**ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ТОМСКИЙ ЭКОНОМИКО-ПРОМЫШЛЕННЫЙ КОЛЛЕДЖ»**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ API СЕРВИСА**

***ОТЧЁТ ПО УЧЕБНОЙ ПРАКТИКЕ***

*СПЕЦИАЛЬНОСТЬ 09.02.07 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ*

Выполнил:

Студент группы 1912с

\_\_\_\_\_\_\_\_\_\_\_\_ / Евсеев Д.В.

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

Проверил:

Руководитель УП

\_\_\_\_\_\_\_\_\_\_\_\_ / Уляхин В.А.

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

Томск 2024

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc164389549)

[**Информация о проекте и команде разработчиков** 3](#_Toc164389550)

[**Назначение и структура документа** 3](#_Toc164389551)

[**Глоссарий** 3](#_Toc164389552)

[**ОБЩЕЕ ОПИСАНИЕ** 4](#_Toc164389553)

[**Общий взгляд на предлагаемое решение** 4](#_Toc164389554)

[**Ролевая модель** 4](#_Toc164389555)

[**ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ** 6](#_Toc164389556)

[**Концептуальная модель данных** 6](#_Toc164389557)

[**Логическая модель данных** 6](#_Toc164389558)

[**Сущности** 6](#_Toc164389559)

[**Описание API** 8](#_Toc164389560)

[***Формат ответа общих требований*** 8](#_Toc164389561)

[***Авторизация пользователя*** 8](#_Toc164389562)

[**ТЕСТИРОВАНИЕ** 10](#_Toc164389563)

[**Авторизация пользователя** 10](#_Toc164389564)

[**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ** 11](#_Toc164389565)

# **ВВЕДЕНИЕ**

## **Информация о проекте и команде разработчиков**

|  |  |
| --- | --- |
| Название проекта |  |
| Цель проекта |  |
| Задачи проекта | 1. Разработать технического задания API сервиса. 2. Разработать технического задания веб-сайта. 3. Разработать технического задания десктоп приложения. 4. Разработать API сервиса на фреймворке Laravel 10. 5. Разработать веб-сайта на фреймворке Vue 3. 6. Разработать десктоп приложения на C# WPF. 7. Разработать автоматизированные тесты и провести тестирование API сервиса. 8. Разработать автоматизированные тесты и провести тестирование веб-сайта. 9. Разработать автоматизированные тесты и провести тестирование десктоп приложения. |
| Название команды | Команда «Анлакеры» |
| Состав команды и роли | Заикин Вадим Анатольевич – Аnalyst, Backend developer, QA.  Шейкина Елизавета Викторовна – Аnalyst, Frontend developer, QA.  Гусейнов Мурад Шахин Оглы – Аnalyst, C# developer, QA. |

## **Назначение и структура документа**

Настоящий документ рассчитан на использование в проектной команде в качестве ключевого документа, определяющего требование по реализации задачи по разработке API сервиса для проекта «…».

## **Глоссарий**

|  |  |
| --- | --- |
| **Понятие** | **Определение понятия** |
| API | API (интерфейс программирования приложений) – это набор определенных правил и соглашений, который используется для взаимодействия между различными программными компонентами. API определяет, как различные части программного обеспечения могут взаимодействовать друг с другом, обычно путем обмена данными и выполнения определенных операций. |
| Postman | Postman – это инструмент для тестирования и разработки API. Он предоставляет удобный интерфейс для отправки HTTP запросов к API, а также для анализа ответов. |
|  |  |
|  |  |
|  |  |

# **ОБЩЕЕ ОПИСАНИЕ**

Общее описание проекта

Все пользователи системы подразделяются на четыре группы:

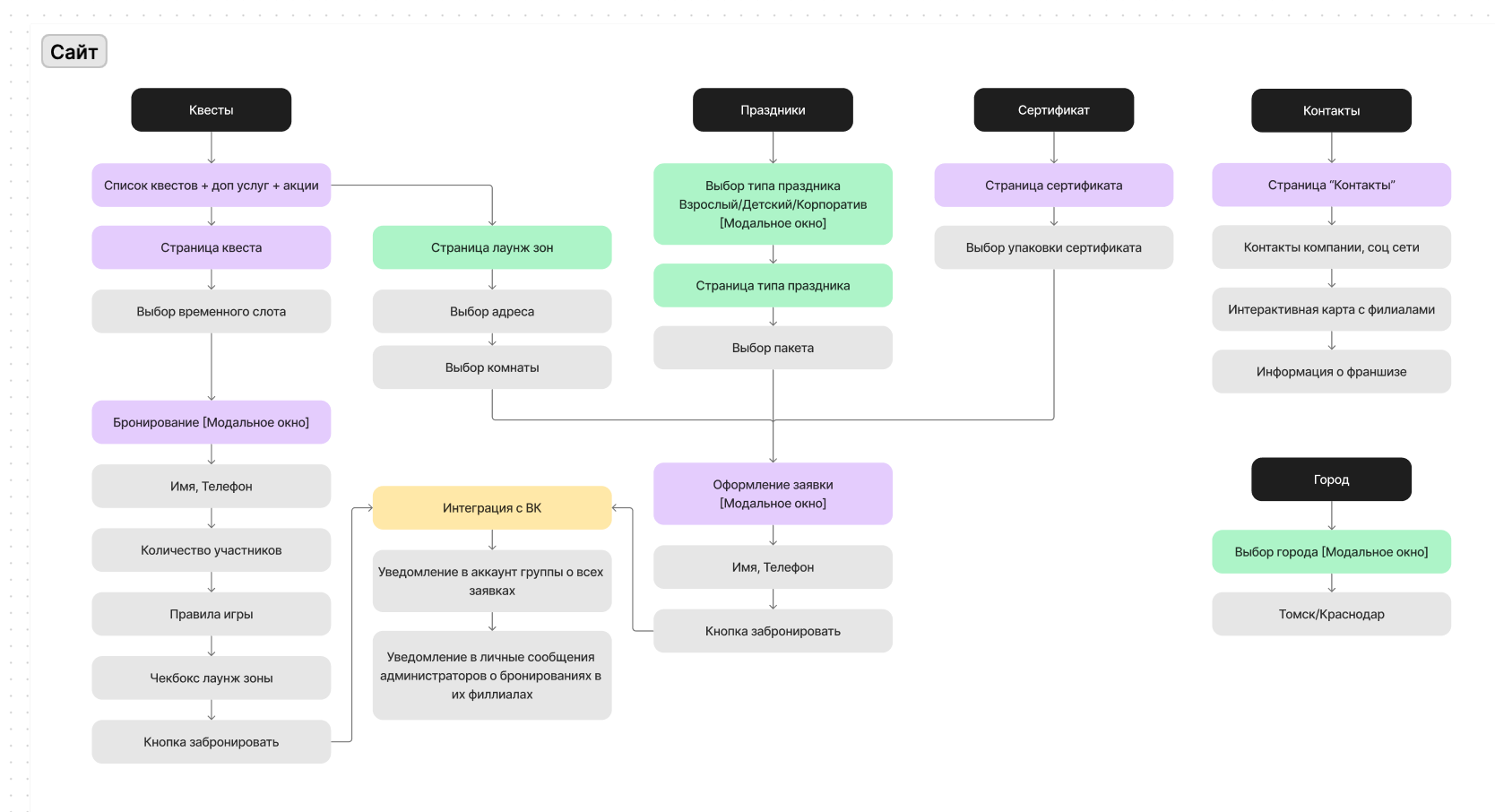
1. Администраторы;

2. Менеджеры;

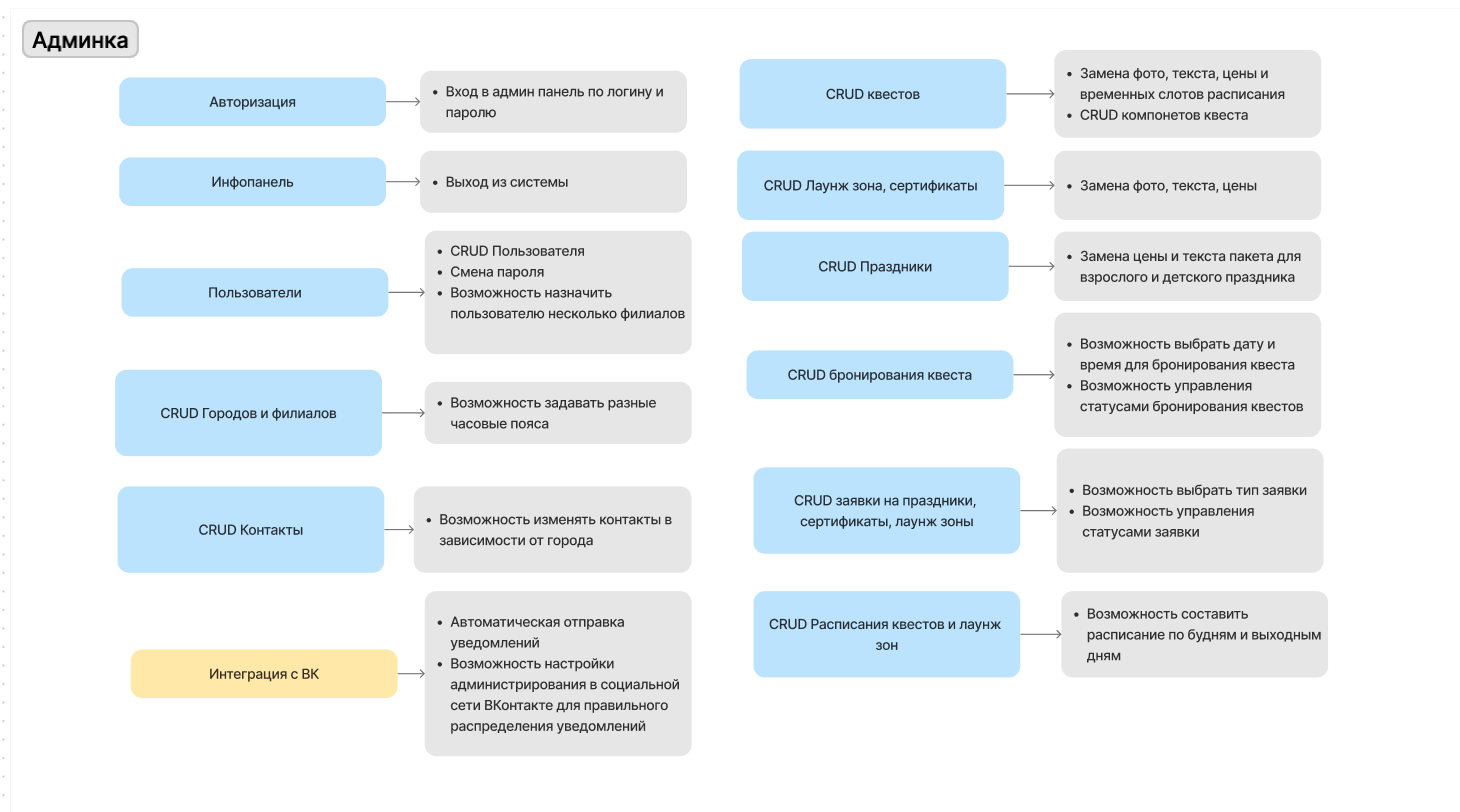
3. Сотрудник.

4. Клиент

## **Общий взгляд на предлагаемое решение**



(вставить свою концепцию)



(вставить свою концепцию)

## **Ролевая модель**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Раздел** | **Фичи** | **Админ** | **Оператор** | **Клиент** |
| Инфопанель | Просмотр инфопанели | да | да | да |
| Пользователи | Просмотр списка пользователей | да | да | да (видит только себя) |
| Создание нового пользователя | да | да | да (только себя) |
| Редактирование пользователя | да | да | да (только себя) |
| Изменение пароля пользователя | да | да (только себя) | да (только себя) |
| Удаление пользователя | да | нет | нет |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# **ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ**

## **Концептуальная модель данных**

Описание концептуальной модели:

1. Пользователи: содержит информацию о пользователях, которые имеют доступ к системе внутри конкретного филиала компании. Может содержать поля, такие как имя, фамилия, логин, пароль.
2. Филиалы: содержит информацию о различных филиалах компании в разных городах. Может содержать поля, такие как название филиала, адрес и другую дополнительную информацию.

(вставить сюда скриншот концептуальной модели)

## **Логическая модель данных**

(вставить сюда скриншот концептуальной модели)

## **Сущности**

**roles (роли)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK  Заполняется авто инкрементированием | Идентификатор роли |
| name | varchar (255) | Обязательное | Имя роли |
| code | varchar (255) | Обязательное  Unique | Код роли |

**users (пользователи)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK  Заполняется авто инкрементированием | Идентификатор пользователя |
| surname | varchar (255) | Обязательное | Фамилия пользователя |
| name | varchar (255) | Обязательное | Имя пользователя |
| login | varchar (255) | Обязательное  Unique | Логин пользователя |
| password | varchar (255) | Обязательное | Пароль пользователя |
| remember\_token | varchar (100) | Не обязательное | Токен авторизованного пользователя |
| created\_at | timestamp(0) | Не обязательное | Дата создания |
| updated\_at | timestamp(0) | Не обязательное | Дата обновления |
| role\_id | biginteger | FK  Обязательное | Роль пользователя |

Отзывы

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK  Заполняется авто инкрементированием | Идентификатор роли |
| name | varchar (255) | Обязательное | Имя роли |
| code | varchar (255) | Обязательное  Unique | Код роли |

## **Описание API**

### ***Формат ответа общих требований***

1. Попытка доступа гостя к защищенным авторизацией функциям системы.

Status: 401

Content-Type: application/json

Body:

{  
 "message": "Ошибка входа"  
}

1. Попытка доступа авторизованного пользователя к функциям недоступным для него.

Status: 403

Content-Type: application/json

Body:

{  
 "message": "Доступ запрещён"  
}

1. Попытка получить не существующий ресурс.

Status: 403

Content-Type: application/json

Body:

{  
 "message": "Не найдено"  
}

1. В случае ошибок связанных с валидацией данных.

Status: 422

Content-Type: application/json

Body:

{  
 "message": "Ошибка валидации данных",  
 "errors": {  
 "key1": "error message 1",  
 ...  
 "keyN": "error message N"  
 }  
}

### ***Авторизация пользователя***

1. Описание ресурса: Метод позволяет получить токен авторизированного пользователя.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/login
4. Заголовки запроса:
   * Content-Type: application/json
5. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| login | string | Логин пользователя | От 3 до 255 символов | Да |
| password | string | Пароль пользователя | От 6 до 255 символов | Да |

1. Тело запроса:

{  
 "login": "evseev-dv",  
 "password": "QWEasd123"  
}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

{  
 "token": "random\_srting"  
}

1. Возможные статусы: 200, 422

# **ТЕСТИРОВАНИЕ**

## **Авторизация пользователя**

# **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**