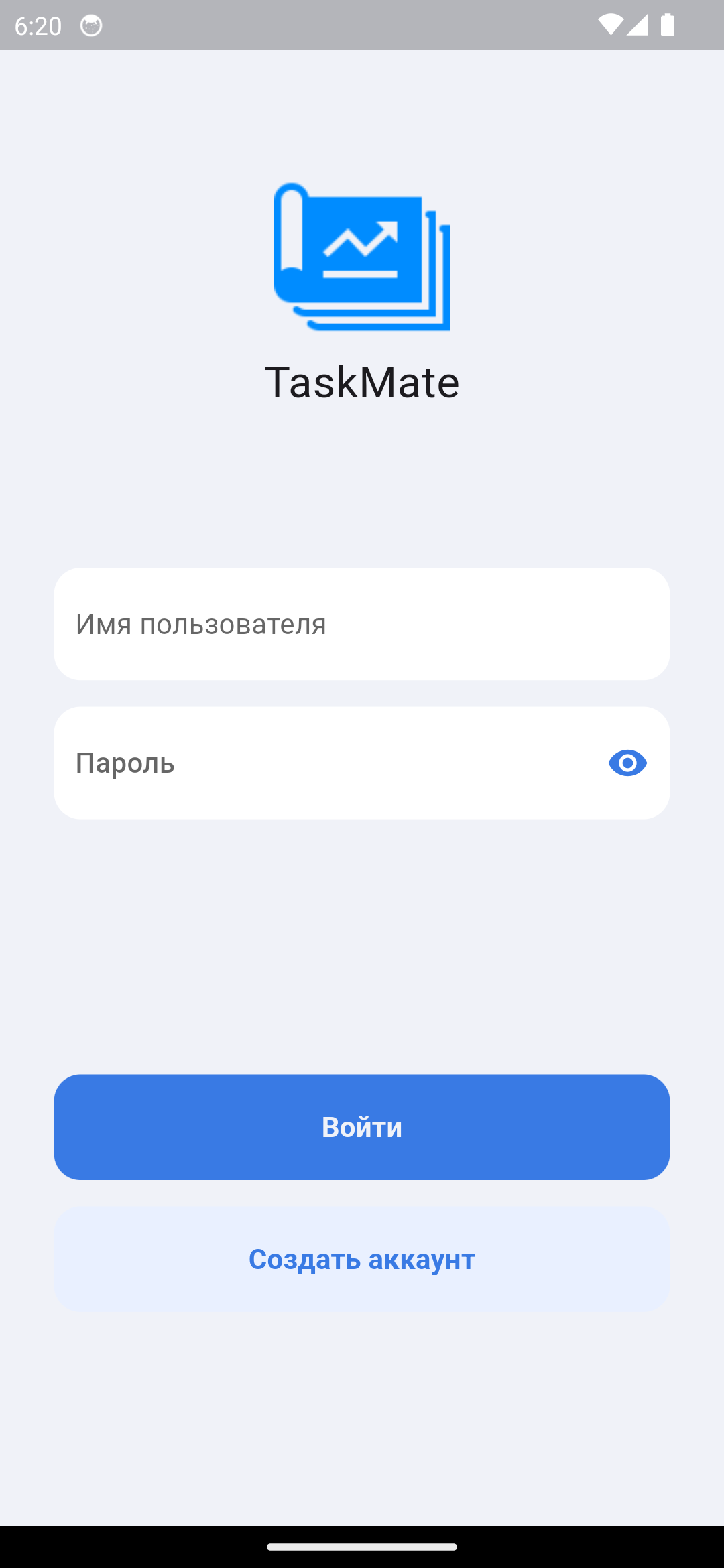


Рисунок 6.1 – Страница «Вход» мобильного приложения

Сперва пользователю необходимо нажать на кнопку «Создать аккаунт», после чего будет совершен переход на страницу «Регистрация» (рис. 6.2).

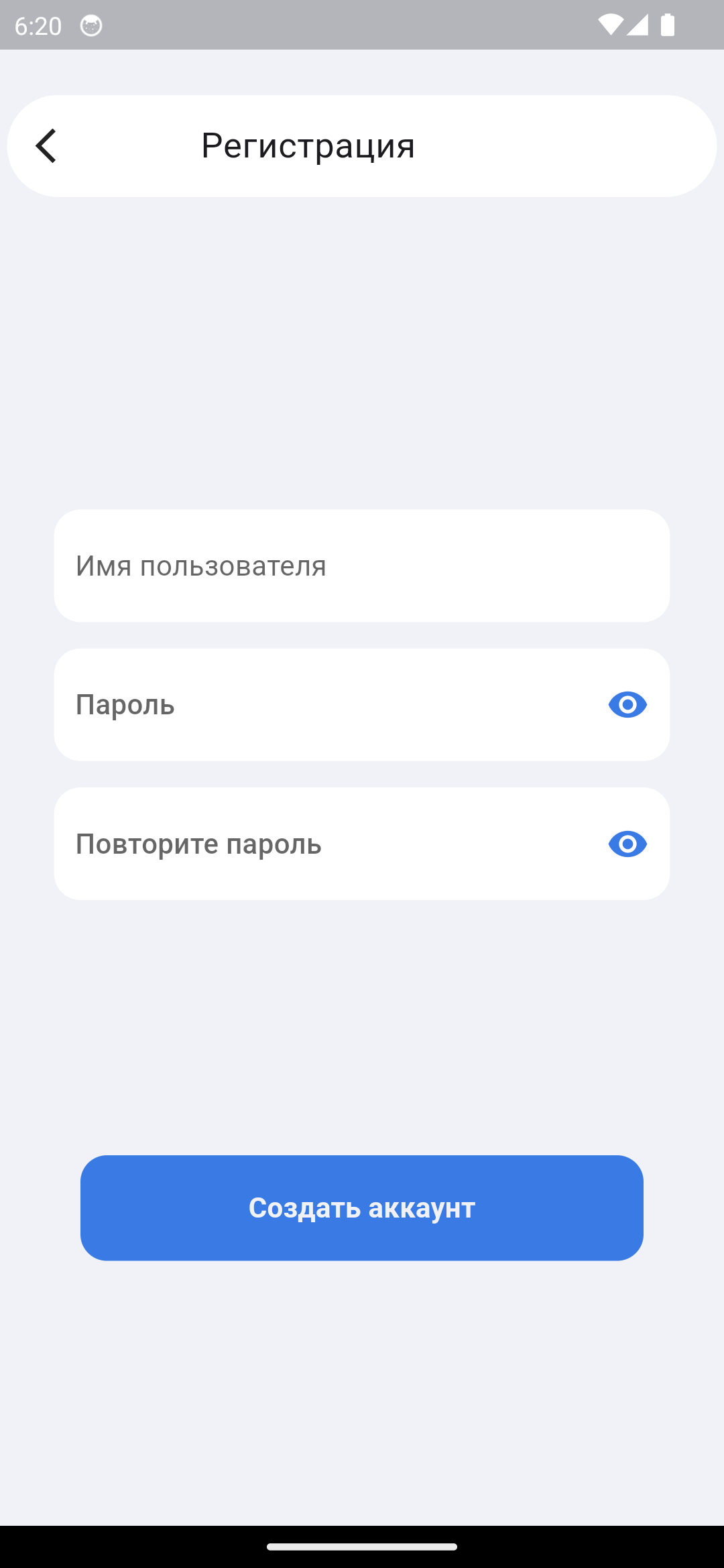


Рисунок 6.2 – Страница «Регистрация» мобильного приложения

После регистрации пользователь возвращается на страницу входа и вводит свои данные, после чего нажимает на кнопку «Войти» и переходит на первую из двух главных страниц – страницу «Все проекты».

Страница все проекты представлена на рисунке 6.2.

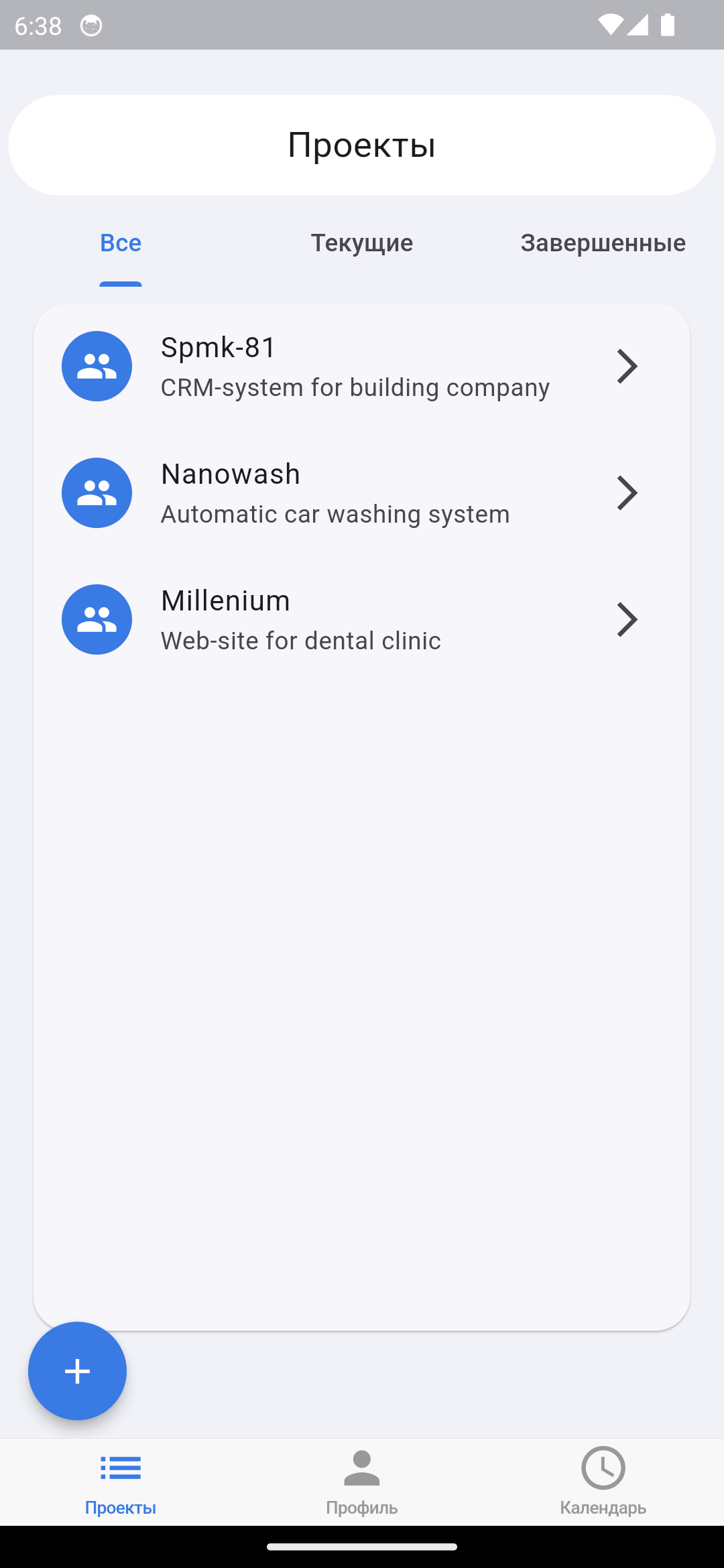


Рисунок 6.3 – Страница «Все проекты» мобильного приложения

На странице «Все проекты» пользователь видит список текущих проектов, для создания нового проекта пользователю необходимо нажать на кнопку «Добавить», расположенную в нижнем левом углу. После нажатия на кнопку добавить пользователь перенесется на страницу «Создание проекта» (рис 6.4).

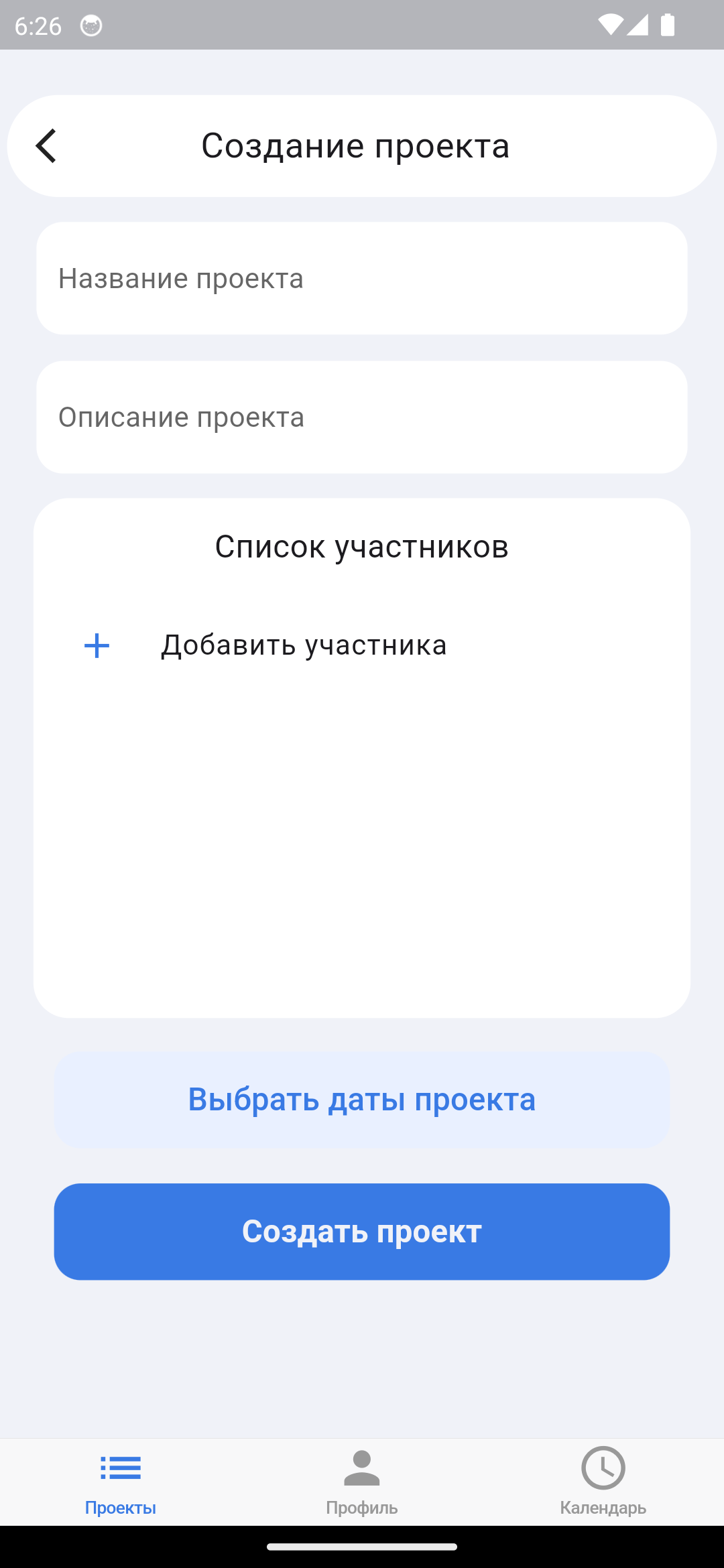


Рисунок 6.4 – Страница «Создание проекта»

На данном экране пользователь вводит название и описание проекта, выбирает даты начала и конца и добавляет участников в проект. После ввода информации пользователь нажимает на кнопку «Создать проект».

Для того чтобы открыть страницу конкретного проекта пользователю необходимо нажать на проект, после этого пользователь переходит на страницу конкретного проекта (рис. 6.5).

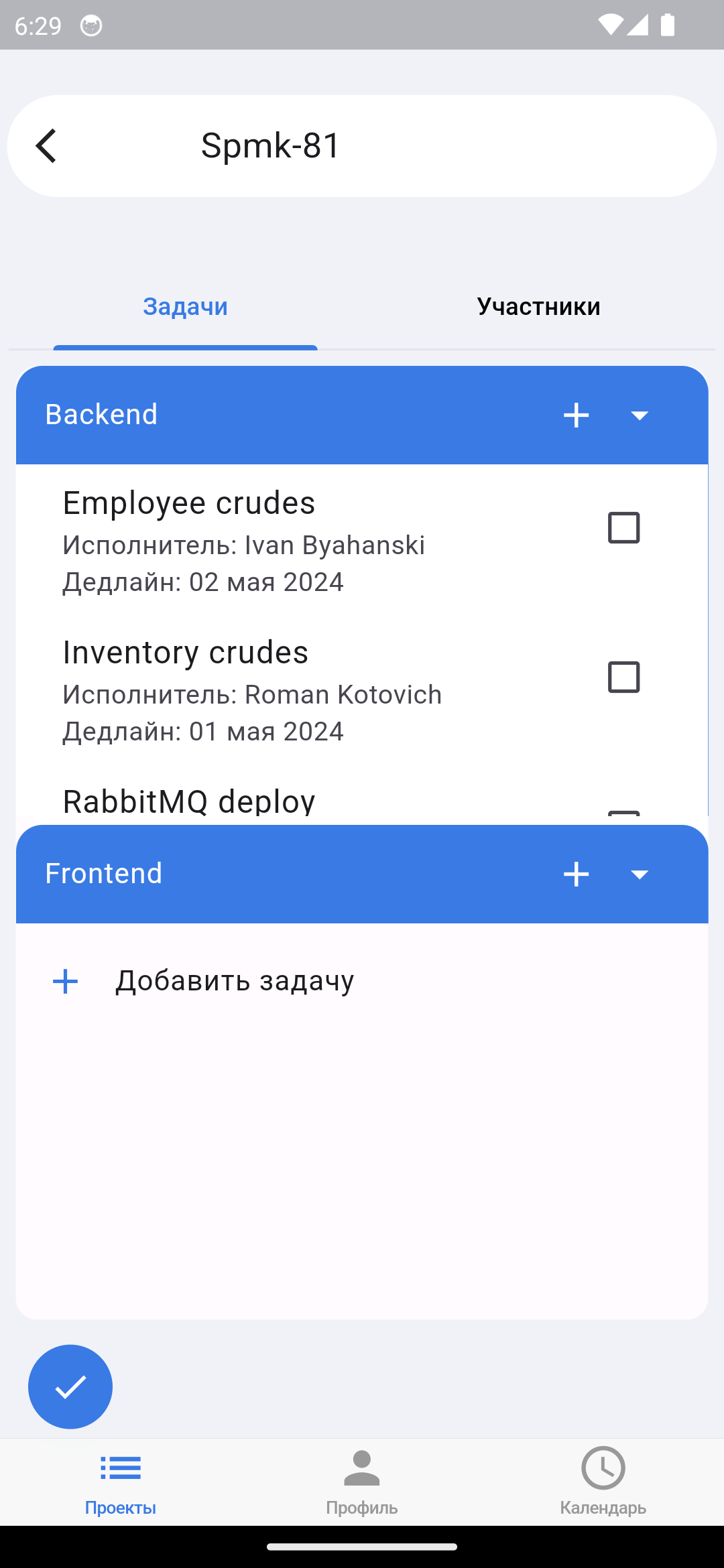


Рисунок 6.5 – Страница конкретного проекта

На данной странице пользователь может добавить участника, нажав на кнопку «Добавить участника» и выбрав в диалоговом окне участника организации.

Диалоговое окно для добавления участника проекта представлено на рисунке 6.6.

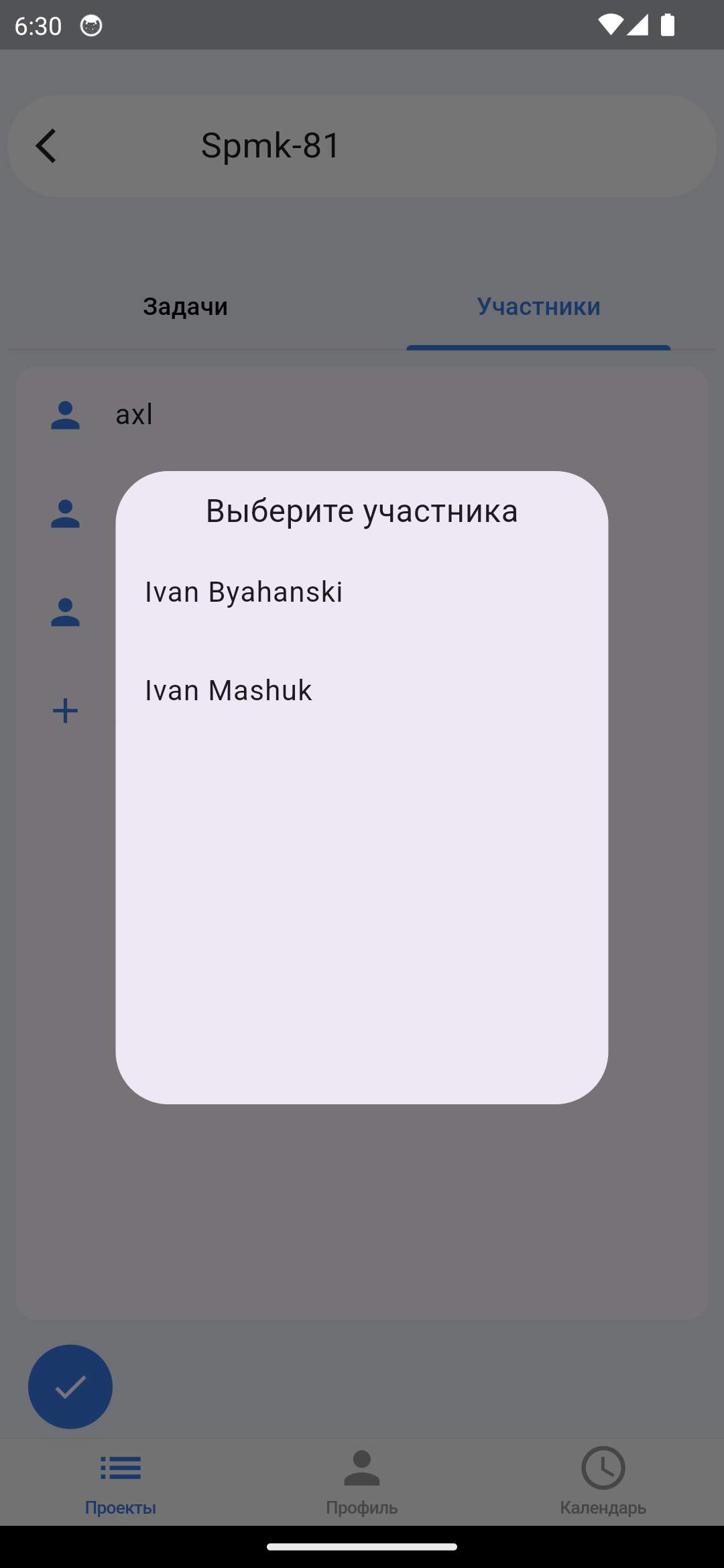


Рисунок 6.6 – Добавление участника в проект

Для добавления задачи пользователю необходимо нажать на кнопку «Добавить задачу», после чего откроется диалоговое окно, в которое пользователю необходимо ввести имя задачи.

На рисунке 6.7 представлено диалоговое окно, в которое необходимо ввести имя задачи.

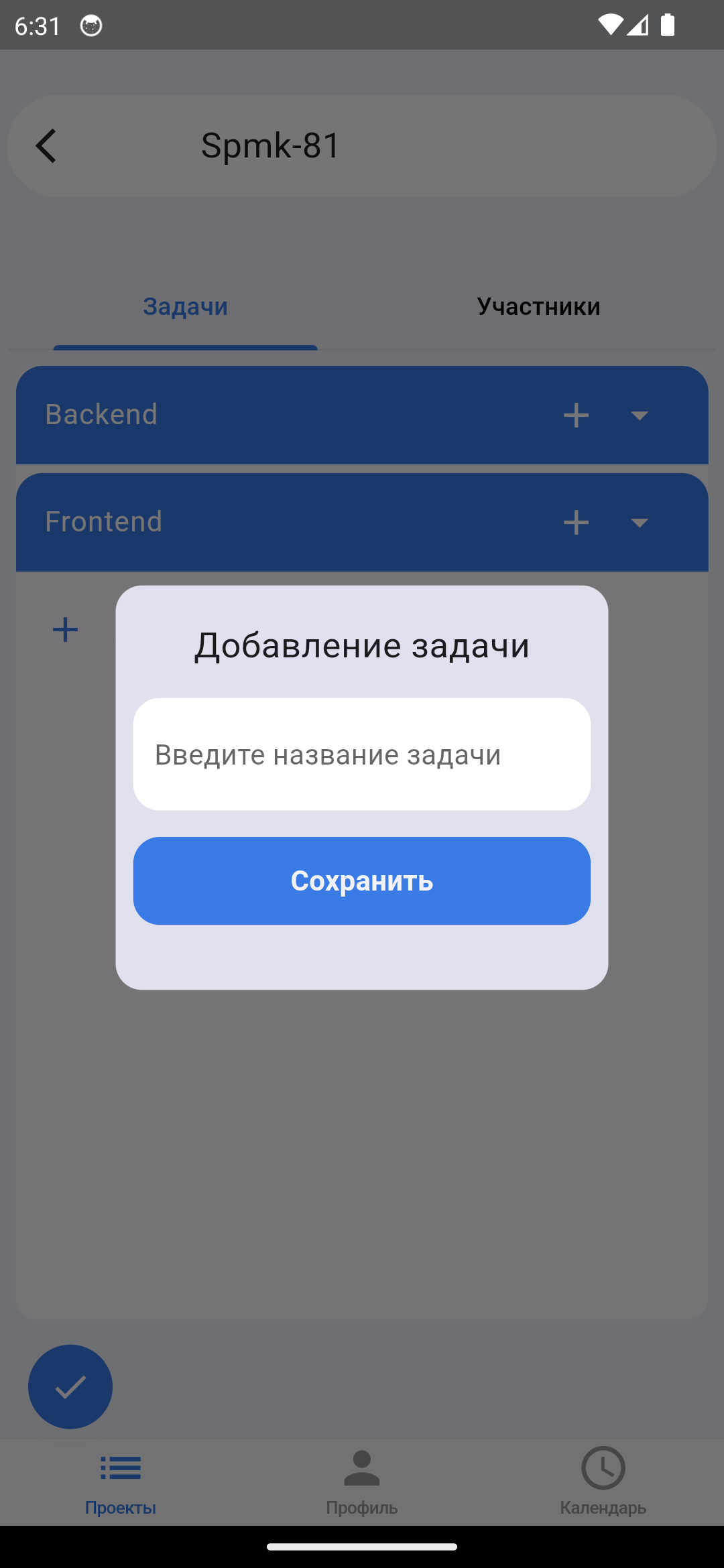


Рисунок 6.7 – Добавление задачи в проект

Для добавления подзадачи пользователю необходимо на кнопку «Добавить подзадачу», которая является родительской для добавляемой подзадачи, после чего откроется диалоговое окно для добавления подзадачи. (рис. 6.8).

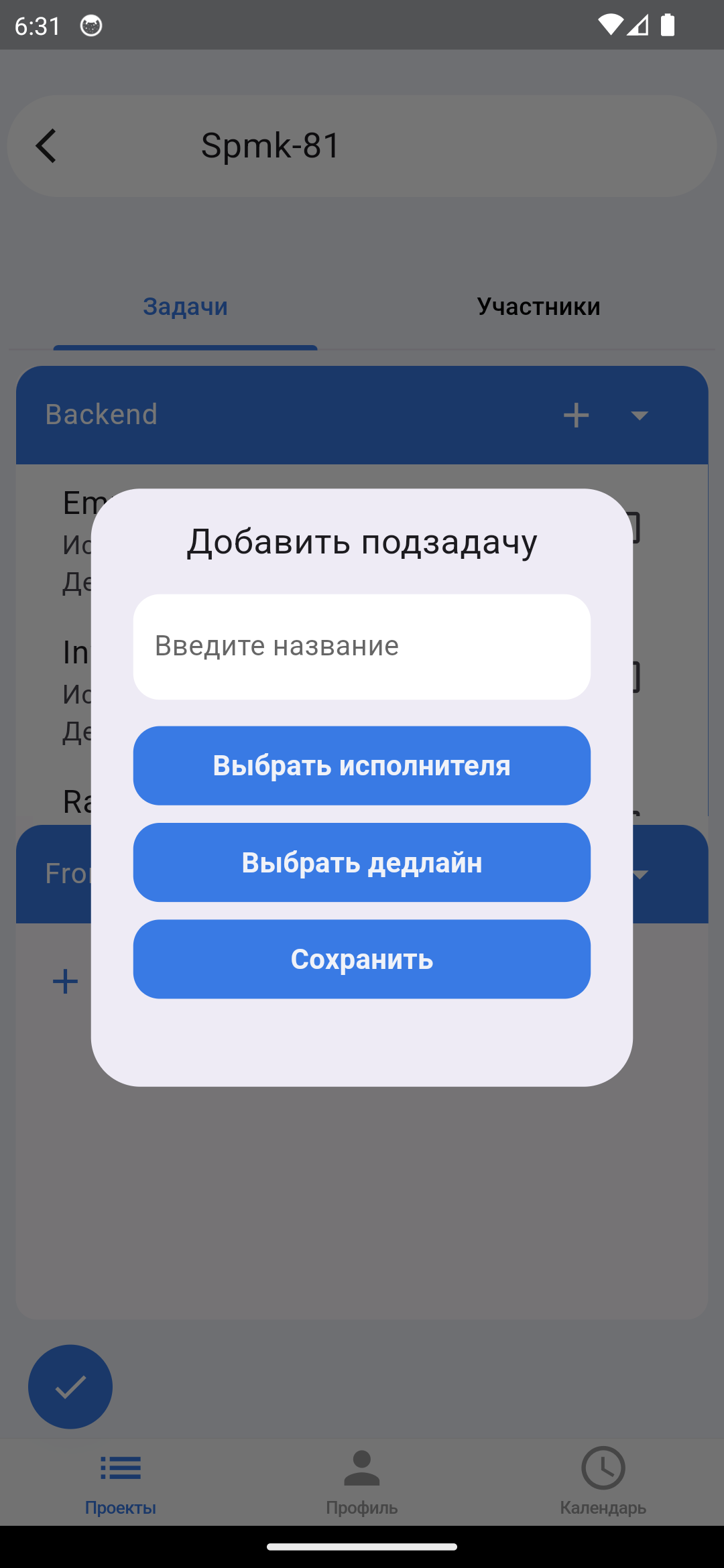


Рисунок 6.8 – Добавление подзадачи в проект

В данном окне пользователю необходимо ввести название подзадачи и выбрать её исполнителя, после чего нажать на кнопку «Сохранить».

Второй из главных страниц является страница «Профиль» (рис. 6.9).

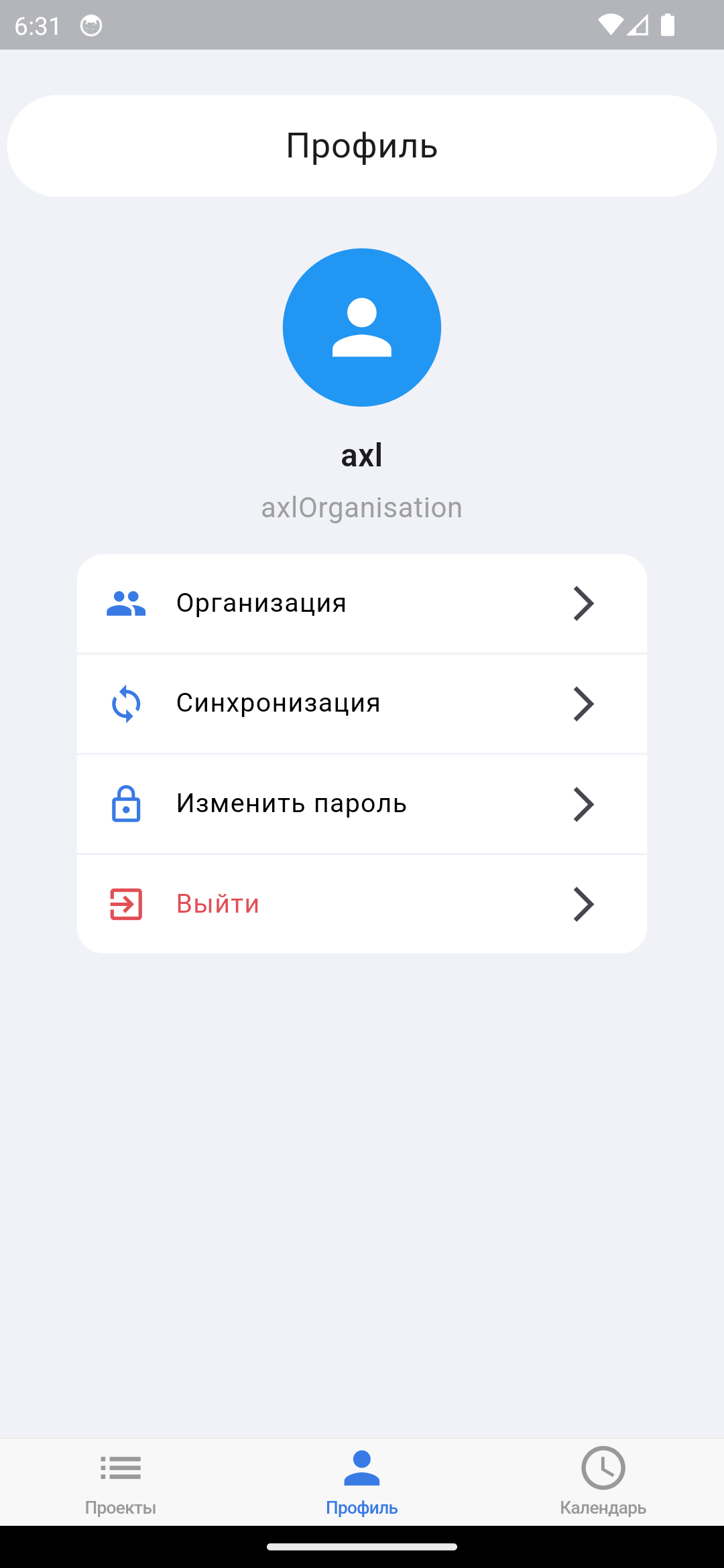


Рисунок 6.9 – Страница «Профиль»

На странице «Профиль» пользователь имеет возможность сменить пароль, выйти из аккаунта и провести синхронизацию.

При нажатии на кнопку «Вступить в организацию», если пользователь на данный момент не состоит в организации, будет совершен переход на страницу «Вступить в организацию» (рис. 6.10).

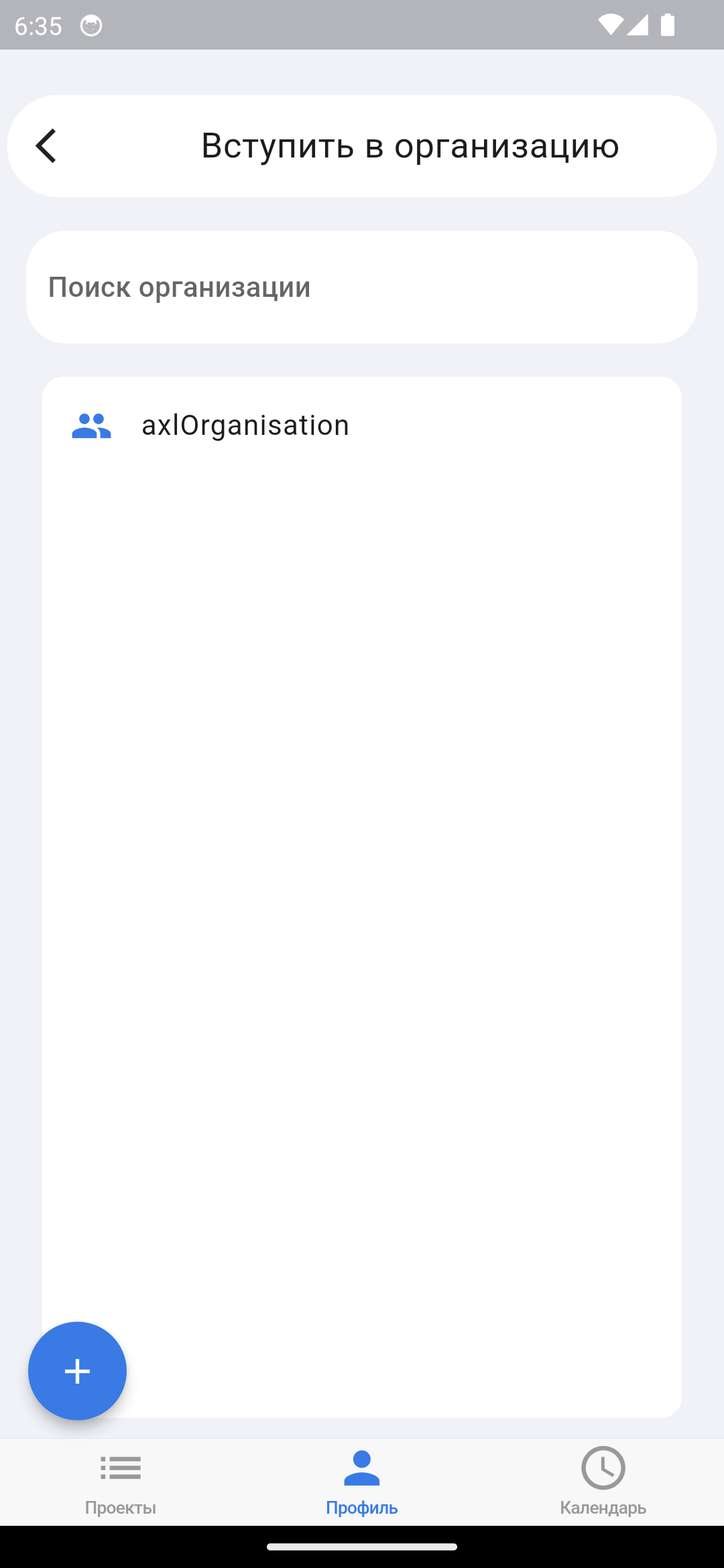


Рисунок 6.10 – Страница «Вступить в организацию»

На странице «Вступить» в организацию пользователь может создать свою организацию, нажав на кнопку добавить, расположенную в нижнем левом углу.

После нажатия на кнопку создать организацию пользователь перейдет на страницу «Создание организации» (рис. 6.11).

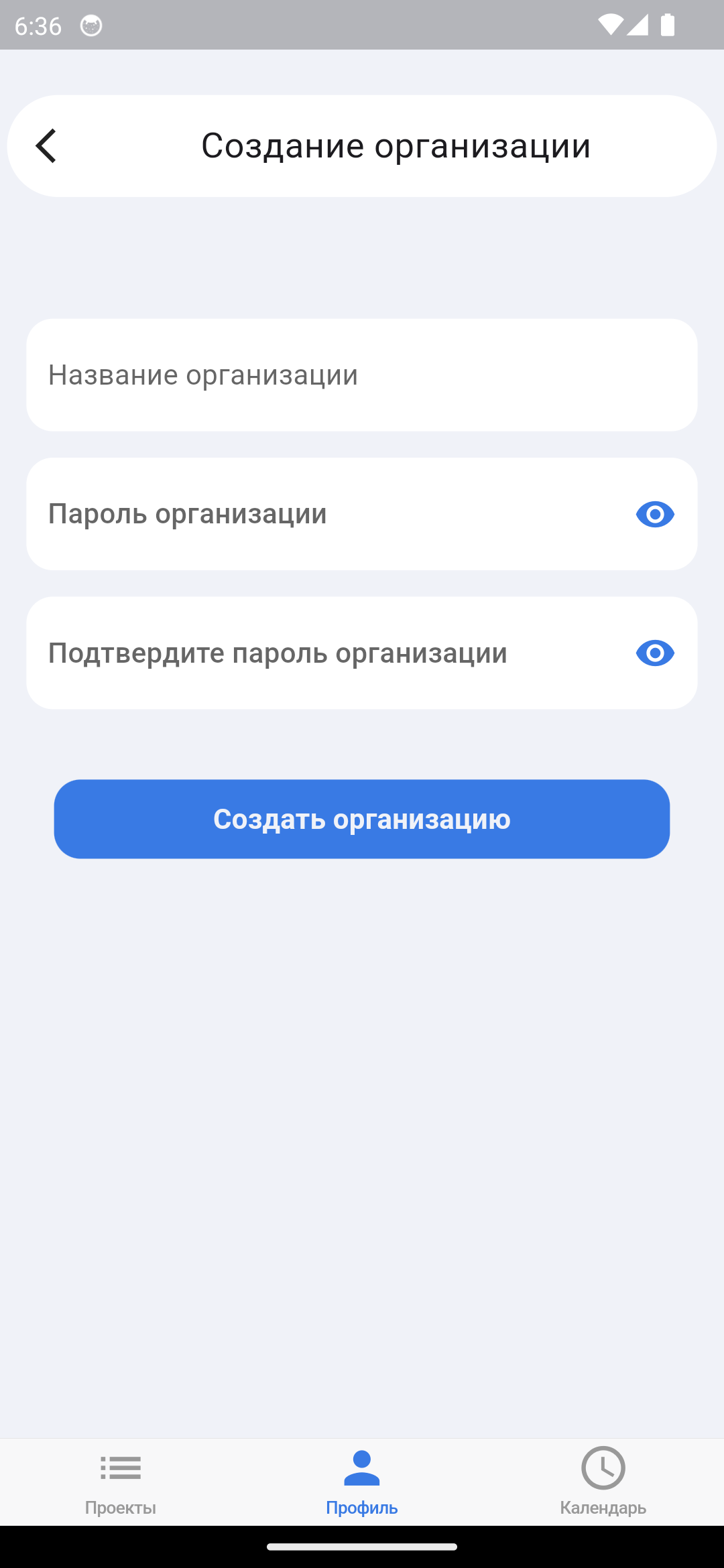


Рисунок 6.11 – Страница «Создание организацию»

На данной странице пользователь должен ввести необходимую информацию и нажать на кнопку «Создать организацию», после чего организация будет создана.

Также на странице «Профиль» пользователь может открыть страницу управления организацией, нажав на элемент с названием названия организации (рис. 6.12).

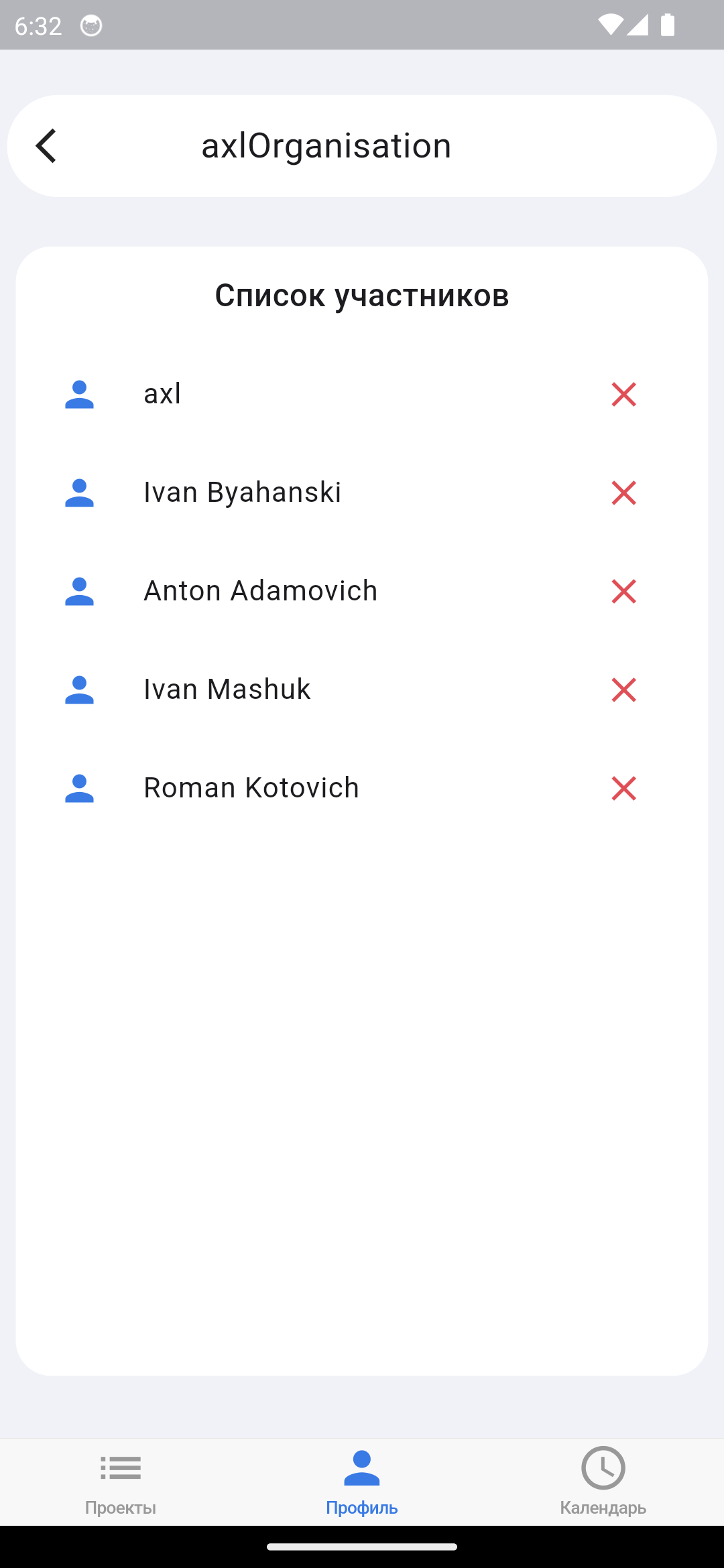


Рисунок 6.12 – Страница управления организацией

На данной странице пользователь может изменить название организации, нажав на кнопку «Сохранить изменения». Также пользователь может удалить участника, нажав на кнопку удаления, расположенную напротив имени пользователя.

## 6.2 Руководство пользователя веб приложения

При запуске веб приложения пользователь переходит на страницу «Вход» (рис. 6.13).

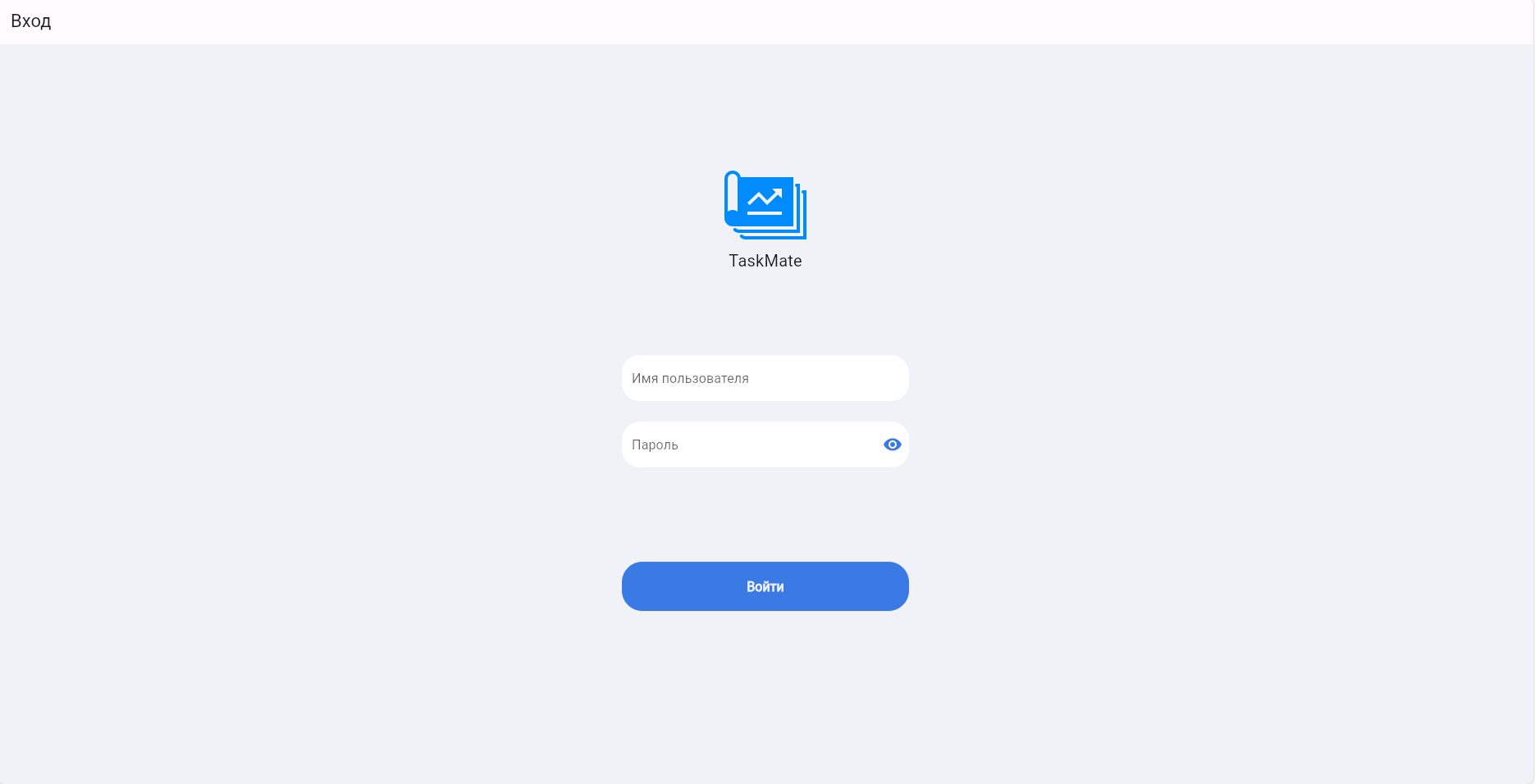


Рисунок 6.13 – Страница «Вход» веб приложения

После нажатия на кнопку вход пользователь переходит на страницу «Проекты» (рис. 6.14).

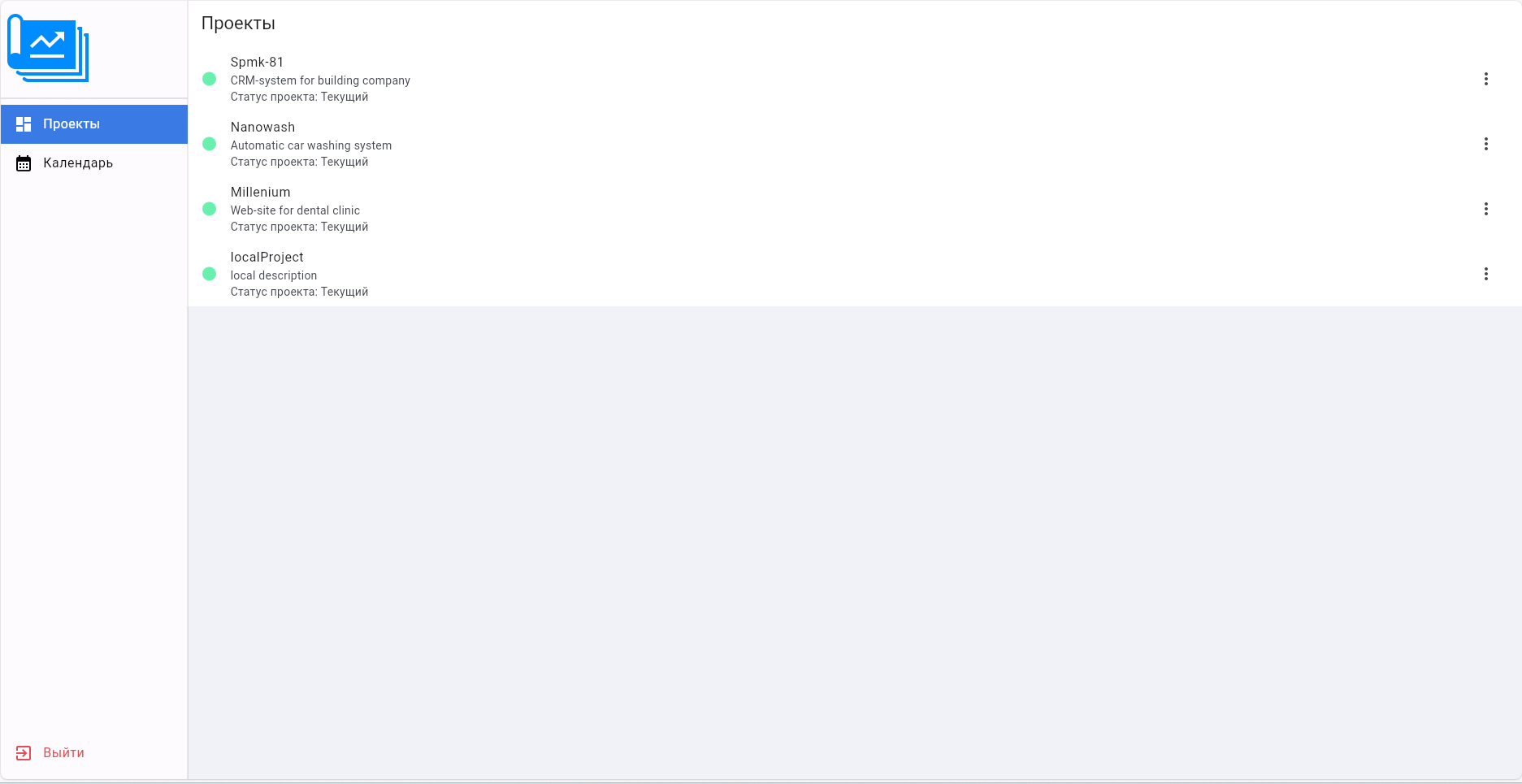


Рисунок 6.14 – Страница «Проекты» веб приложения

На данной странице пользователь может выполнить три действия: продлить дату проекта, завершить проект или удалить проект. Для выполнения данных действий пользователю необходимо совершить долгое нажатие на проект, надо которым он хочет совершить действие. После этого появится контекстное меню со списком действий (рис. 6.15).

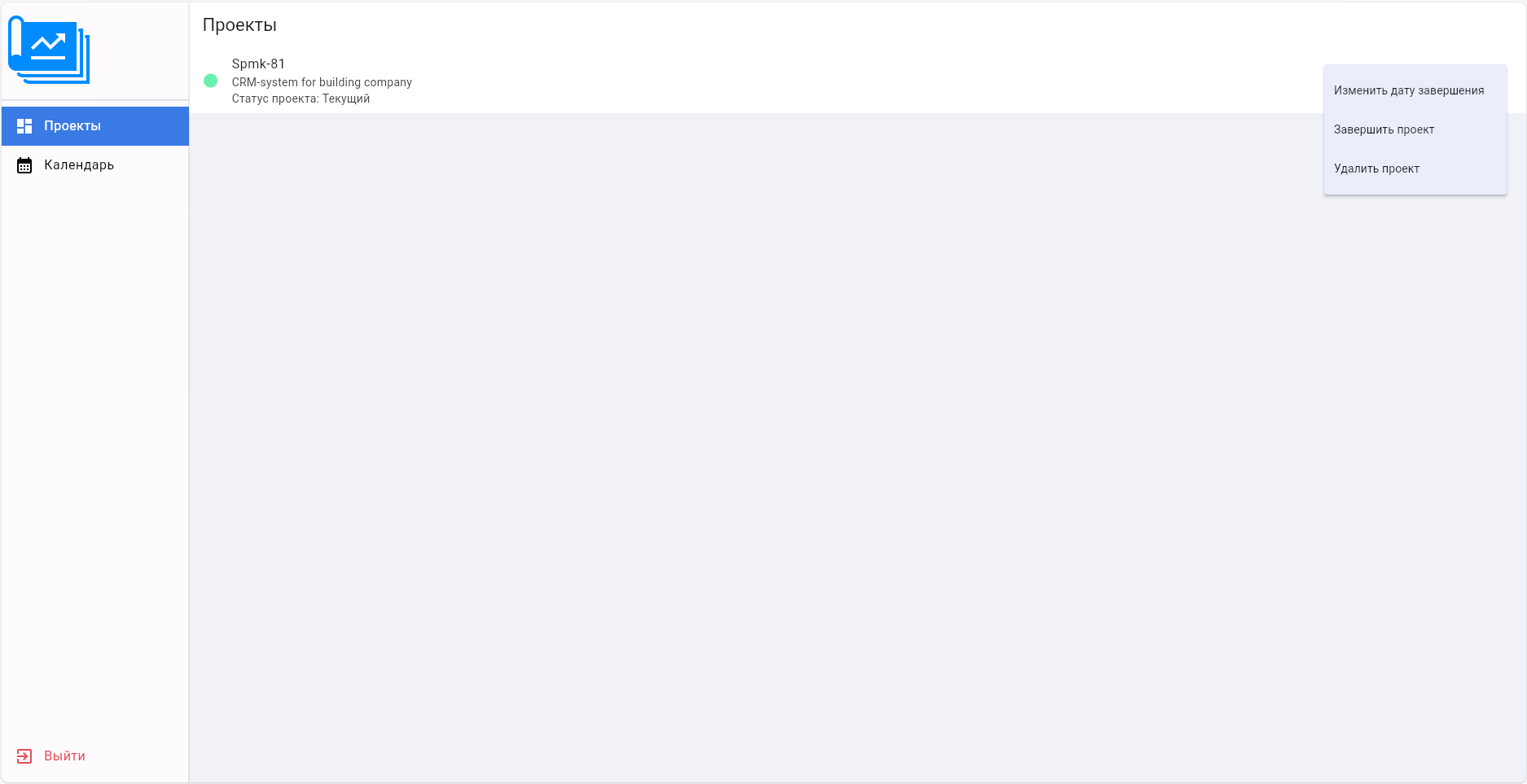


Рисунок 6.15 – Контекстное меню со списком действий

Контекстное меню предоставляет три возможных действия: продление проекта, удаление проекта и завершение проекта.

При нажатии на элемент меню «Продлить дату проекта» открывается календарь для выбора новой даты завершения проекта (рис.6.16).

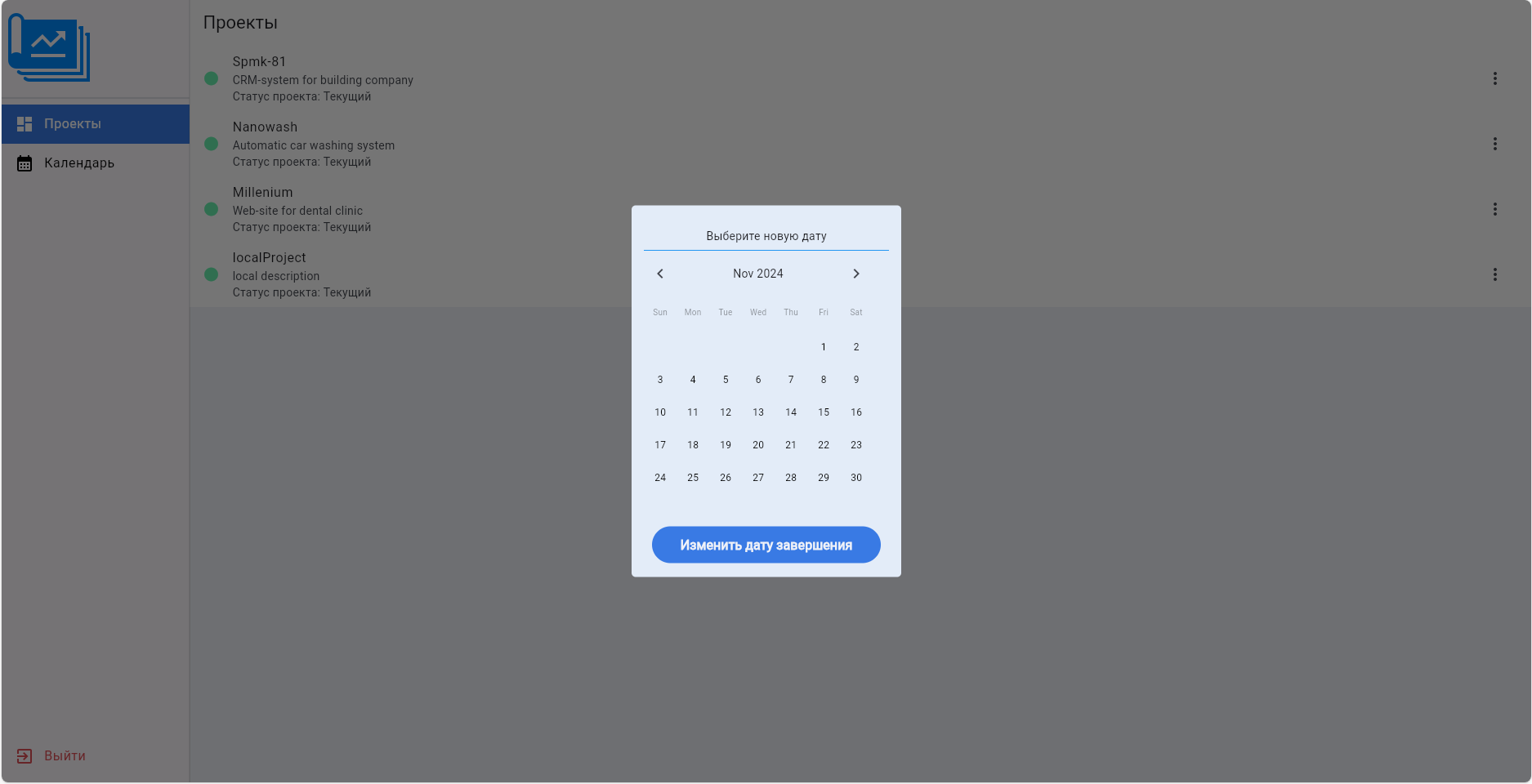


Рисунок 6.16 – Календарь для выбора новой даты завершения проекта

При нажатии на элемент меню «Удалить проект» будет выполнен запрос к серверу и проект будет удален. После нажатия на кнопку «Завершить проект», будет проведена проверка, все ли задачи проекта выполнены, и в случае, если все задачи проекта выполнены значение поля isDone данного проекта будет установлено в 1 и проект будет перемещен в раздел «Завершенные проекты» (рис. 6.17).

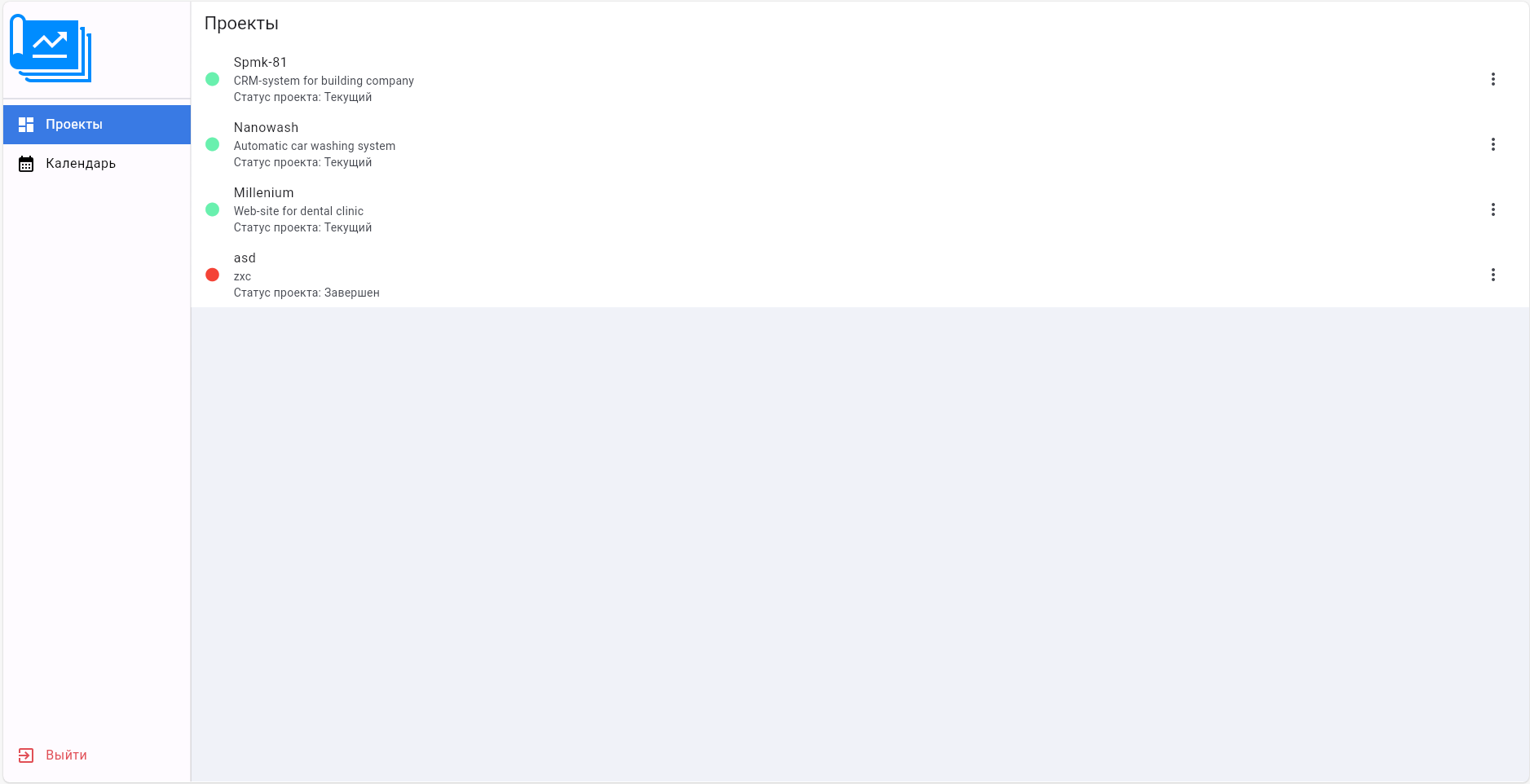


Рисунок 6.17 – Завершенные проекты

Для непосредственного управления проектом пользователю необходимо нажать на один из проектов, после чего произойдет переход на страницу для управления конкретным проектом для выбранного проекта. Страница управления конкретным проектом представлена на рисунке 6.18.

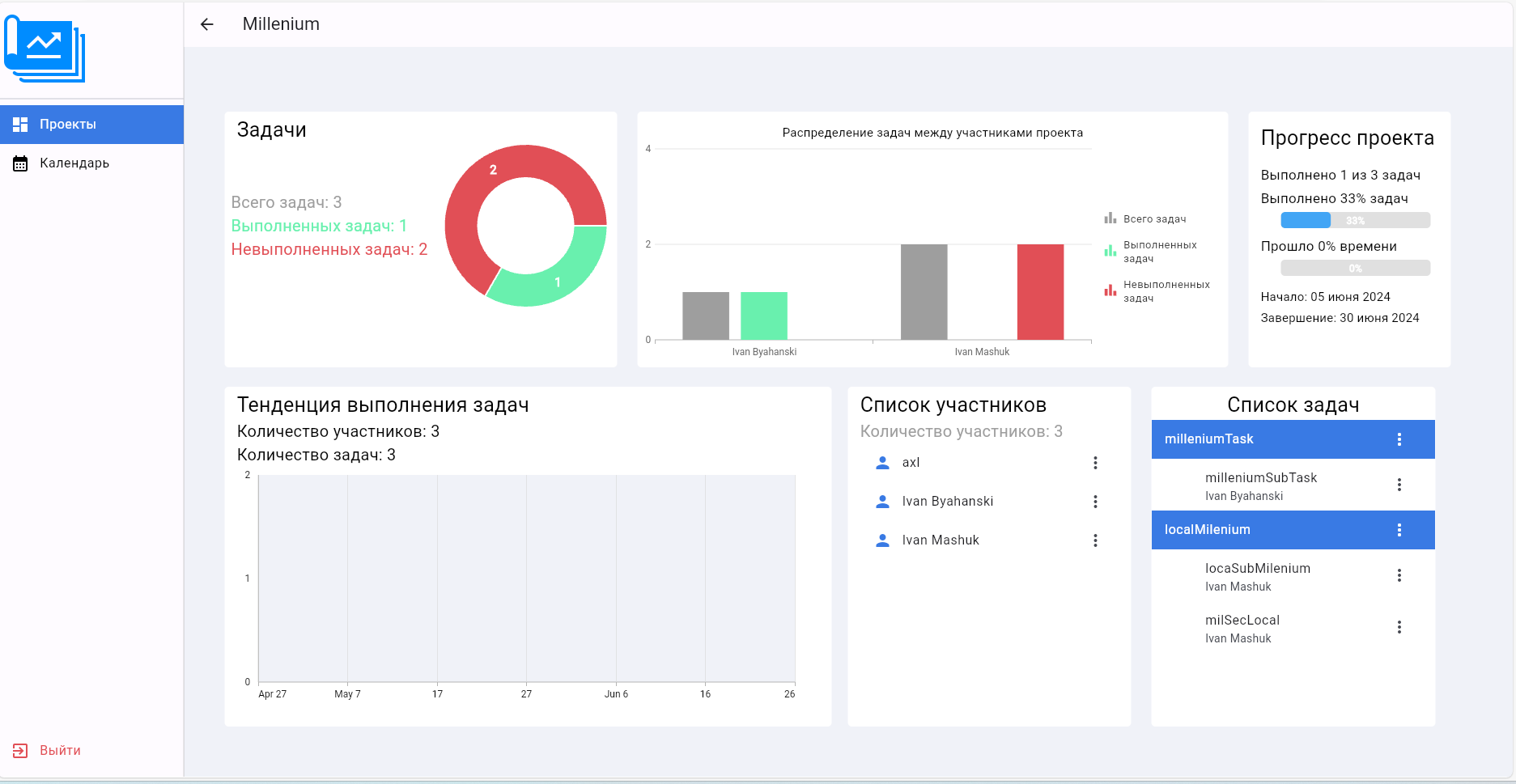


Рисунок 6.18 – Страница проекта

На данной странице пользователь может просмотреть задачи и подзадачи проекта, а также удалить задачу, подзадачу или исключить пользователя из проекта. Для совершения данных действий необходимо нажать на соответствующую кнопку, расположенную напротив названия задачи или имени пользователя.

Также в панели администратора есть возможность представления все поставленных задач и проектов в виде расписания для более удобного пользователю представления.

Для того чтобы открыть страницу с расписанием необходимо нажать в боковой панели на соответствующую вкладку.

Представление задач в виде расписания представлено на рисунке 6.19.

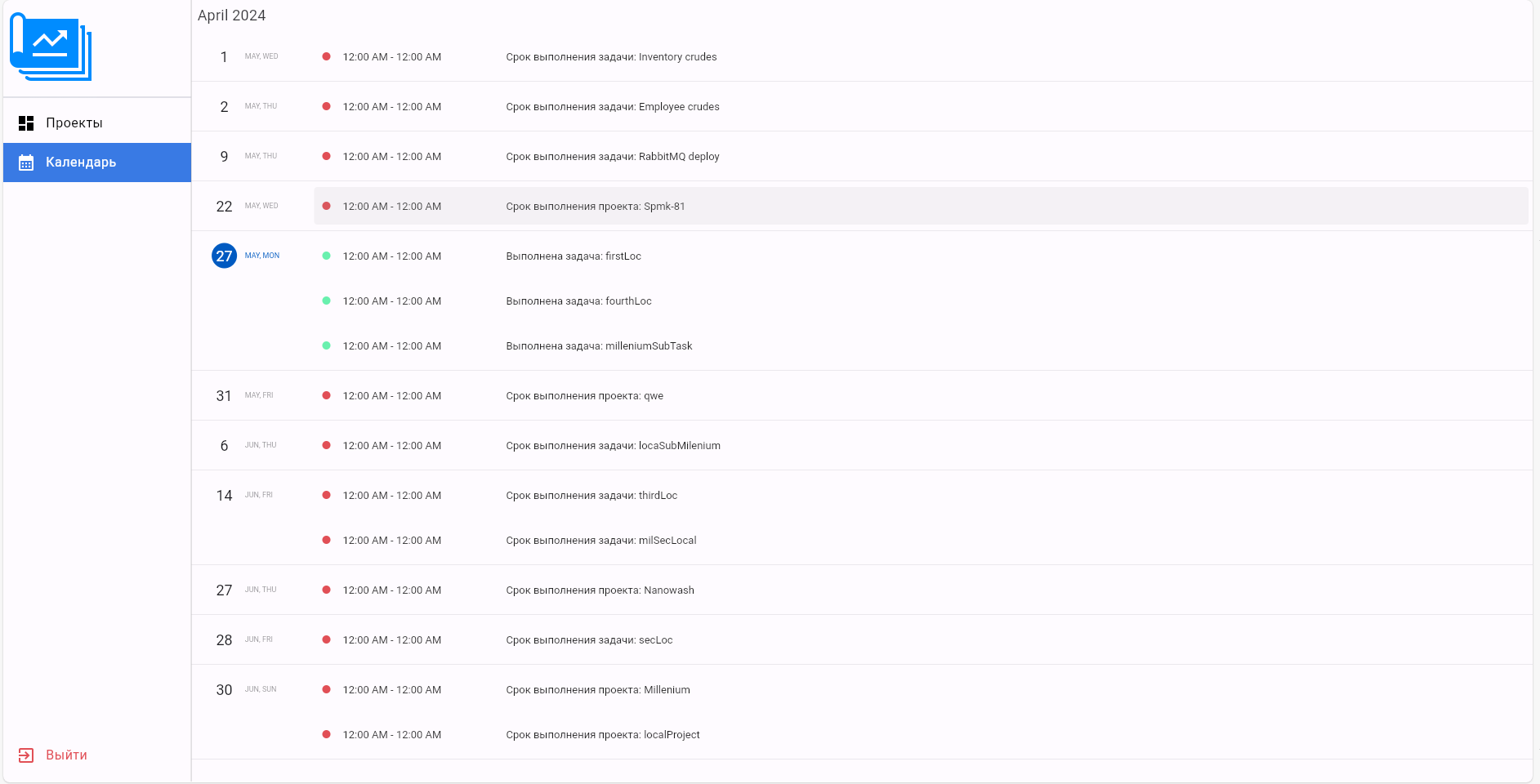


Рисунок 6.18 – Представление задач в виде расписания

Как видно из рисунка 6.18 задачи приводятся списком и отмечаются маркером.

**7 Технико-экономическое обоснование проекта**

Листов

*ФИО*

*Тихон А.А.*

*Бурмакова А.В.*

*Блинова Е.А.*

*Подп.*

*Дата*

Разраб.

Пров.

Утв.

Лит.

Лист

*БГТУ 07.00.ПЗ*

*74201055, 2024*

*Технико-экономическое*

*обоснование проекта*

Н. контр.

*Николайчук А.Н.*

1

9

**7.1** **Общая характеристика разрабатываемого программного средства**

Во время разработки дипломного проекта использовались технологии Express, Flutter и язык программирования Dart. Разработанное программное средство представляет собой мобильное приложение, предназначенное для управления и ведения проектов TaskMate. Приложение необходимо для повышения продуктивности, производительности и снижения количества ошибок.

По результатам анализа применяемых продуктами-аналогами стратегий монетизации следует выбрать стратегию монетизации: расширенная подписка на использование продукта (пользователь покупает подписку, которая расширяет функционал приложения).

**7.****2 Исходные данные для проведения расчётов**

Источниками исходных данных для данных расчетов выступают действующие законы и нормативно-правовые акты. Исходные данные для расчета приведены в таблице 7.1.

Таблица 7.1 – Исходные данные для расчета

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование показателя | Единица измерения | Условные обозначения | Норматив |
| Численность разработчиков | чел. | Чр | 1,0 |
| Норматив дополнительной заработной платы | % | Ндз | 15,0 |
| Ставка отчислений в Фонд социальной защиты населения | % | Нфсзн | 34,0 |
| Ставка отчислений в БРУСП «Белгосстрах» | % | Нбгс | 0,6 |
| Цена одного машино-часа | руб. | Смч | 0,5 |
| Норматив прочих затрат | % | Нпз | 18,5 |
| Норматив накладных расходов | % | Нобп, обх | 10,0. |
| Норматив расходов на сопровождение и адаптацию | % | Нрса | 17,0 |
| Ставка НДС | % | ННДС | 20,0 |

**7.3 Объем программного средства**

В таблице 7.2 указаны в укрупнённом виде все работы, реально выполненные для создания, указанного в дипломной работе программного средства и количество рабочих дней, реально потраченных для выполнения этих работ. Были учтены затраты на разработку программного средства, поиск нужных компонентов платформы, куда будет интегрировано программное средство, а также тестирование и написание руководства пользователя.

Таблица 7.2 – Затраты рабочего времени на разработку ПС

|  |  |
| --- | --- |
| Содержание работ | Затраты рабочего времени, дней |
| 1.Построение диаграмм для проектирование дипломного проекта | 4 дня |
| 2.Поиск необходимых функций платформы | 5 дней |
| 3.Разработка серверной части | 6 дней |
| 4.Разработка клиентской части | 7 дней |
| 6.Тестирование внешнего вида | 2 дня |
| 6.Тестирование бизнес-логики | 3 дня |
| 7.Написание руководства пользователя | 2 дня |
| Всего | 29 |

Результат (значение «всего = 29 дня») по данной таблице будет использовано далее для расчётов.

**7.4 Основная заработная плата**

Для определения величины основной заработной платы, было проведено исследование величин заработных плат для junior full stack разработчиков. В итоге было установлено, что средняя месячная заработная плата на позиции junior full stack разработчика составляет 1 925 рублей.

Согласно таблице 7.2, проект разрабатывался одним человеком на протяжении 29 рабочих дня, что соответствует 1,5 месяца. Таким образом, основная заработная плата будет рассчитываться по формуле 7.3:

|  |  |  |
| --- | --- | --- |
|  | , | (7.3) |

где Соз– основная заработная плата, руб.;

Траз – время раработки, месяцев;

Краз – количество разработчиков, человек;

Сзп – средняя месячная заработная плата.

|  |  |  |
| --- | --- | --- |
|  | руб. |  |

В дальнейшем для других расчётов используется основная заработная плата, рассчитанная по указанной выше методике.

**7.5 Дополнительная заработная плата**

Дополнительная заработная плата на конкретное программное средство включает выплаты, предусмотренные законодательством о труде, и определяется по нормативу в процентах к основной заработной плате по формуле (7.4):

|  |  |  |
| --- | --- | --- |
|  | , | (7.4) |

где Соз– основная заработная плата, руб.;

Ндз – норматив дополнительной заработной платы, %.

руб.

**7.****6 Отчисления в Фонд социальной защиты населения**

Отчисления в Фонд социальной защиты населения (ФСЗН) и по обязательному страхованию от несчастных случаем на производстве и профессиональных заболеваний определяются в соответствии с действующими законодательными актами по нормативу в процентном отношении к фонду основной и дополнительной зарплаты исполнителей.

Отчисления в Фонд социальной защиты населения вычисляются по формуле 7.5

|  |  |  |
| --- | --- | --- |
|  |  | (7.5) |

где – основная заработная плата, руб.;

– дополнительная заработная плата на конкретное ПС, руб.;

– норматив отчислений в Фонд социальной защиты населения, %.

Отчисления в БРУСП «Белгосстрах» вычисляются по формуле 7.6

|  |  |  |
| --- | --- | --- |
|  |  | (7.6) |

Таким образом, общие отчисления в БРУСП «Белгосстрах» составили 19,9 руб., а в фонд социальной защиты населения – 1 128,8 руб.

**7.7 Расходы на материалы**

Сумма расходов на материалы СМ определяется как итог таблицы 7.3, в которой собраны реально понесённые затраты на разработку программного средства.

Таблица 7.3 – Материальные затраты на разработку ПС

|  |  |
| --- | --- |
| Вид материальных затрат | Фактические затраты, руб. |
| 1. Бумага, 0,2 пачки | 2,64 |
| 2. Маркер для доски, 1 шт. | 2,6 |
| Всего | 5,24 |

Сумма расходов на материалы составляет 5,24 руб.

**7.8** **Расходы на специальное оборудование и платные услуги**

Сумма расходов на специальное оборудование и платные услуги ССОПУ определяется как итог таблицы 7.4. Для реализации данного проекта была использована подписка платформы чат-ботов POE AI Creative Cloud компании Quora по тарифу 19,99 USD в месяц, что по курсу 3,22 рублей за USD составляет 64,38 рубля в месяц. Специального оборудования для разработки рассматриваемого программного средства приобретено не было.

Таблица 7.4 – Стоимость оборудования, специальных программ и платных услуг для разработки ПС

|  |  |
| --- | --- |
| Вид затрат | Фактические затраты, руб. с НДС |
| 1. Подписка приложений платформы чат-обтов, 1 месяц | 64,38 |
| Всего | 64,38 |

Расходы на специальное оборудование и платные услуги, необходимые для разработки и отладки разрабатываемого программного средства,равны 64,38 рублей.

**7.9 Прочие прямые затраты**

Сумма прочих затрат Спз определяется как произведение основной заработной платы исполнителей на конкретное программное средство Соз на норматив прочих затрат в целом по организации Нпз, и находится по формуле 7.9.

|  |  |  |
| --- | --- | --- |
|  |  | (7.9) |

Все данные необходимые для вычисления есть, поэтому можно определить сумму прочих затрат.

Спз = 2 887 ⋅ 18,5 / 100 = 534,09 руб.

В итоге мы получили прочих прямых затрат на сумму 534,09 руб.

**7.10** **Накладные расходы**

Сумма накладныхрасходовСобп,обх – произведение основной заработной платы исполнителей на конкретное программное средство Соз на норматив накладных расходов в целом по организации Нобп,обх, по формуле 7.10.

|  |  |  |
| --- | --- | --- |
|  |  | (7.10) |

Все данные необходимые для вычисления есть, поэтому можно определить сумму накладных расходов.

= 2 887⋅ 10 / 100 = 288,7 руб.

В итоге накладных расходов оказалось на сумму 288,7 руб.

**7.11** **Сумма расходов на разработку программного средства**

Сумма расходов на разработку программного средства Ср определяется как сумма основной и дополнительной заработных плат исполнителей на конкретное программное средство, отчислений на социальные нужды, расходов на материалы, расходов на оплату машинного времени, суммы прочих затрат и суммы накладных расходов, по формуле 7.11:

|  |  |  |
| --- | --- | --- |
|  | Ср = Соз + Сдз + Сфсзн + Сбгс + См + ССОПУ + Спз + Собп,обх. | (7.11) |

Все данные необходимые для вычисления есть, поэтому можно определить сумму расходов на разработку программного средства.

Ср = 2 887 + + + 19,9 + 5,24 + 64,38 + 534,09 + 288,7 =

= 5 361,16 руб.

Сумма расходов на разработку программного средства была вычислена на основе данных, рассчитанных ранее в данном разделе, и составила 5 361,16 рублей.

**7.12** **Расходы на сопровождение и адаптацию**

Сумма расходов на сопровождение и адаптацию программного средства Срса определяется как произведение суммы расходов на разработки на норматив расходов на сопровождение и адаптацию Нрса, и находится по формуле 7.12.

|  |  |  |
| --- | --- | --- |
|  |  | (7.12) |

= 5 361,16 ⋅ 17 / 100 = 911,39 руб.

Сумма расходов на сопровождение и адаптацию была вычислена на основе данных, рассчитанных ранее в данном разделе.

Все проведенные выше расчеты необходимы для вычисления полной себестоимости проекта.

**7.13 Полная себестоимость**

Полная себестоимость Сп определяется как сумма двух элементов: суммы расходов на разработку Ср и суммы расходов на сопровождение и адаптацию программного средства Срса.

Полная себестоимость Сп вычисляется по формуле 7.13:

|  |  |  |
| --- | --- | --- |
|  |  | (7.13) |

Сп = 5 361,16 + 911,39 = 6 272,55 руб.

Полная себестоимость программного средства была вычислена на основе данных, рассчитанных ранее в данном разделе.

**7.14 Определение цены, оценка эффективности**

Так как программное обеспечение разрабатывается для потребления другими лицами, а выбранный способ монетизации – расширенная подписка, рассмотрим продукты-аналоги:

1. Trello – облачная программа для управления проектами небольших групп, разработанная Fog Creek Software. озволяет создавать карточки, которые представляют из себя задачу, изменять статус задачу: задача может находиться в одном из трех состояний: «Сделать», «В работе», «Готово». Карточки, в зависимости от состояния, распределяются в три колонки. Trello позволяет создавать множество проектов, называемых «досками». Каждая из досок также имеет свое состояние. Также Trello поддерживает функцию «Чек -листы». По сути, данная функция позволяет разбивать задачи на подзадачи для более удобного оформления задачи
2. Todoist – функциональный таск менеджер, который позволяет удобно работать над задачами и отслеживать прогресс. Приложение обладает следующими преимуществами: простой и интуитивный интерфейс, кроссплатформенность, возможность быстрого добавления задач и поддержка общих проектов для совместной работы. Также приложение обладает функцией синхронизации между устройствами, что позволяет получить доступ к приложению в любом месте.
3. Приложение TickTick в первую очередь полезно для индивидуальных пользователей и для отслеживания повседневных задач, а для команд, работающих совместно это приложение может быть недостаточно функциональным. Также, как и Todoist обладает интуитивным интерфейсом и кроссплатформенностью, помимо этого предлагает пользователю интеллектуальные напоминания, приоритеты задач и promodo-таймер

По результатам анализа применяемых продуктами-аналогами стратегий монетизации следует выбрать стратегию монетизации: продажа расширенной подписки на создаваемый нами продукт.

Были выбраны следующие характеристики для показателей качества рассматриваемого программного продукта и программного продукта конкурента:

1. Дизайн – внешний вид приложения, который является первым на что смотрит пользователь при выборе продукта.
2. Юзабилити – показатель, который отражает то, насколько удобно пользователю работать с приложением.
3. Функционал – количество функций и инноваций, внедряемых в приложение
4. Отказоустойчивость – способность приложения работать при возникновении непредвиденных ошибок и багов.

Расчет показателей качества базового и нового продуктов, согласно балловому методу, приводится в таблицах 7.5 и 7.6.

Таблица 7.5 – Содержание и объем функций в программном средстве

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Показатель качества | Весовой коэффициент | Trello | Todoist | TickTick | Разработанное приложение |
| Дизайн | 0,3 | 8 | 7 | 7 | 8 |
| Юзабилити | 0,3 | 8 | 7 | 6 | 7 |
| Функционал | 0,3 | 9 | 8 | 5 | 7 |
| Отказоустойчивость | 0,1 | 7 | 6 | 7 | 7 |
| *Всего* | *1* | *8,2* | *7,2* | *6,1* | *7,3* |

Расчёт прогнозного количества установок программного средства К1 при монетизации методом подписок, рассчитывается по формуле 7.14:

|  |  |
| --- | --- |
|  | (7.14) |

где К0 − количество установок ПС конкурента;

T0 – количество лет существования приложения;

ИР – показатель рассматриваемого программного продукта;

ИК – показатель программного продукта конкурента.

К1 = (800 000 / 13 \* 7,3) / 8,2 = 54 784 (установок в год),

К2 = (650 000 / 17 \* 7,3) / 7,2 = 38 766 (установок в год),

К3 = (300 000/ 13 \* 7,3) / 6,1 = 27 616 (установок в год),

К = (54 784 + 38 766 + 27 616) / 3 = 40 388 (установок в год).

Определение цены подписки нового продукта Ц1 осуществляется по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (7.15) |

где Ц0 − цена программного продукта конкурента,

ИР – показатель рассматриваемого программного продукта,

ИК – показатель программного продукта конкурента.

Ц1 = (193,26 7,3) / 8,2 = 172,1 рублей

Ц2 = (154,61 7,3) / 7,2 = 156,7 рублей

Ц3 = (115,95 7,3) / 6,1 = 138,7 рублей

Ц = (172,1 + 156,7 + 115,95) / 3 = 155,8 рублей

По данным расчета рыночной цены рассмотренных конкурентов такого решения, если среднее количество установок 311 553 за год, и при этом расширенную подписку приобретают приблизительно 3 процента от общих пользователей (согласно данным по приложению Todoist) то денежные поступления от продажи расширенной подписки Пост.в год = 188 773,5 рублей за год.

Количество покупателей продукта необходимых для окупаемости расширения Пп вычисляется по формуле 7.16:

|  |  |  |
| --- | --- | --- |
|  |  | (7.16) |

где – полная себестоимость, руб.;

Пост.в год – денежные поступления от продажи подписки расширения за год, руб.;

Ток = 6 272,55 / 155,8 = 41 покупатель.

**7.15 Вывод по разделу**

В таблице 7.6 представлены результаты расчётов для основных показателей данной главы в краткой форме.

Таблица 7.6 – Результаты расчетов

|  |  |
| --- | --- |
| Наименование показателя | Значение |
| Время разработки, мес. | 1,5 |
| Количество программистов, чел. | 1 |
| Основная заработная плата, руб. | 2 887 |
| Дополнительная заработная плата, руб |  |
| Отчисления в Фонд социальной защиты населения и по обязательному страхованию от несчастных случаев на производстве и профессиональных заболеваний, руб | 1 148,7 |
| Расходы на материалы, оплату машинного времени, прочие, руб | 603,71 |
| Накладные расходы, руб | 288,7 |
| Себестоимость разработки программного средства, руб. | 5 361, 16 |
| Расходы на сопровождение и адаптацию, руб. | 911,39 |
| Полная себестоимость, руб. | 6 272,55 |
| Годовые денежные поступления от продажи подписки, руб. | 188 773,5 |
| Количество покупателей для окупаемости | 41 |

Разработка программного средства, осуществляемая одним программистом в течении 1,5 месяца, при заданных условиях обойдется в 6 272,55 руб. Реализации данного программного средства будет приносить годовые денежные поступления от продажи подписки в размере 174 428,4 рублей и окупиться при покупке 41 подписки на год.

При количестве установок в год равном 40 388 единице и с учетом того факта, что подписку оплачивает примерно 3 процента от общего числа пользователей, что равняется 1 211 подписчикам, проект успешно окупится и начнет приносить прибыль в течение первого года.

Заключение

Листов

*ФИО*

*Тихон А.А.*

*Бурмакова А.В.*

*Блинова Е.А.*

*Подп.*

*Дата*

Разраб.

Пров.

Утв.

Лит.

Лист

*БГТУ 01.00.ПЗ*

*74201055, 2024*

*Заключение*

Н. контр.

*Николайчук А.Н.*

1

1

Целью дипломного проекта была разработка мобильного приложения, предоставляющего следующие задачи:

– регистрация/авторизация пользователей;

– создание проекта;

– создание подзадач проекта;

– добавление участников в проект;

– изменение прогресса выполнения проекта;

– удаление участников из проекта.

Также необходимо было разработать веб приложение, сервер, локальную и глобальную базы данных.

В ходе выполнения поставленных задач были разработаны макеты мобильного и веб приложений, созданы диаграммы вариантов действий.

При проектировании баз данных были разработаны необходимые таблицы, выбраны первичные и вторичные ключи для каждой из таблиц, использован подход Soft Delete.

После завершения этапа проектирования были выбраны технические для реализации поставленных задач. Для разработки глобальной базы данных была выбрана СУБД Microsoft SQL Server. В качестве средств разработки использовались Android Studio и Visual Studio Code. Для написания реализации использовались языки программирования Dart в совокупности с фреймворком Flutter и Node.js в совокупности с фреймворком Express.

Для обеспечения безопасности личных данных пользователя используется шифрования с помощью библиотеки crypto/

Мобильное приложение, кроме вышеперечисленных функций, также обладает возможностью синхронизации.

В результате выполнения задач было разработано мобильное приложение, обеспечивающее следующие функции:

– регистрация/авторизация пользователей;

– создание проекта;

– создание подзадач проекта;

– добавление участников в проект;

– изменение прогресса выполнения проекта.

Список использованных источников

Листов

*ФИО*

*Тихон А.А.*

*Бурмакова А.В.*

*Блинова Е.А.*

*Подп.*

*Дата*

Разраб.

Пров.

Утв.

Лит.

Лист

*БГТУ 00.00.ПЗ*

*74201055, 2024*

*Список использованных*

*источников*

Н. контр.

*Николайчук А.Н.*

*Николайчук А.Н.*

1

1

1. Syncflusion\_flutter\_calendar [Электронный ресурс] ­– Режим доступа: https://pub.dev/packages/syncflusion\_flutter\_calendar. – Дата доступа: 10.03.2023.
2. WebSocket: разбираем как работает [Электронный ресурс] ­– Режим доступа: https://habr.com/ru/sandbox/171066/. – Дата доступа: 30.02.2023.
3. Widget Catalog [Электронный ресурс] – Режим доступа: https://docs.flutter.dev/ui/widgets. – Дата доступа: 26.02.2023.
4. Fetch data from the internet [Электронный ресурс] – Режим доступа: https://docs.flutter.dev/cookbook/networking/fetch-data. – Дата доступа: 29.02.2023.
5. Руководство по SQLite [Электронный ресурс] – Режим доступа: https://metanit.com/sql/sqlite. – Дата доступа: 26.02.2023.
6. Синхронизация различных хранилищ данных [Электронный ресурс] – Режим доступа: https://habr.com/ru/companies/semrush/articles/665618/. – Дата доступа – 26.06.2024.
7. Expansion tile in flutter [Электронный ресурс] – Режим доступа: https://aakashpp.medium.com/expansion-tile-in-flutter-7ba50c7386ab. Дата доступа – 12.04.2024.

# ПРИЛОЖЕНИЕ А

Логическая схема глобальной базы данных

# ПРИЛОЖЕНИЕ Б

Логическая схема локальной базы данных

# ПРИЛОЖЕНИЕ В

Архитектура приложения

# ПРИЛОЖЕНИЕ Г

Блок-схема алгоритма выгрузки задач и подзадач