Петрозаводский государственный университет

Институт математики и информационных технологий

**Техническое задание**

**на разработку веб-сервиса для коллективных переводов**

**«Desman Translate»**

Заказчик

М. А. Колчин 22306

Исполнитель

С. Э. Зименкова 22305

# Термины, используемые в техническом задании

API – сервер с набором функций и описанием их применения.

Библиотека Bootstrap — модуль, являющийся дополнением функционала языка разметки HTML.

БД — сокращение для термина «база данных».

Библиотека React.js – модуль, являющийся обширным дополнением функционала языка JavaScript.

# Общие сведения

* 1. Название сайта – «Desman Translate».
  2. Основания для проведения работ

Заявка от заказчика, в которой описаны необходимый функционал и требования к дизайну.

* 1. Сведения о заказчиках и разработчиках

Заказчик, Колчин М. А., заинтересован в создании веб-сервиса для коллективных переводов, так как существующие аналоги являются сильно устаревшими более не поддерживающимися сервисами или корпоративными платными решениями, а также обладает готовым API для создания веб-сайта.

Разработчик, Зименкова С. Э., имеет опыт в разработке дизайна веб-страниц, верстке с помощью Bootstrap. Также заинтересована в создании подобного веб-сервиса и приобретении опыта создания веб-сайтов.

* 1. Плановые сроки начала – окончания работ

Начало – заказ на разработку принят 12 сентября 2023 года, с того же момента началась разработка сайта. Срок окончания работ – 24 декабря 2023 года.

* 1. Источники финансирования

Отсутствуют.

* 1. Порядок оформления и предъявления заказчику результатов работ.

До 12 октября – предоставление заказчику документа проектирования.

До 17 ноября – предоставлены эскизы страниц и архитектура сайта.

До 24 декабря – предоставлена готовая версия сайта.

# Назначения и цели создания системы

* 1. Назначение системы, цель создания системы

Цель: предоставить командам переводчиков удобный инструмент для асинхронной работы над переводом текстов.

* 1. Основные задачи, решаемые при помощи сайта

В настоящий момент существующие веб-сайты для коллективных переводов не подходят для удобного и качественного перевода любительскими командами: существующие бесплатные сервисы или не имеют необходимых функций, или слишком устарели для использования, а актуальные удобные сервисы представляют собой корпоративные решения для крупных организаций, в силу чего лицензия на их использования имеет достаточно высокую цену. Создание этого веб-сервиса позволит командам переводчиков бесплатно и удобно совместно работать над переводами книг, субтитров, ПО и других текстов.

Функциональные возможности сервиса:

* + Регистрация на сайте, настройка аккаунта;
  + Создание проектов (разделов сайта, хранящих информацию и файлы отдельного произведения);
  + Приглашение пользователей в проекты, распределение ролей между участниками проекта, модерация своих проектов;
  + Загрузка текста в проект, экспорт переведенного варианта;
  + Работа над переводом текста в специальном инструменте для переводчика:
    - разбиение текста на отрывки;
    - добавление своего варианта перевода отрывка текста;
    - голосование за лучший вариант перевода отрывка;
    - настройка экспорта переведенного текста.
  + Поиск существующих проектов (в т. ч. с фильтрами);
  + Настройка собственных проектов (в т. ч. возможность сделать проект публичным или закрытым, настроить доступ пользователей к нему);
  + Распределение ролей между участниками проекта;
  + Инструмент для перевода текста;
  + Инструмент для разбиения текста на отрывки;
  + Импорт и экспорт текста произведения.
  1. Целевая аудитория

Переводчики-любители и профессиональные команды переводчиков.

* 1. Внутренние и внешние интересы

Внутренние интересы заказчика – использование удобного бесплатного инструмента для коллективного асинхронного перевода текста и его дальнейшее развитие.

Внутренние интересы исполнителя – использование готового инструмента для собственных проектов по переводу, наработка опыта дизайна интерфейса, работы с библиотекой React.js.

Внешние интересы – получение удобного в использовании веб-сайта для коллективных переводов.

* 1. Показатели (индикаторы) назначения – целевые показатели (посещаемость, доходность)

Сайт не планируется использоваться для получения дохода. Для достижения целевых показателей веб-сервиса (при которых переводчики-любители могут опубликовать проект и найти себе команду волонтеров) веб-сервис должен иметь достаточно большой охват в тысячи пользователей, однако это достижимо только с помощью активной рекламы. В настоящий момент сервис будет выполнять свои функции в полной мере, если будет иметь приблизительно 50 активных пользователей.

# Характеристика объектов автоматизации

Сервис является инструментом создания и редактирования, а не просмотра, поэтому многие процессы выполняются пользователем. Процессы, которые должны быть автоматизированы, включают в себя следующее:

1. Разбиение текста на отрывки;
2. Экспортирование готового перевода в виде текстового файла.

# Требования к сайту, программному обеспечению

* 1. Требования к системе в целом (нефункциональные требования)
     1. Требования к архитектуре системы.

Сайт состоит из следующих страниц:

* Страница регистрации и входа;
* Главная страница с недавними проектами и панелью навигации;
* Набор страниц поиска проектов;
* Страница пользователя (в т. ч. настройки пользователя);
* Страница создания проекта;
* Страница проекта, в т. ч. страницы настройки проекта;
* Страница загрузки и экспорта глав;
* Страница с инструментом для перевода.
  + 1. К доступности и надежности (в т.ч. скорость отклика).
* Сервис должен быть достаточно стабильным, т. к. сбои в его работе помешают переводчикам работать над своими проектами;
* Дизайн должен быть удобным и минималистичным, чтобы не отвлекать переводчиков от работы;
* Сервис должен предоставлять возможность для переводчиков работать над переводом отрывков независимо друг от друга, поэтому добавление новых вариантов перевода не должно конфликтовать со старыми;
* Сервис должен предоставлять инструмент для синхронной работы над переводом непосредственно на самом сайте.  
  + 1. К численности и квалификации обслуживающего персонала (образование, квалификация)

Разработчик веб-сервиса должен иметь навыки работы с HTML, JavaScript, библиотеками React.js и React Bootstrap, опыт создания сайтов и графического дизайна.

* + 1. К эргономике и технической эстетике: оформлению (фирменный стиль, логотип, цвета-шрифты, примеры сайтов)

Примеры сайтов:

* Notabenoid.org <http://notabenoid.org>
* Crowdin <https://crowdin.com>

Интерфейс сайта должен быть как можно проще и эргономичнее. Приветствуется современный графический дизайн, присутствующий в библиотеке Bootstrap. Интерфейс должен быть светлым и иметь черный текст для удобного чтения. Приветствуется разработка фирменной цветовой палитры и логотипа сайта в дальнейшей разработке.

* + 1. К верстке

Верстка сайта обязана быть адаптивной. Сайт должен поддерживаться в самых популярных браузерах.

* + 1. К безопасности

Сервис должен надежно защищать скрытые проекты, т. к. они могут быть защищены авторским правом и не должны быть доступны для публики.

* + 1. К языкам и кодировкам

Сайт должен быть реализован с помощью HTML-элементов, которые создаются функциями, написанными с помощью библиотеки React.js на языке JavaScript. Применяется библиотека React Bootstrap.

* 1. Требования к функциям, выполняемым системой (функциональные требования)
     1. Типы пользователей / роли.

Существуют авторизированные и неавторизованные пользователи. Каждый авторизованный пользователь, являющийся участником некоторого проекта, имеет в этом проекте роль, независимую от его ролей в других проектах.

* + - 1. Права пользователей

Все пользователи:

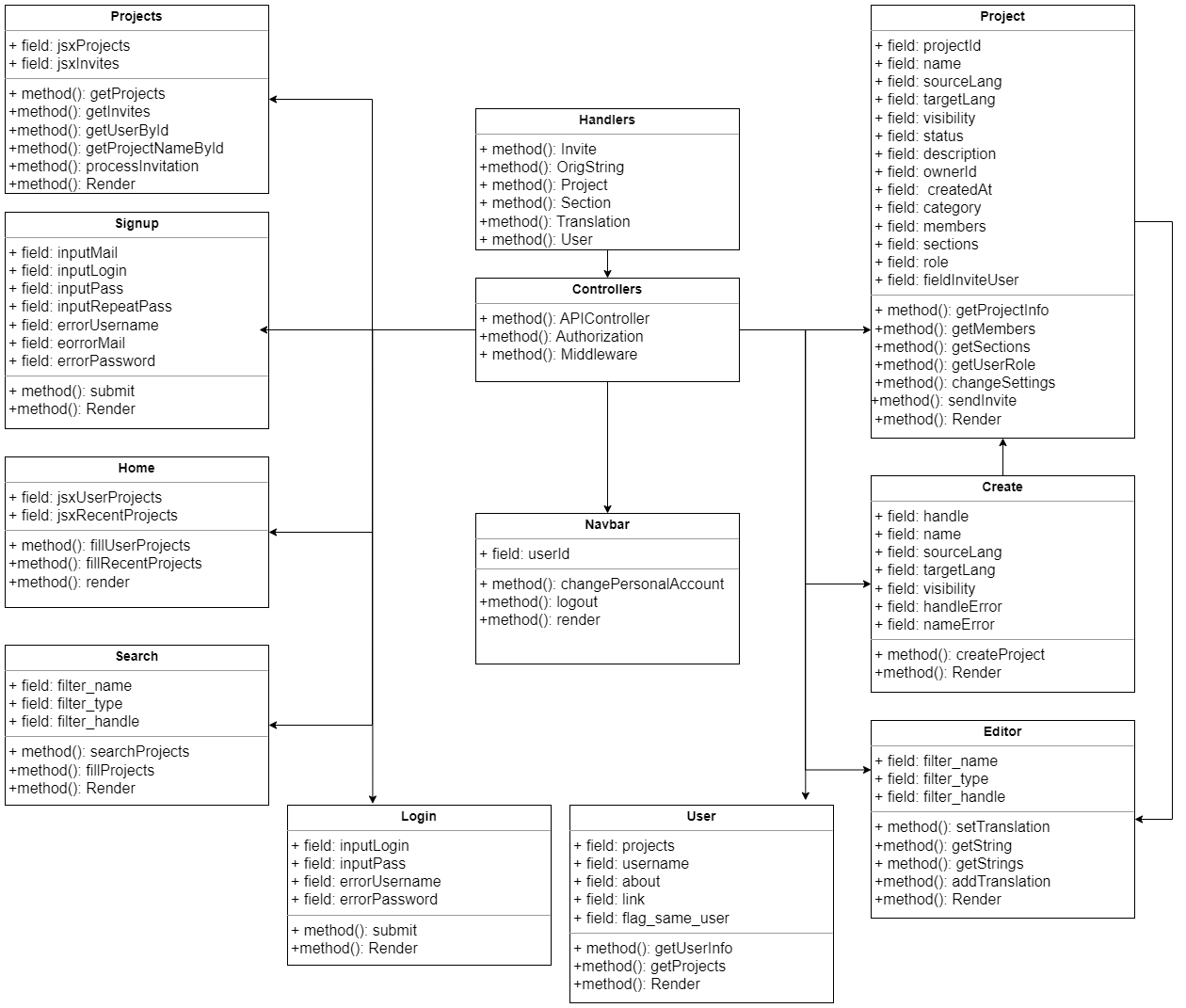
* Посетить главную страницу.
* Увидеть популярные проекты.

Неавторизованный пользователь:

* Пройти регистрацию в окне регистрации, чтобы создать новую учетную запись.
* Войти в существующую учетную запись.

Авторизованный пользователь:

* Выйти из учетной записи.
* Войти в личный кабинет:
* Перейти на окно смены пароля.
* Увидеть информацию о пользователе.
* Увидеть проекты, в которых он участвует.
* Создать новый проект.
* Зайти в один из проектов, в котором он участвует.
* Увидеть участников проекта.
* Увидеть процесс выполнения проекта.
* Увидеть информацию о проекте.
* Увидеть описание проекта.
* Выполнять действия в соответствии с ролью в проекте.
* Увидеть публичные проекты.
* Осуществить поиск по названию проекта.
* Написать разработчику.
  + 1. Перечень объектов предметной области и правила их взаимодействия – диаграмма классов



*Рисунок 2. Доступ и взаимодействие пользователя со страницами сайта*

* + 1. Перечень подсистем (частей, модулей) и описание их функциональных возможностей - на основе диаграмм использования. В разбивке по частям системы (внешняя часть, система администрирования, кабинеты…)
       1. Хэндлеры – API

#### Модуль Invite.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **static async add(invite)** | Добавляет приглашение в БД. Принимает на вход содержимое приглашения. |
| **static async find\_project\_invites(params)** | Возвращает список приглашений проекта, отфильтрованный на основе переданных параметров. Принимает на вход параметры запроса. |
| **static async find\_user\_invites(params)** | Возвращает список приглашений пользователя, отфильтрованный на основе переданных параметров. Принимает на вход параметры пользователя. |
| **static async find\_invite\_by\_id(params)** | Возвращает приглашение из БД по id. Принимает на вход id пользователя. |
| **static async find\_invite\_by\_project\_and\_user(params)** | Возвращает приглашение из БД по id пользователя и id проекта. Принимает на вход Id пользователя и проекта. |
| **async delete()** | Удаляет приглашение |
| **static async delete\_by\_user(user\_id)** | Удаляет приглашения по id пользователя. Принимает на вход id пользователя. |
| **static async delete\_by\_inviter(inviter\_id)** | Удаляет приглашения по id приглашающего пользователя. Принимает на вход id приглашающего пользователя. |
| **static async delete\_by\_project(project\_id)** | Удаляет приглашения по id проекта. Принимает на вход id проекта. |
| **async accept(is\_accepted)** | Добавляет пользователя в БД участников проекта. Принимает на вход флаг статуса приглашения. |

#### Модуль OrigString.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **static async add(string)** | Добавляет в БД отрывок. Принимает на вход запись отрывка. |
| **async edit(edits)** | Добавляет вариант редактирования к отрывку. Принимает на вход изменения. |
| **static async find\_strings(params)** | Возвращает список отрывков по фильтру. Принимает на вход параметры фильтра. |
| **static async find\_string(params)** | Возвращает отрывок по фильтру. Принимает на вход параметры фильтра. |
| **async delete()** | Удаляет строку из БД. |
| **static async delete\_by\_section(section\_id)** | Удаляет все отрывки раздела. Принимает на вход id раздела. |

#### Модуль Project.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **static async is\_unique\_name(name, project\_id)** | Возвращает, является ли имя проекта уникальным. Принимает на вход имя проекта и id проекта. |
| **static async is\_unique\_identificator (identificator, project\_id)** | Возвращает, является ли хэндл уникальным. Принимает на вход хэндл и id проекта. |
| **static async add(project)** | Добавляет проект в БД. Принимает на вход запись проекта. |
| **async edit(edits)** | Вносит изменения в проект. Принимает на вход изменения. |
| **static async find\_project(params)** | Возвращает проект по фильтру. Принимает на вход параметры поиска проекта. |
| **static async find\_projects(params)** | Возвращает список проектов, отфильтрованный на основе переданных параметров. Принимает на вход параметры поиска проектов. |
| **async is\_user\_member(user)** | Возвращает, является ли пользователь участником проекта. Принимает на вход данные пользователя. |
| **async user\_can\_see(user)** | Возвращает, может ли пользователь видеть проект. Принимает на вход данные пользователя. |
| **async user\_has\_access(rule, user)** | Возвращает, имеет ли пользователь роль. Принимает на вход роль и данные пользователя. |
| **async delete()** | Удаляет проект из БД. |

#### Модуль Section.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **static async is\_unique\_name(name, section\_id, project\_id)** | Возвращает, является ли имя раздела уникальным. Принимает на вход имя и id раздела |
| **static async add(section)** | Добавляет раздел в БД. Принимает на вход запись раздела. |
| **async edit(edits)** | Вносит изменения в раздел. Принимает на вход данные изменения. |
| **static async find\_section(params)** | Возвращает раздел по фильтру. Принимает на вход параметры поиска. |
| **static async find\_sections(params)** | Возвращает список разделов по фильтру. Принимает на вход параметры поиска. |
| **async delete()** | Удаляет раздел из БД. |
| **static async delete\_by\_project(project\_id)** | Удаляет разделы по id проекта. Принимает на вход id проекта. |

#### Модуль Translation.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **static async add(translation)** | Добавляет вариант перевода отрывка. Принимает на вход объект - вариант перевода. |
| **async edit(edits)** | Вносит в БД изменение варианта перевода. Принимает на вход объект - изменения перевода. |
| **static async get\_max\_length(translation)** | Возвращает максимальную длину отрывка. Принимает на вход объект - исходный отрывок. |
| **static async find\_translations(params)** | Возвращает список вариантов перевода для одного отрывка. На вход принимает параметры запроса. |
| **static async find\_translation(params)** | Возвращает вариант перевода отрывка. На вход принимает ID варианта перевода. |
| **async delete()** | Удаляет вариант перевода. |
| **static async delete\_by\_string(string\_id)** | Удаляет все варианты перевода отрывка. Принимает ID удаляемого отрывка. |

#### Модуль User.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **static async is\_unique\_username(username, user\_id)** | Проверяет хэндл пользователя на уникальность. Принимает на вход хэндл и ID пользователя. |
| **static async add(user)** | Добавляет пользователя в БД. Принимает на вход объект - пользователь. |
| **async edit(edits)** | Вносит в БД изменения о пользователе. Принимает на вход объект - изменения. |
| **static async find\_user(params)** | Находит пользователя по введенным параметрам. Принимает на вход ID или хэндл. |
| **static async find\_users(params)** | Возвращает массив пользователей по введенным параметрам. Принимает на вход параметры запроса. |
| **async add\_project\_membership(project\_id)** | Добавляет пользователя в проект как участника. Принимает на вход ID проекта. |
| **async remove\_project\_membership(project\_id)** | Удаляет пользователя из проекта. Принимает на вход ID проекта. |
| **async delete()** | Удаляет пользователя. |

* + - 1. Контроллеры – API

#### Модуль APIController.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **async get\_users(req, res)** | Получает пользователей из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные о пользователях, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_user(req, res)** | Получает пользователя из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные о пользователе, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_user\_invites(req, res)** | Получает приглашения пользователя из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные о приглашениях пользователя, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_user\_invite(req, res)** | Получает приглашение пользователя из БД и возвращает 200 и данные о приглашении пользователя, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_project(req, res)** | Получает проект из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные о проекте, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_projects(req, res)** | Получает проекты из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные о проектах, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_project\_roles(req, res)** | Получает роли проекта из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные о ролях проекта, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_project\_role(req, res)** | Получает роль проекта из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные о роли проекта, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_project\_members(req, res)** | Получает участников проекта из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные об участниках проекта, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_project\_member(req, res)** | Получает участника проекта из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные об участнике проекта, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_project\_invites(req, res)** | Получает приглашения в проект из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные о приглашениях в проекте, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_project\_invite(req, res)** | Получает приглашение в проект из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные о приглашении в проекте, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_section(req, res)** | Получает раздел проекта из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные о разделе проекта, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_sections(req, res)** | Получает разделы проекта из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные о разделах проекта, возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_string(req, res)** | Получает отрывок из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные об отрывке, возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_strings(req, res)** | Получает отрывки из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные об отрывках, возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_translation(req, res)** | Получает перевод из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные о переводе, возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async get\_translations(req, res)** | Получает переводы из БД, фильтрует согласно параметрам запроса и возвращает 200 и данные о переводах, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async add\_project(req, res)** | Обрабатывает тело запроса с данными о проекте, добавляет проект в БД и возвращает 201 и данные о проекте, возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async add\_section(req, res)** | Обрабатывает тело запроса с данными о разделе, добавляет раздел проекта в БД и возвращает 201 и данные о разделе проекта, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async add\_strings(req, res)** | Обрабатывает тело запроса с данными об отрывка, добавляет отрывки в БД и возвращает 201 и данные об отрывках, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async add\_translation(req, res)** | Обрабатывает тело запроса с данными о переводе, добавляет перевод в БД, возвращает 201, данные о переводе; возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async add\_project\_role(req, res)** | Обрабатывает тело запроса с данными о роли проекта, добавляет роль проекта в БД и возвращает 201 и данные о роли проекта, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async invite\_to\_project(req, res)** | Обрабатывает тело запроса с данными о приглашении, создаёт приглашение в проект для пользователя и возвращает 201 и данные о приглашении, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async resolve\_invite(req, res)** | Разрешает (пользователь принимает или отвергает) приглашение и удаляет его из БД и возвращает 204, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async project\_member\_grant\_role(req, res)** | Обрабатывает тело запроса с данными о пользователе и присваиваемой роли, присваивает пользователю проекта роль и возвращает 200 и данные о участнике проекта, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async edit\_user(req, res)** | Обрабатывает тело запроса с данными о пользователе, изменяет пользователя в БД и возвращает 200 и данные о пользователе, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async edit\_project(req, res)** | Обрабатывает тело запроса с данными о проекте, изменяет проект в БД и возвращает 200 и данные о проекте, возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async edit\_project\_role(req, res)** | Обрабатывает тело запроса с данными о роли проекте, изменяет роль проекта в БД и возвращает 200 и данные о роли проекта, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async edit\_section(req, res)** | Обрабатывает тело запроса с данными о разделе проекте, изменяет раздел проекта в БД и возвращает 200 и данные о разделе проекта, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async edit\_string(req, res)** | Обрабатывает тело запроса с данными об отрывке, изменяет отрывок в БД и возвращает 200 и данные об отрывке, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async edit\_translation(req, res)** | Обрабатывает тело запроса с данными о переводе, изменяет перевод в БД и возвращает 200 и данные о переводе, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async delete\_project(req, res)** | Удаляет проект из БД и возвращает 204, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async delete\_section(req, res)** | Удаляет раздел проекта из БД и возвращает 204, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async delete\_string(req, res)** | Удаляет отрывок из БД и возвращает 204, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async delete\_translation(req, res)** | Удаляет перевод из БД и возвращает 204, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async delete\_project\_invite(req, res)** | Удаляет приглашение в проект из БД и возвращает 204, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async delete\_project\_role(req, res)** | Удаляет роль проекта из БД и возвращает 204, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async kick\_project\_member(req, res)** | Выгоняет участника из проекта и возвращает 204, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |

#### Модуль Authorization.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **async create\_user(req, res)** | Обрабатывает тело запроса с данными о пользователе, добавляет пользователя в БД и возвращает 201 и данные о пользователе, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async login\_user(req, res)** | Авторизует пользователя на основе полученных данных и возвращает 200 и данные о пользователе, или возвращает 4xx в случае ошибки. Принимает на вход параметры запроса. |
| **async authorization(req, res, next)** | Добавляет в параметры запроса данные о текущем пользователе (или отсутствии авторизации) и передаёт запрос дальше. Принимает на вход параметры запроса. |
| **async check\_authorization(req, res, next)** | Проверяет авторизацию пользователя и передаёт запрос дальше, или возвращает 401 в случае ошибки. Принимает на вход параметры запроса. |

#### Модуль Middleware.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **async get\_project(req, res, next)** | Добавляет в параметры запроса данные о проекте и передаёт запрос дальше, или возвращает 404 в случае несуществующего проекта. Принимает на вход параметры запроса. |
| **async get\_section(req, res, next)** | Добавляет в параметры запроса данные о разделе проекта и передаёт запрос дальше, или возвращает 404 в случае несуществующего раздела. Принимает на вход параметры запроса. |
| **async get\_string(req, res, next)** | Добавляет в параметры запроса данные о отрывке и передаёт запрос дальше, или возвращает 404 в случае несуществующего отрывка. Принимает на вход параметры запроса. |
| **async get\_translation(req, res, next)** | Добавляет в параметры запроса данные о варианте перевода и передаёт запрос дальше, или возвращает 404 в случае несуществующего перевода. Принимает на вход параметры запроса. |

* + - 1. Модуль клиентского интерфейса

#### Модуль Navbar.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **сhangePersonalAccount()** | Определяет, что необходимо отображать: кнопки авторизации или регистрации или личный кабинет и кнопку выхода. Изменяет значение переменной navbarValue. |
| **logout()** | Очищение cookie авторизованного пользователя - выход из аккаунта |
| **Render()** | Добавляет jsx-элемент шапки страницы на основной файл index.html Возвращает массив jsx-элементов. |

#### Модуль Home.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **fillUserProjects()** | Заполняет список проектов, в которых участвует пользователь. Устанавливает значение для переменной userProjects. |
| **fillRecentProjects()** | Заполняет список недавно созданных проектов. Устанавливает значение для переменной recentProjects. |
| **Render()** | Добавляет jsx-элемент главной страницы на основной файл index.html Возвращает массив jsx-элементов проектов. |

#### Модуль Projects.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **getProjects()** | Заполняет список проектов, в которых участвует пользователь. Устанавливает значение для переменной projects. |
| **getInvites()** | Заполняет список приглашений пользователя. Устанавливает значение для переменной invites. |
| **getUserById(id)** | Принимает на вход id пользователя. Возвращает логин пользователя. |
| **getProjectNameById(id)** | Принимает на вход id проекта. Возвращает название проекта. |
| **processInvitation(invite\_id, is\_accepted)** | Принимает на вход id приглашения и статус - принято оно или нет. Обрабатывает приглашение - добавляет пользователя в проект или не добавляет. |
| **Render()** | Добавляет jsx-элемент главной страницы на основной файл index.html Возвращает массив jsx-элементов проектов. |

#### Модуль Signup.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **submit()** | Вызывается по нажатию кнопки "Зарегистрироваться". Проверяет корректность полей и добавляет запись пользователя. |
| **Render()** | Добавляет jsx-элемент страницы регистрации на основной файл index.html Возвращает массив jsx-элементов проектов. |

#### Модуль Login.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **submit()** | Вызывается по нажатию кнопки "Авторизоваться". Проверяет корректность полей и авторизовывает пользователя - появляются куки-файлы. |
| **Render()** | Добавляет jsx-элемент страницы авторизации на основной файл index.html Возвращает массив jsx-элементов проектов. |

#### Модуль Project.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **getProjectInfo()** | Получает данные проекта по хэндлу из ссылки (project/:handle) Заполняет переменные category, description, name, source\_lang, target\_lang, visibility, owner\_id, created\_at, id |
| **getMembers()** | Получает список участников проекта. Изменяет переменную members. |
| **getSections()** | Получает список разделов проекта. Изменяет переменную sections. |
| **getUserRole()** | Получает роль авторизованного пользователя в текущем проекте. Изменяет переменную userRole |
| **changeSettings()** | Получает данные из полей ввода настроек и меняет параметры проекта. Вызывается по нажатию кнопки сохранения настроек. |
| **sendInvite()** | Отправляет приглашение присоединиться к проекту пользователю, логин которого написан в поле ввода. |
| **Render()** | Добавляет jsx-элемент страницы проекта на основной файл index.html Возвращает массив jsx-элементов проектов. |

#### Модуль Create.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **createProject()** | Использует данные из полей projectName, projectHandle, projectSourceLang, projectTargetLang, imagePath. Проверяет данные на корректность и создает проект. |
| **Render()** | Добавляет jsx-элемент страницы создания проектов на основной файл index.html Возвращает массив jsx-элементов проектов. |

#### Модуль User.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **getUserInfo()** | Использует данные из ссылки (user/login) и ищет всю информацию этого пользователя. Заполняет переменные username, about |
| **getProjects()** | Получает список проектов пользователя. |
| **Render()** | Добавляет jsx-элемент страницы создания проектов на основной файл index.html Возвращает массив jsx-элементов проектов. |

#### Модуль Search.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **searchProjects()** | Использует данные из полей ввода(name, category, owner, byDate, byPopularity) и возвращает список id подходящих проектов. |
| **fillProjects(list\_id)** | Помещает список проектов с нужными id в jsx-переменную projects. |
| **Render()** | Добавляет jsx-элемент страницы создания проектов на основной файл index.html Возвращает массив jsx-элементов проектов. |

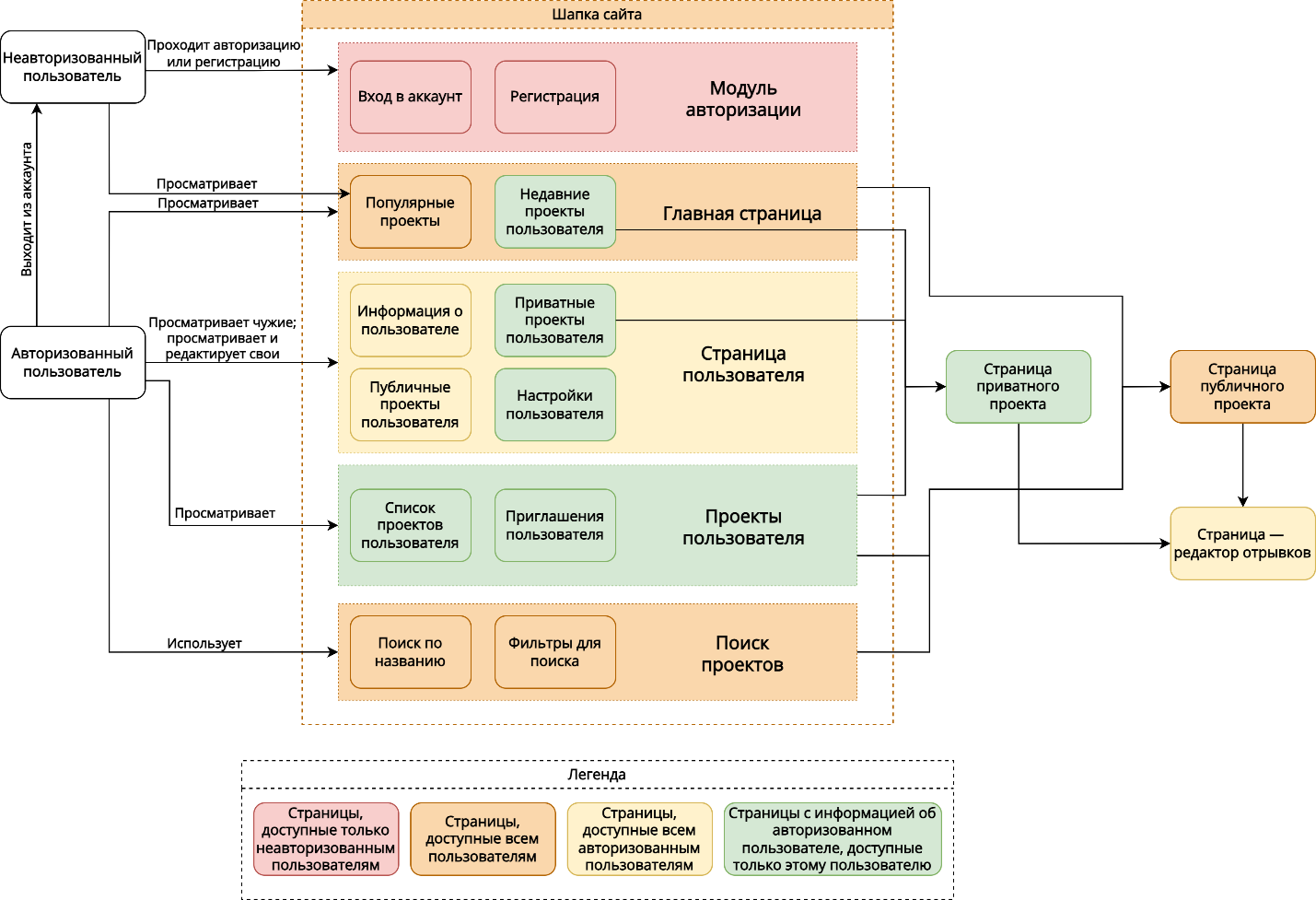
#### Модуль Editor.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **setTranslation(username, text)** | Формирует текст перевода из имени переводчика и текста перевода |
| **getString(id\_string)** | Получает отрывок с переводом по id отрывка |
| **getStrings()** | Получает список отрывков с переводами. |
| **addTranslation()** | Берет текст из поля ввода перевода и добавляет его в БД. |
| **Render()** | Добавляет jsx-элемент страницы редактора на основной файл .html Возвращает массив jsx-элементов проектов. |

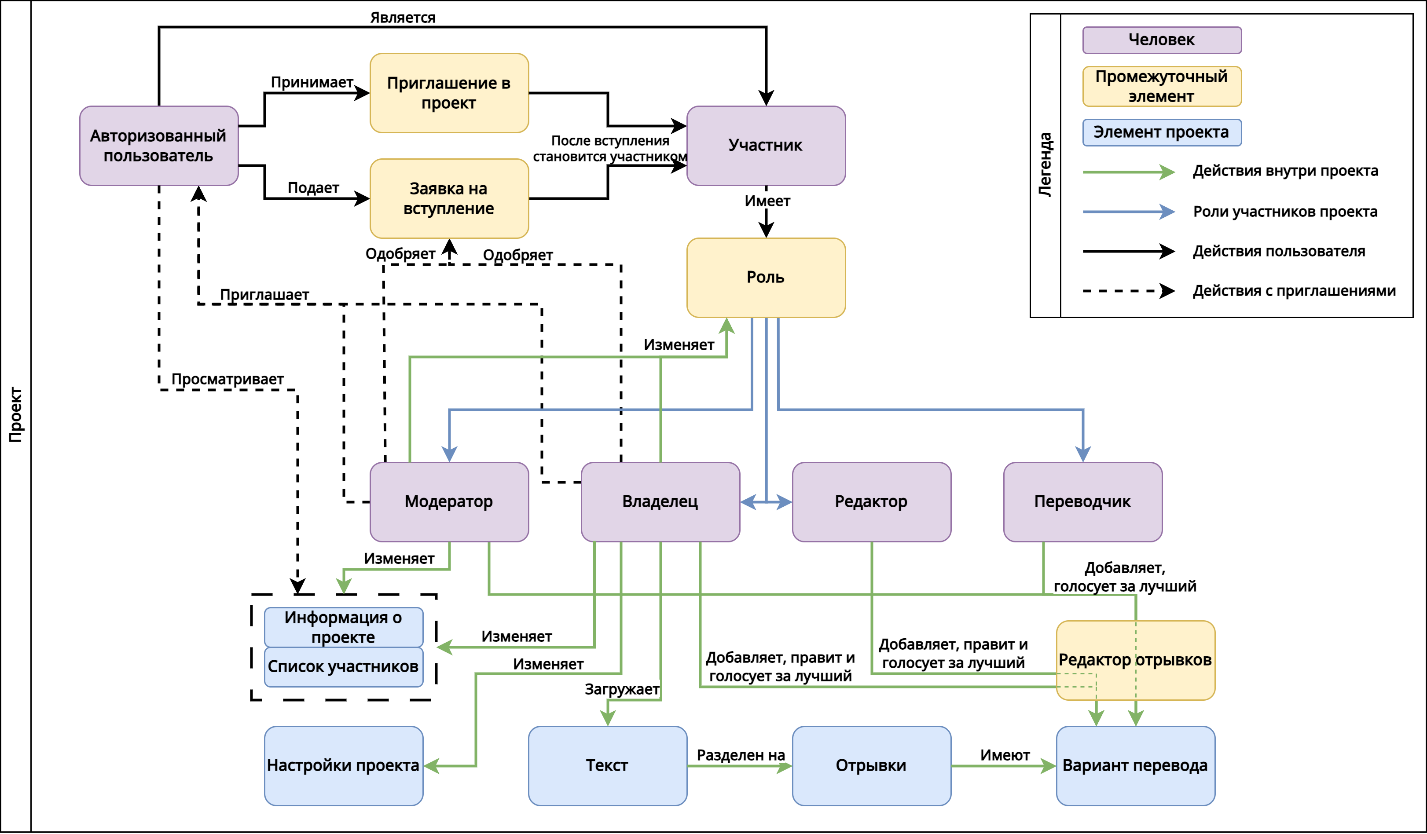
#### Модуль Search.js

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **getListOfProjects()** | Берет параметры из полей ввода фильтров и по ним ищет список проектов. |
| **showProjects()** | Обрабатывает нажатие кнопки поиска и создает переменную проектов. |
| **Render()** | Добавляет jsx-элемент страницы поиска на основной файл .html Возвращает массив jsx-элементов проектов. |

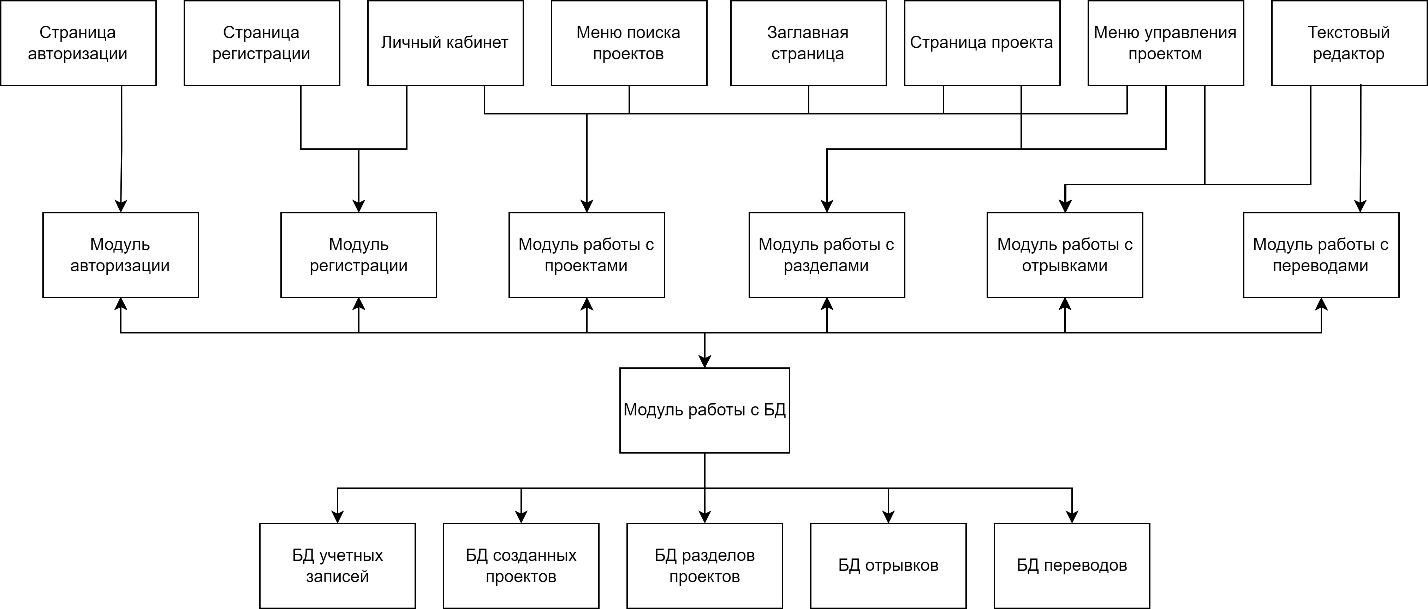
* + 1. Перечень сценариев по реализации наиболее важных и сложных процессов в системе



*Рисунок 2. Доступ и взаимодействие пользователя со страницами сайта*

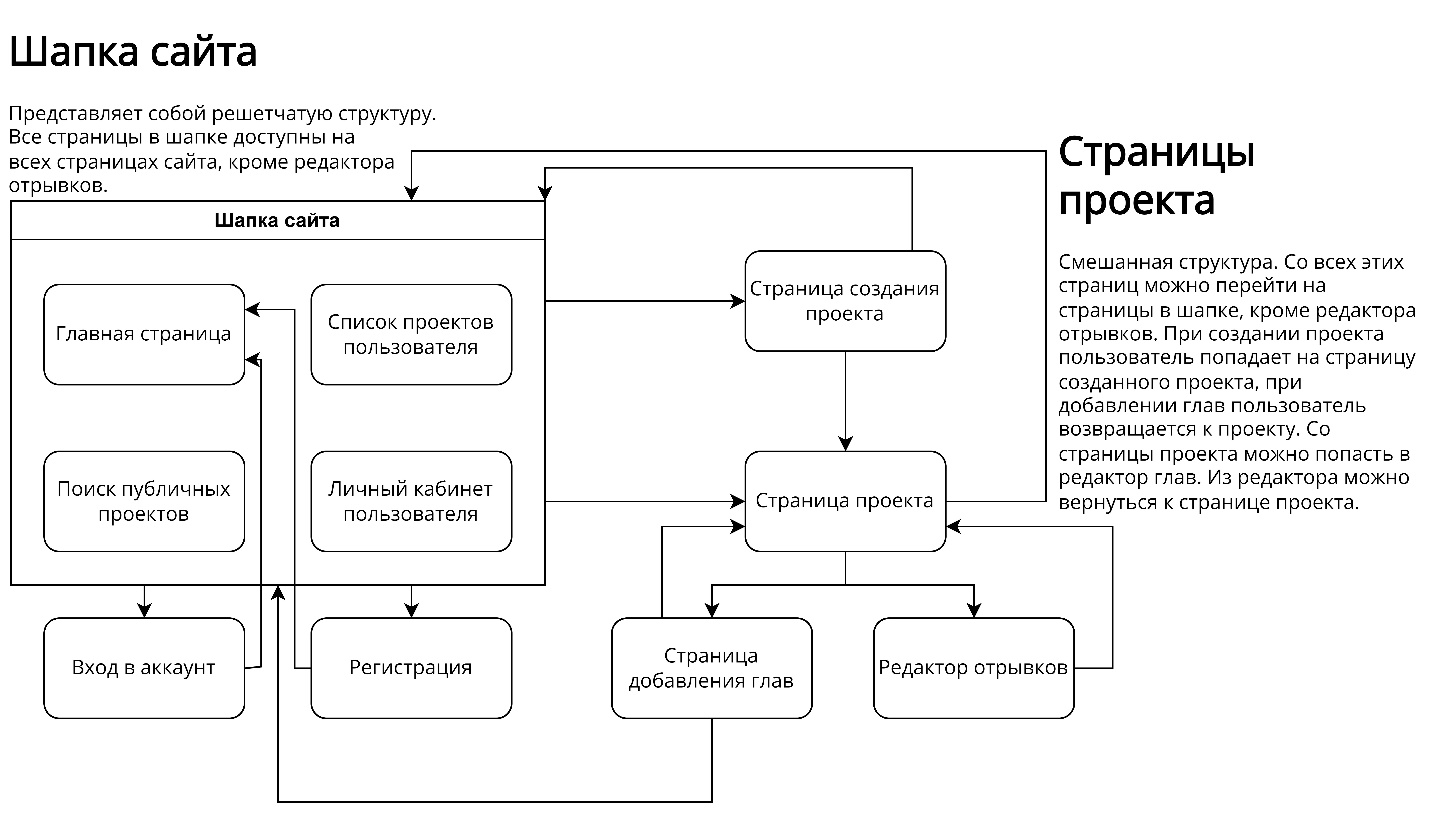


*Рисунок 3. Доступ и взаимодействие пользователей со страницами проекта*

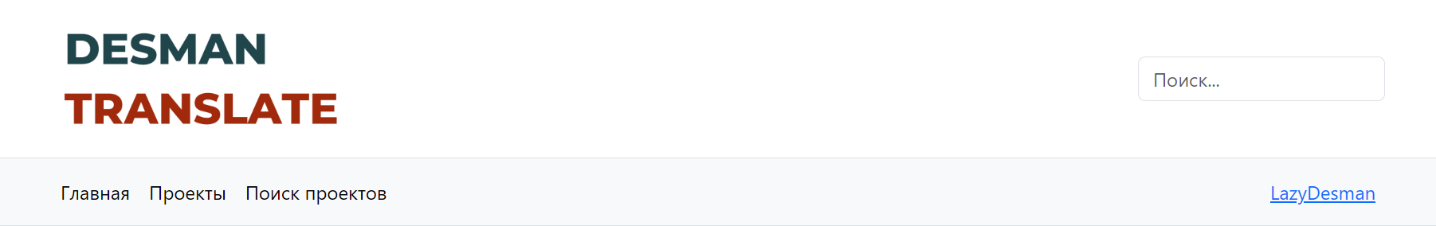


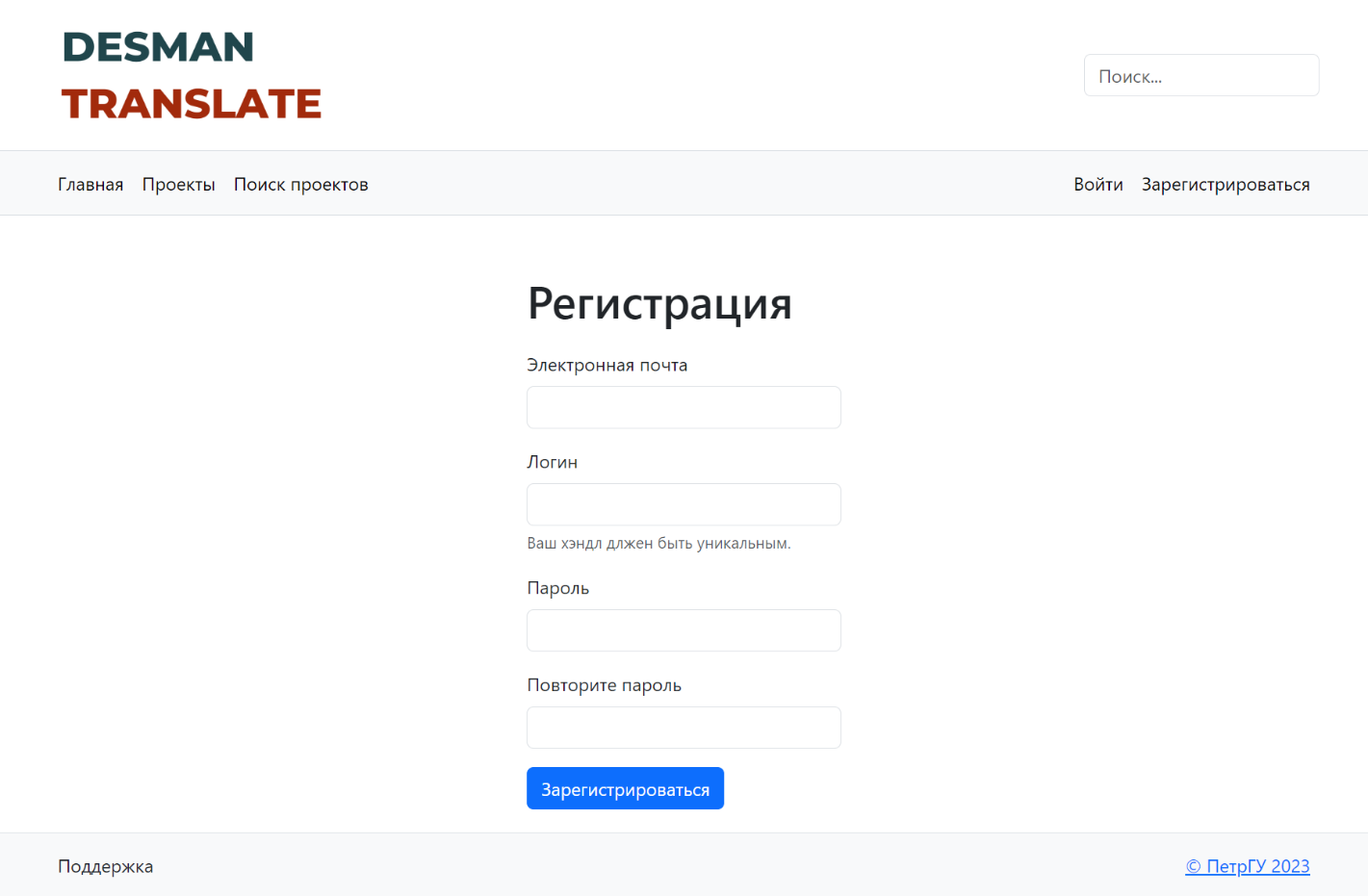
*Рисунок 4. Структура API веб-сервиса*

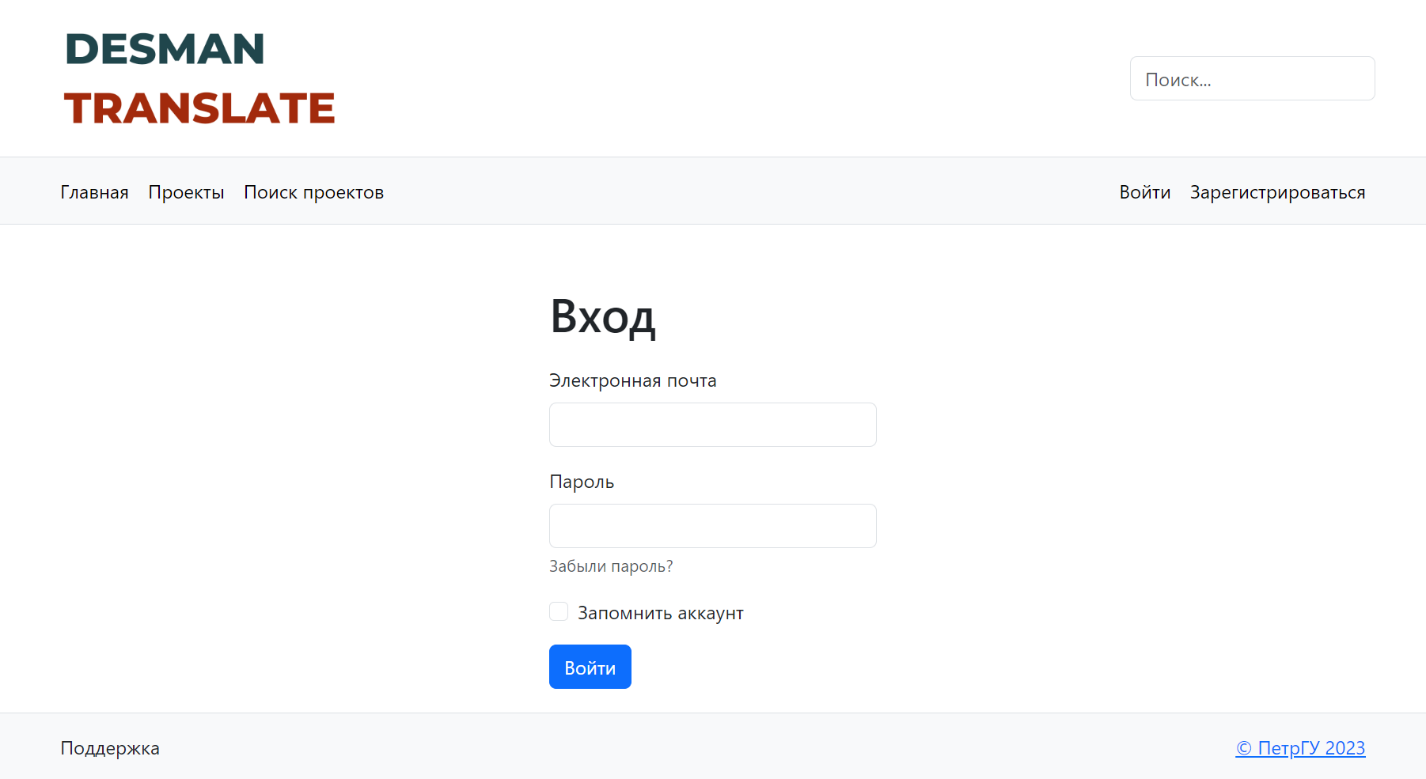
* + 1. Структура сайта, системы, дерево разделов и подразделов

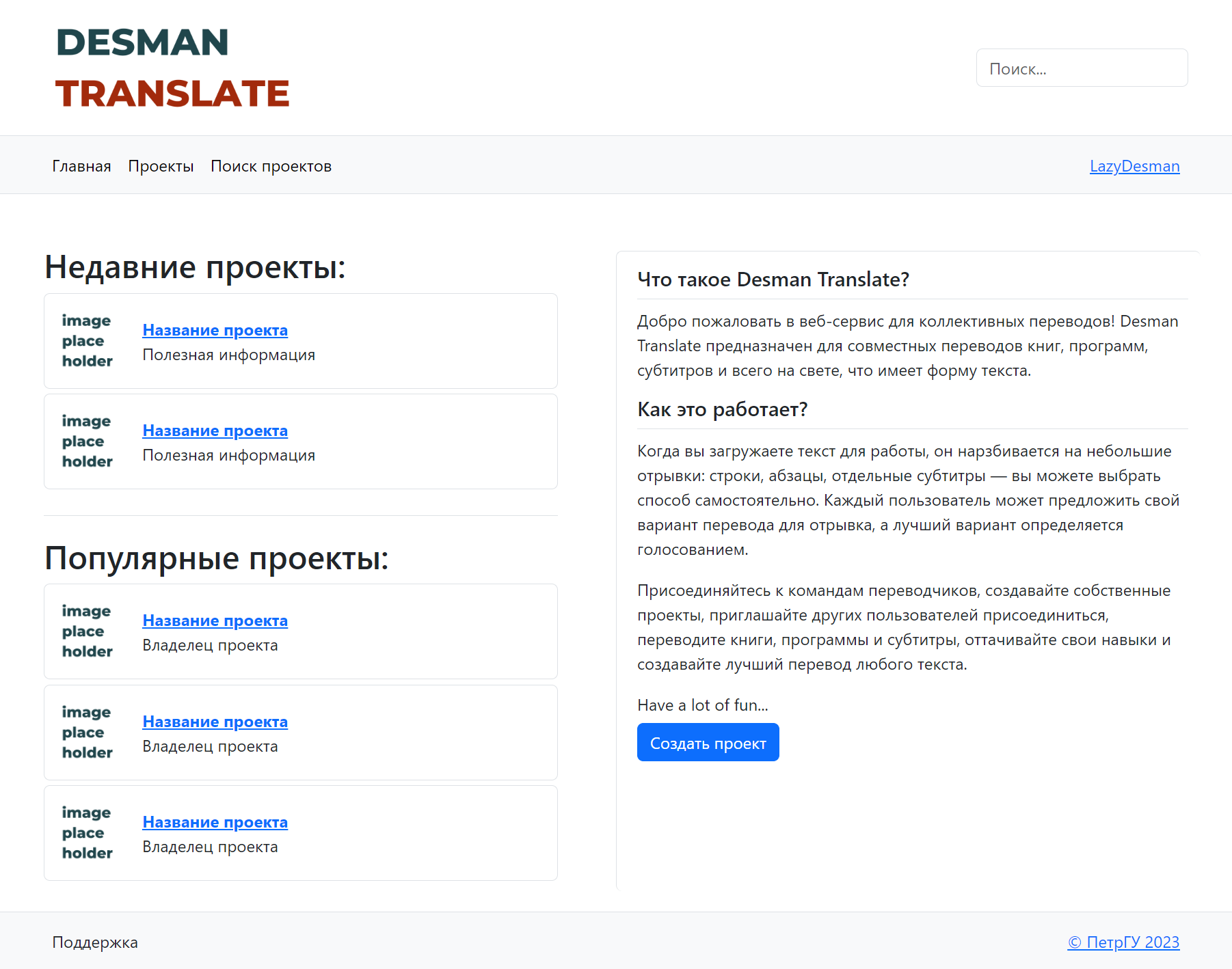


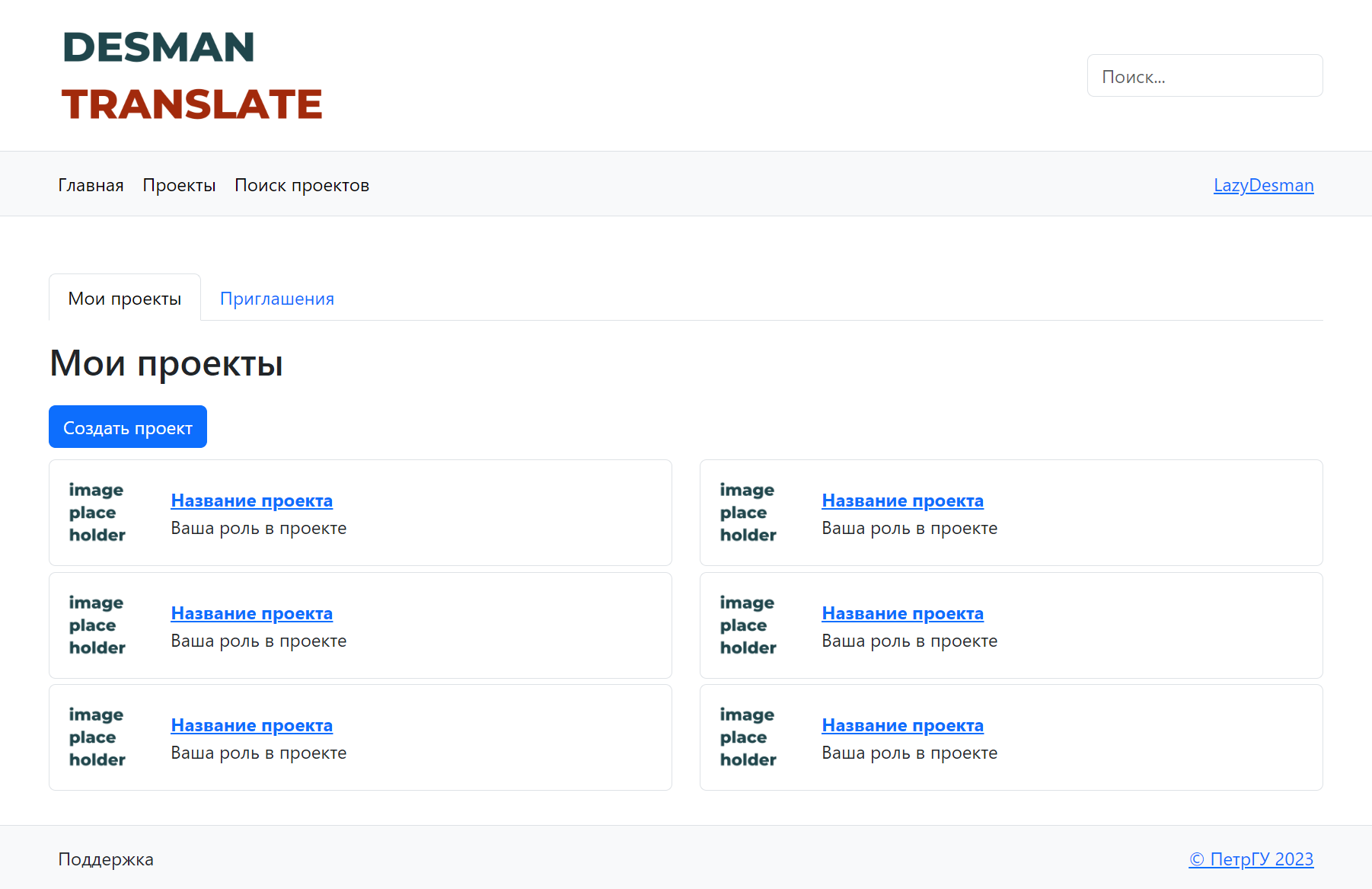
* + - 1. Перечень прототипов и экранных схем: прототипы

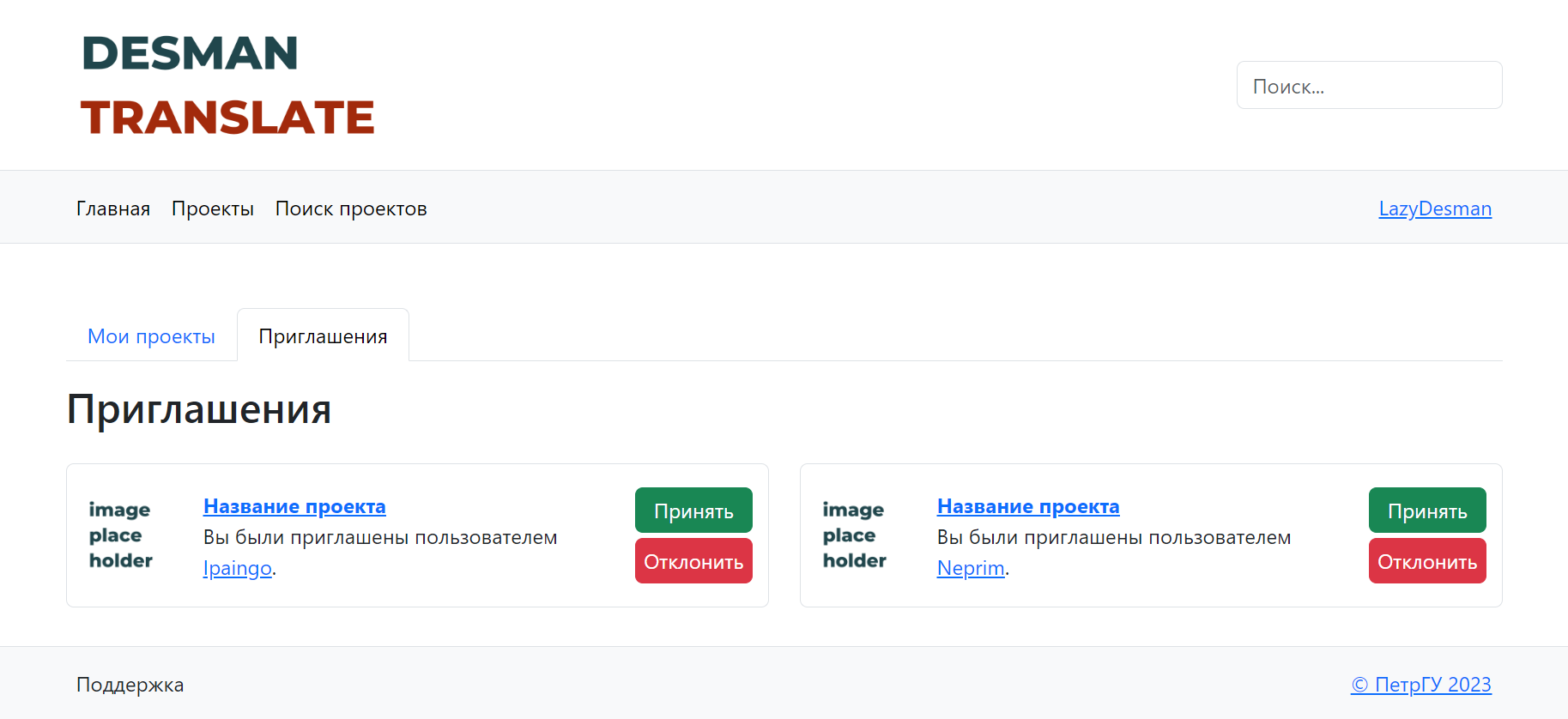
 *Рисунок 5. Прототип шапки сайта*

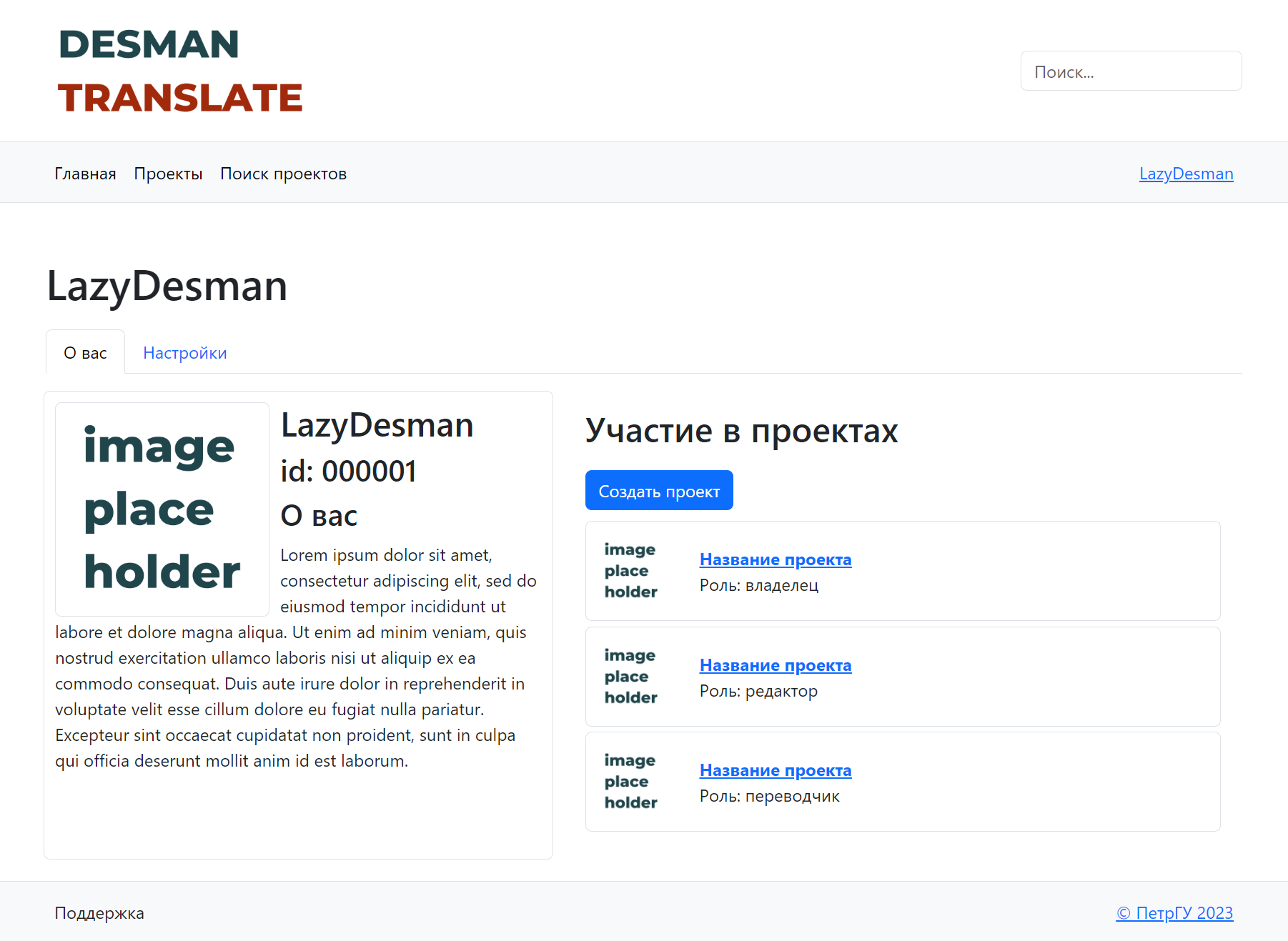
 *Рисунок 6. Прототип страницы регистрации*

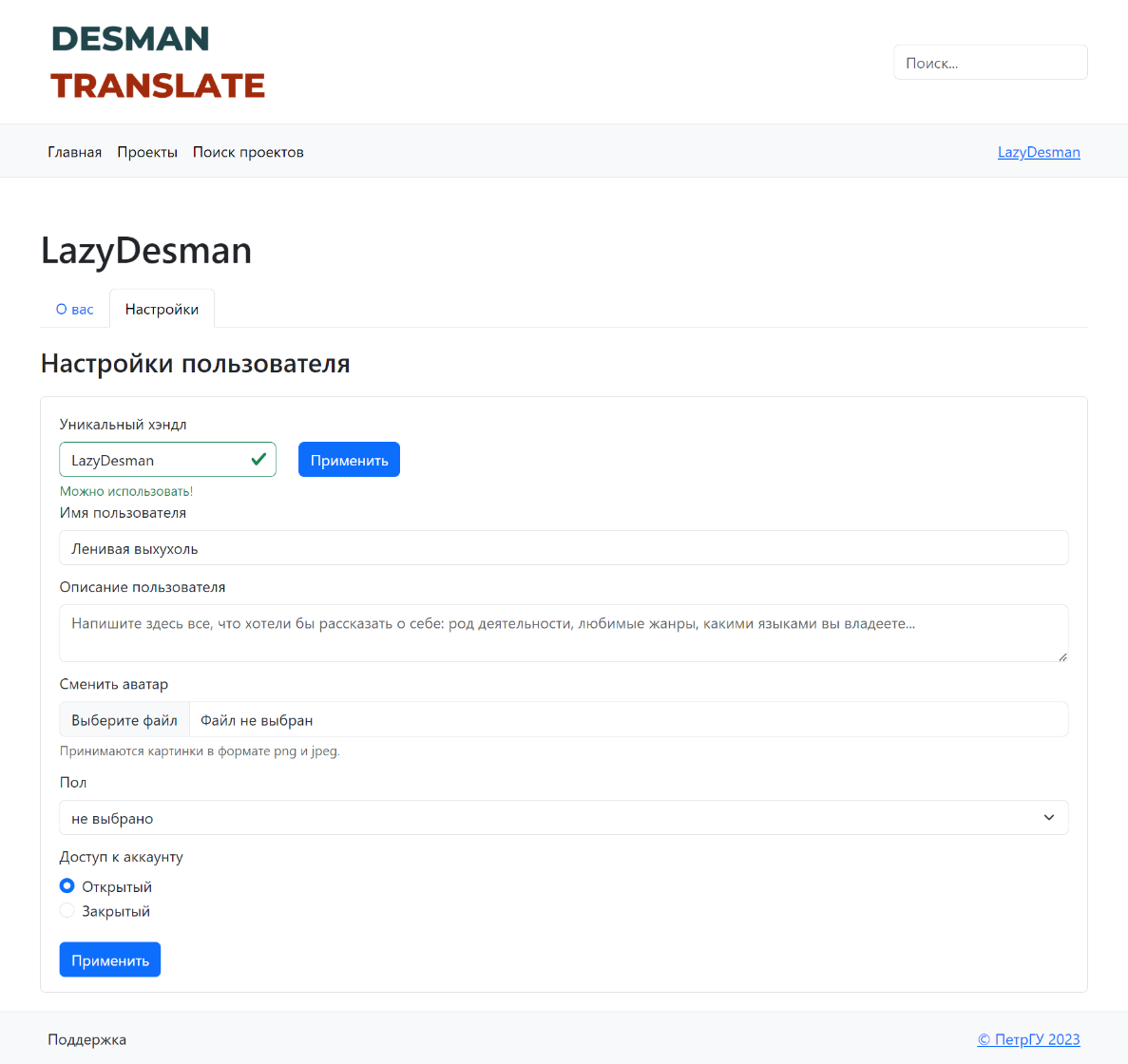
 *Рисунок 7. Прототип страницы авторизации*

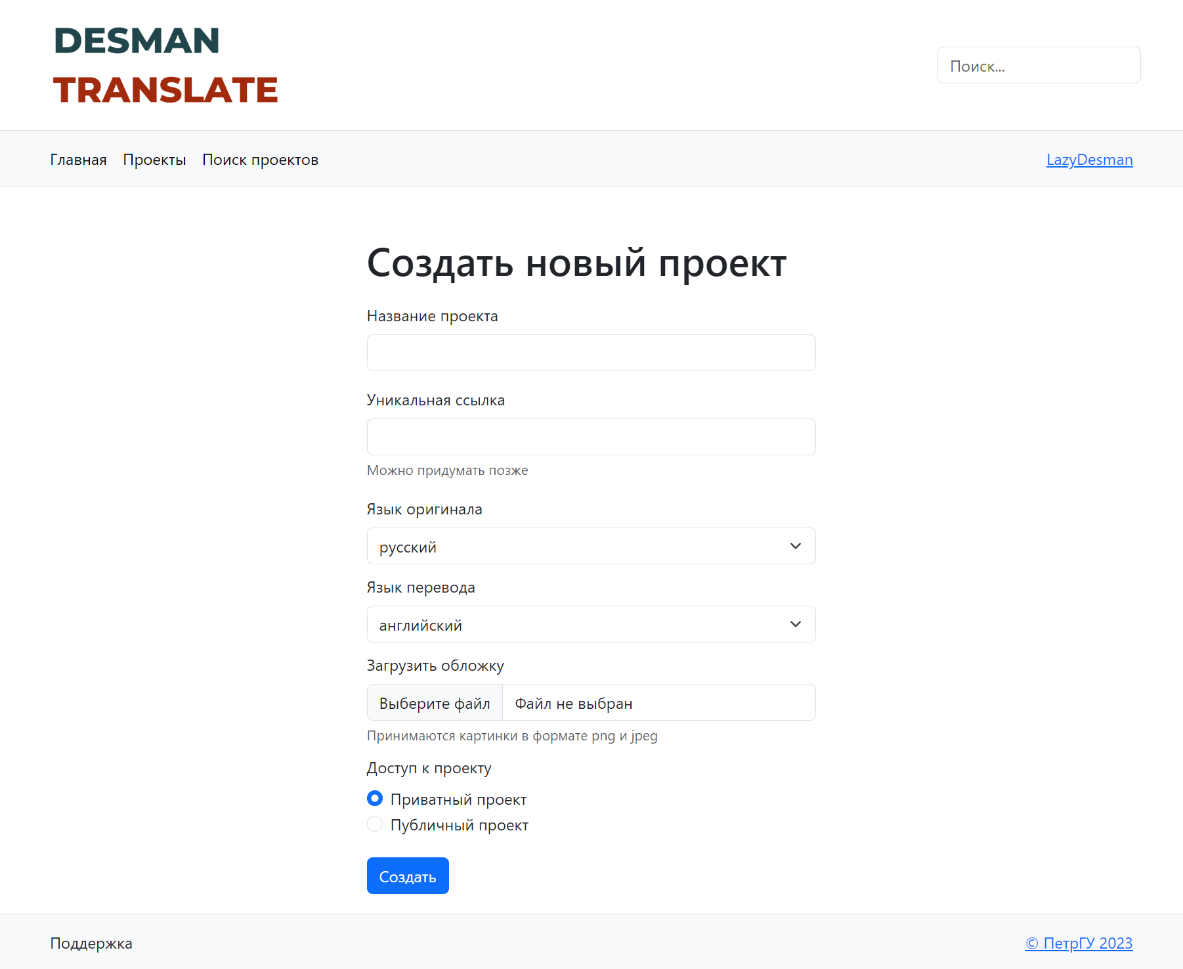
 *Рисунок 8. Прототип главной страницы*

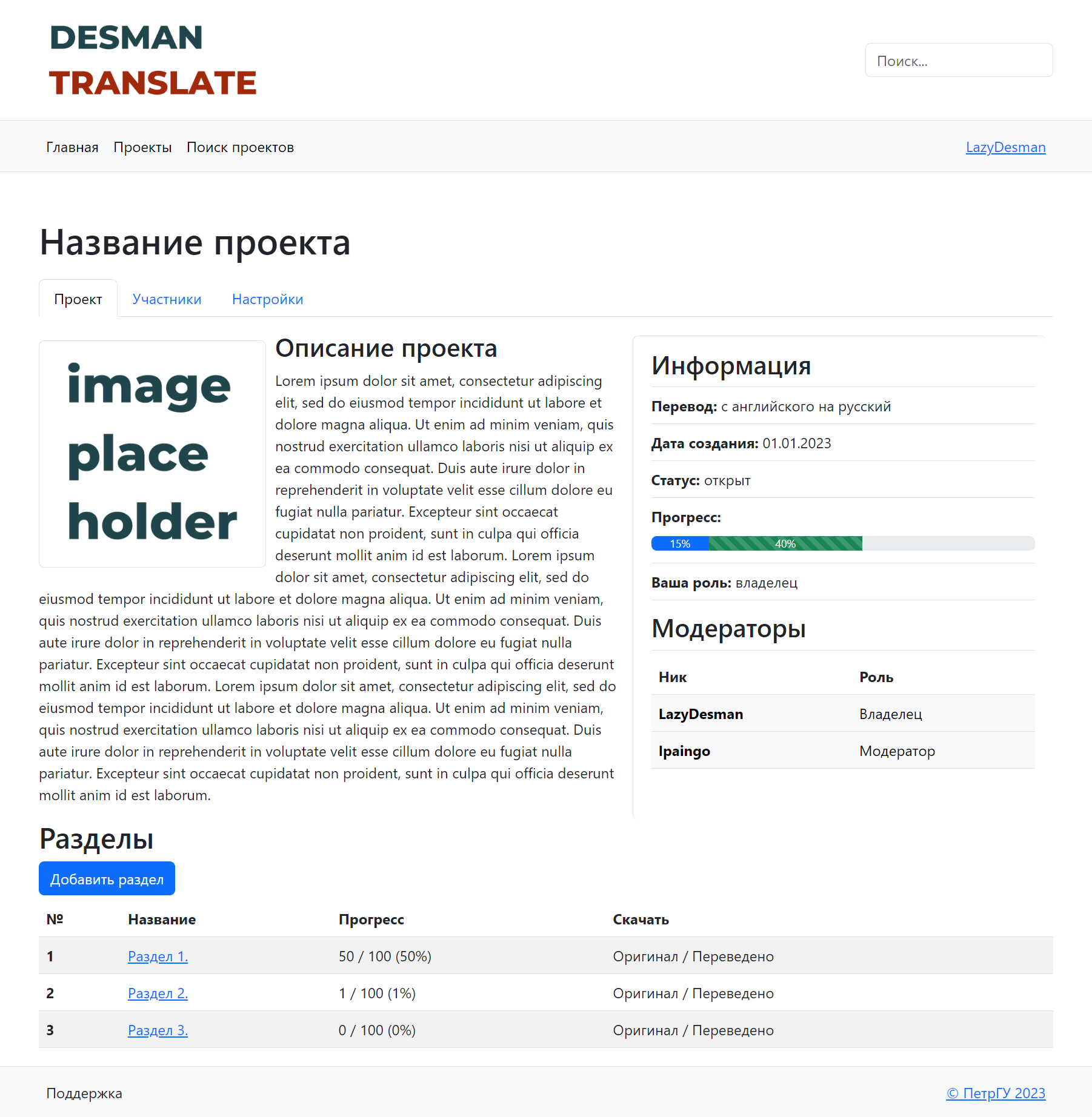
 *Рисунок 9. Прототип страницы списка проектов: проекты пользователя*

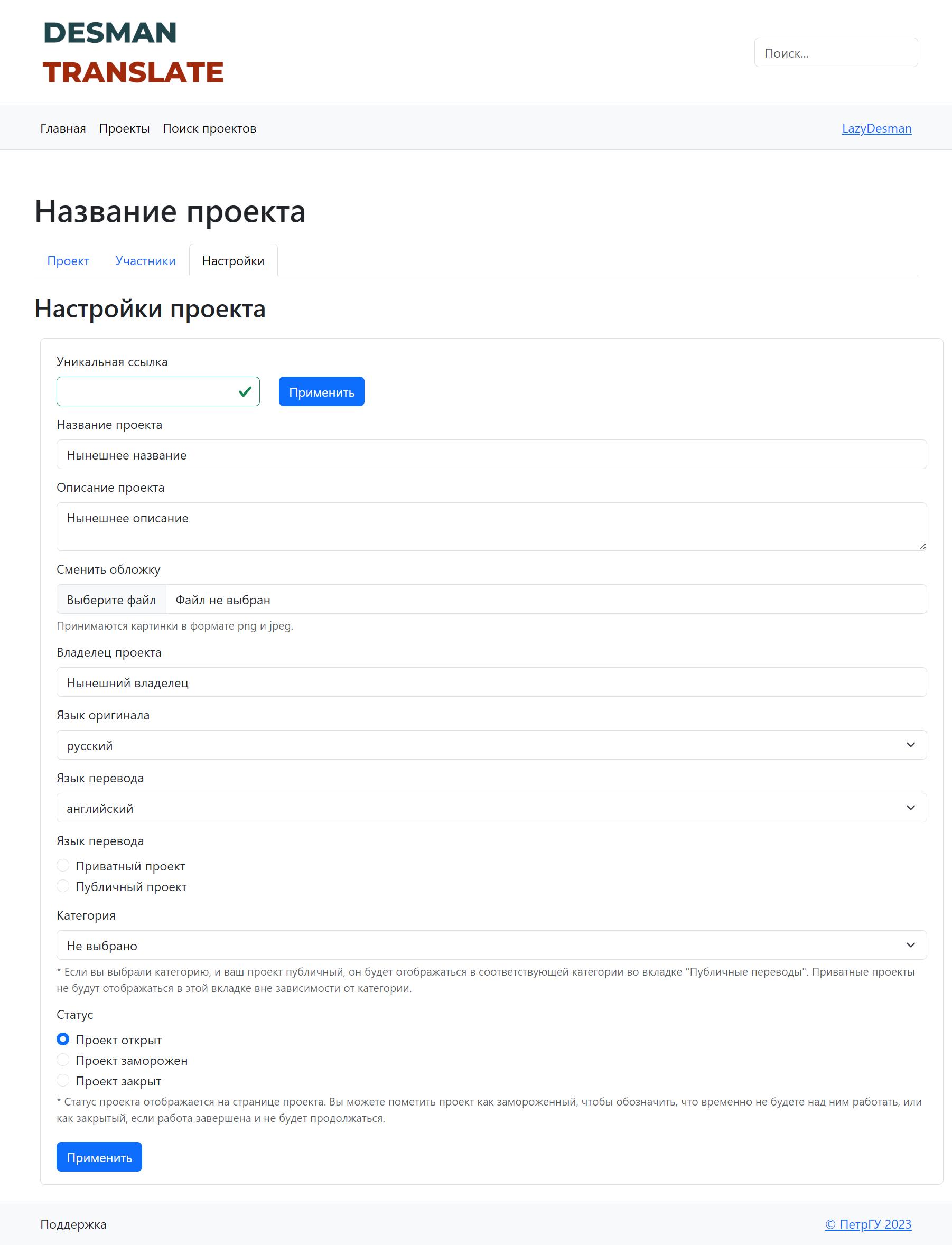
 *Рисунок 10. Прототип страницы списка проектов: приглашения*

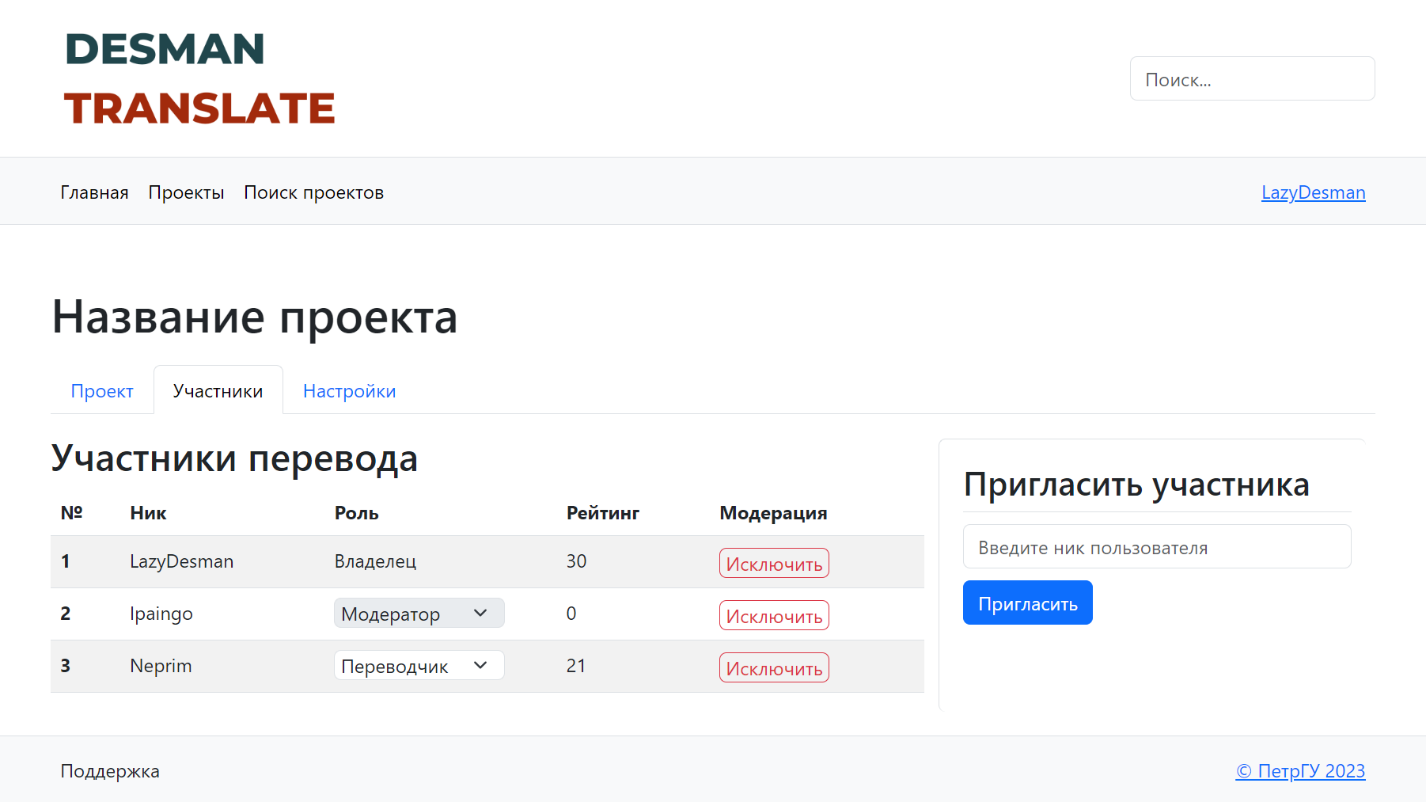
 *Рисунок 11. Прототип страницы пользователя*

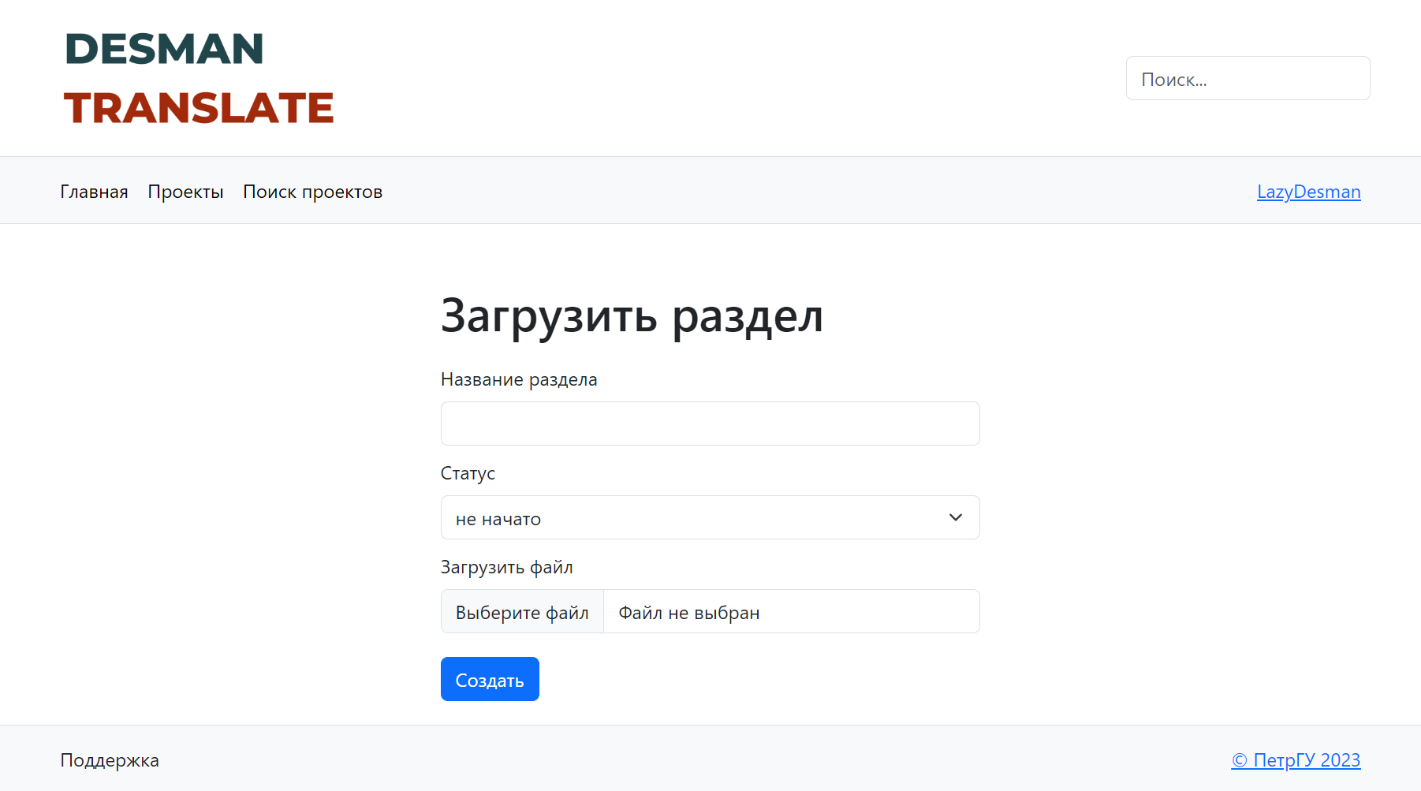
  
*Рисунок 12: Прототип страницы пользователя: настройки*

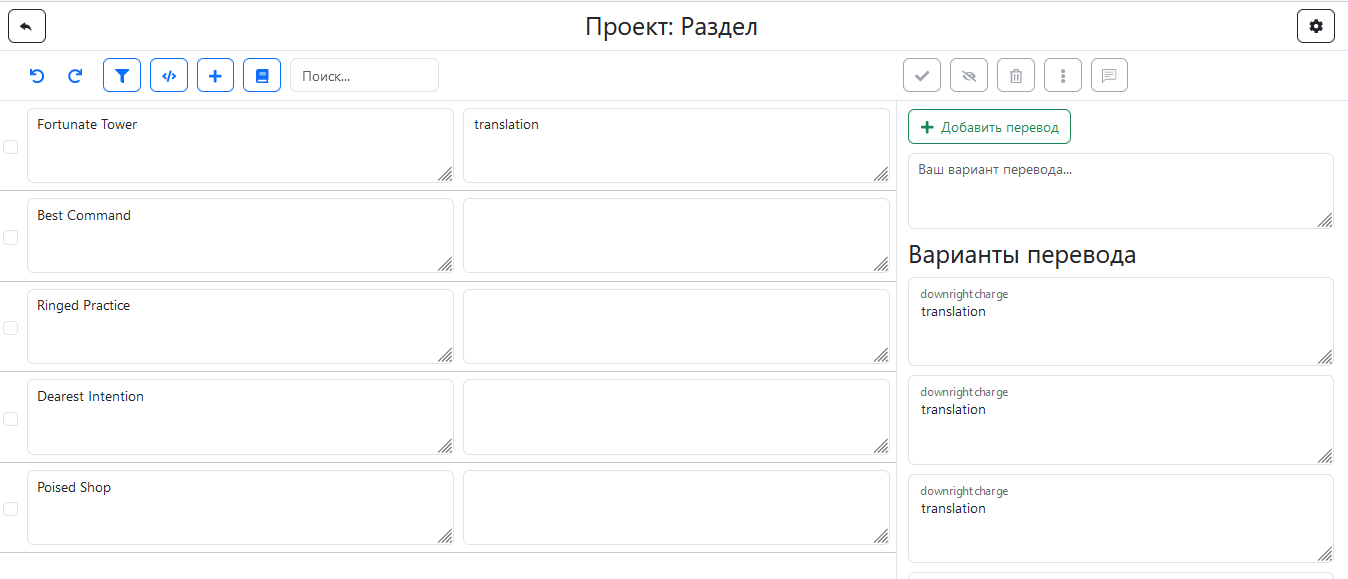
  
*Рисунок 13: Прототип страницы создания проекта*

*Рисунок 14: Прототип страницы проекта*

*Рисунок 15: Прототип страницы проекта: настройки*

*Рисунок 16: Прототип страницы проекта: участники*

*Рисунок 17: Прототип страницы загрузки раздела*

 *Рисунок 18: Редактор отрывков*

* 1. Требования к видам обеспечения

Сайт может быть запущен на локальной машине (персональном компьютере). В случае установки сайта на внешнем сервере, требования по мощности будут низкими. По мере увеличения аудитории и базы данных сервиса может потребоваться улучшение характеристик сервера.

* + 1. Программное обеспечение

Сайт, развернутый на сервере, доступен любому пользователю, чей персональный компьютер или ноутбук обладает поддержкой одного из браузеров: Google Chrome, Yandex, Opera, Safari, FireFox, Edge.

* + 1. Техническое обеспечение

Персональный компьютер или ноутбук пользователя; доступ к сети Интернет.

* + 1. Лицензии

Сайт распространяется по бесплатной лицензии открытого доступа – на страницу может перейти кто угодно, а также любой пользователь может зайти на страницу проекта на [Github](https://github.com/ipaingo/Desman-Translate) и скачать репозиторий с программным кодом.

* + 1. Интеграция с внешними системами, программная совместимость

Любой человек может использовать код сайта в своем приложении. Код сайта будет совместим с проектами, написанными на React.

* + 1. Математическое обеспечение

Математические методы и алгоритмы решения задач не используются на сайте.

* + 1. Организационное обеспечение

Сайт предназначен для использования командами разработчиков, работающими над переводом асинхронно. Сайт предоставляет пользователям инструменты для самостоятельной администрации своих проектов.

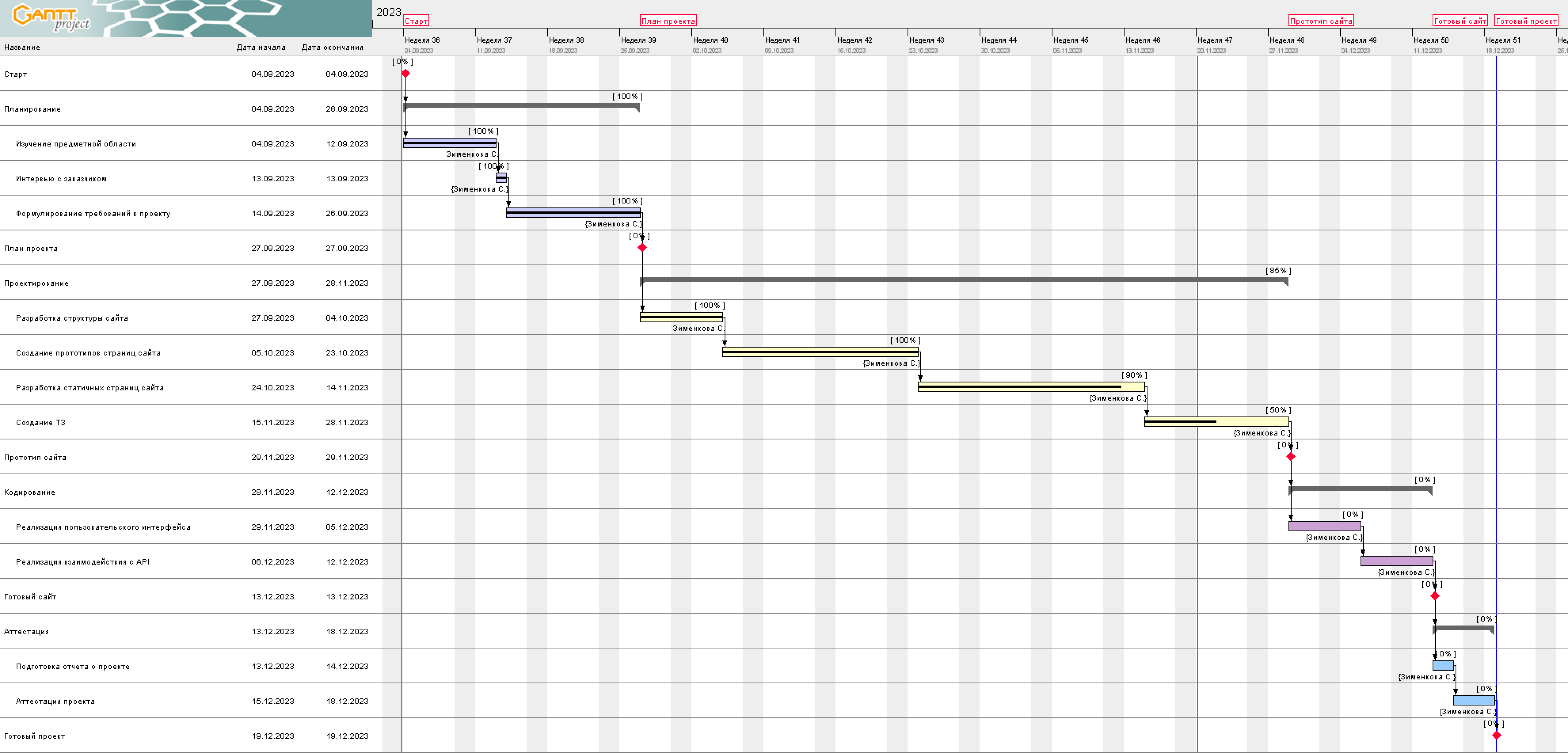
* + 1. Информационное обеспечение (контент, наполнение)

Сайт представляет собой инструмент для размещения, разбиения, перевода и выгрузки текстов. Информационное обеспечение сайта зависит от деятельности пользователей.

# Состав и содержание работ по созданию системы

|  |  |  |
| --- | --- | --- |
| **Этап** | **Срок выполнения** | **Результат** |
| Установление требований заказчика. | 20 дней | Разработчик имеет полное представление о желании заказчика. |
| Создание прототипов страниц сайта, работа над дизайном. | 10 дней | Создан макет сайта. |
| Работа над проектированием архитектуры сайта. | 20 дней | Реализован модуль кода, реализующий запросы по фильтрам к базе данным. |
| Реализация пользовательского интерфейса. | 35 дней | Сайт приобрел конечный внешний вид. |
| Компиляция приложения. | 5 дней | Сайт обладает интерфейсом и функционалом. |
| Тестирование и отладка. | 15 дней | Исправлены ошибки в работе приложения и интерфейсе. |

# Порядок контроля и приёмки системы



Разработчик обязан отчитываться о результате проделанной работы на каждой контрольной точке: он должен показать заказчику результаты в текстовом либо устном виде, предоставлять написанный код, прототипы страниц и согласовывать их с клиентом. В случае нахождения ошибок, разработчик обязан внести правки по словам заказчика.

# Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

Для ввода системы в действие достаточно одного человека, способного скомпилировать сайт, пользуясь соответствующей документацией по установке, администрированию и использованию.

# Требования к документированию (объемы, структура документов, соответствие ГОСТам)

* 1. Документация по установке
     1. Перейти по ссылке на страницу репозитория на сайте [Github](https://github.com/ipaingo/Desman-Translate)
     2. Скачать и разархивировать проект или клонировать git-репозиторий на сервере или в локальном хранилище.
     3. Окрыть терминал, перейти в папку скачанного проекта, убедиться, что пользователь имеет установщик пакетов npm. Выполнить команду "npm install".
     4. Если необходимо запустить сайт на локальном сервере – написать команду в терминале "npm start". Сайт будет доступен по адресу <http://localhost:3000/>
     5. Если необходимо развернуть сайт на сервере – написать команду "npm run build". Создастся директория build – это готовое приложение, которое можно разместить на сервере.
     6. Для запуска проекта на локальной машине:
     7. В терминале, находясь в директории проекта, написать команду npm run build.
     8. В директории создастся папка build, в которой находится файл index.html, который и является исполняемым файлом.
     9. Папку build можно разместить на сервере.
  2. Документация по администрированию

На данном этапе развития сайт не требует администрирования. По мере увеличения аудитории сайту может потребоваться администрация. Будет создан свод правил; пользователи имеют возможность связаться с разработчиком и сообщить о нарушении. Нарушивший правила пользователь теряет доступ к своему аккаунту на сайте.

* 1. Документация по использованию

**Руководство пользователя**

1. Регистрация и авторизация

В первую очередь пользователю сайта необходимо зарегистрироваться и авторизоваться. Для этого нужно в шапке страницы нажать на кнопку "Войти", если пользователь уже создал аккаунт, или "Зарегистрироваться", если пользователь создает аккаунт впервые. Далее необходимо заполнить анкету в зависимости от выбранного действия. После авторизации кнопки "Войти" и "Зарегистрироваться" в шапке сайта заменяются на ссылку с вашим именем пользователя, ведущую на вашу страницу, и кнопку выхода из аккаунта.

1. Главная страница

Главная страница сайта — страница, которая отображается при переходе на веб-сайт. На главной странице пользователю доступно описание веб-сервиса и список недавних проектов. При нажати на название проекта вы будете перенаправлены на страницу выбранного проекта. Также на глвной странице доступна кнопка "Создать проект", при нажатии на которую вы будете перенаправлены на страницу создания проекта.

1. Шапка и футер сайта

Шапка и футер сайта отображаются на всех страницах, кроме редактора отрывков. Они созданы для удобства пользователя.

Шапка сайта используется для удобной навигации по веб-сервису. С помощью нее вы можете вернуться на главную страницу, перейти к странице ваших проектов, перейти к странице поиска проектов, перейти на вашу страницу пользователя, выйти из аккаунта, осуществить быстрый поиск с помощью поля в правом верхнем углу.

Футер сайта содержит ссылки, по которым можно связаться с разработчиками веб-сервиса. При нажатии на кнопку "Поддержка" вы можете отправить письмо на электронную почту разработчика. При нажатии на ссылку "ПетрГУ 2023" вы будете перенаправлены на сайт кафедры ИМО.

1. Список проектов пользователя

При нажатии на кнопку "Проекты" в шапке сайта вы будете перенаправлены на страницу ваших проектов. На этой странице отображаются две вкладки: "Мои проекты" и "Приглашения". На вкладке "Мои проекты" вы можете перейти к одному из своих проектов, нажав на название этого проекта, а также создать новый проект, нажав на кнопку "Создать проект". На вкладке "Приглашения" отображаются ваши приглашения в проекты от других пользователей. Вы можете принять приглашение и стать участником проекта, или отклонить его.

1. Страница пользователя

На странице пользователя отображается имя пользователя, список его публичных проектов, описание и аватар пользователя. Если вы находитесь на своей странице пользователя, вам также доступна вкладка "Настройки", на которой вы можете изменить информацию о себе.

1. Создание проекта

При нажатии на кнопку "Создать проект" на одной из страниц сайта вы будете перенаправлены на страницу создания проекта. Здесь вам необходимо заполнить все поля о новом проекте, а затем нажать на кнопку "Создать проект", после чего ваш новый проект станет отображаться на сайте.

В поле "Название проекта" необходимо вписать название произведения на языке оригинала или перевода, чтобы пользователи могли сразу понять, какое произведение переводится в этом проекте.

В поле "Уникальная ссылка" нужно придумать и ввести уникальную ссылку на проект. Ссылка должна состоять из строчных латинских букв и нижних подчеркиваний. Сервис проверит правильность и уникальность введенной ссылки при нажатии на кнопку "Создать проект" в конце заполнения анкеты.

В полях "Язык оригинала" и "Язык перевода" необходимо указать, с какого языка на какой будет переводиться текст. Это нужно для удобства других пользователей, желающих принять участие в переводе.

Вы можете загрузить обложку с помощью поля "Выберите файл". При нажатии на это поле откроется файловая система на вашем устройстве, и вы сможете загрузить фотографию для обложки. Это не обязательное для заполнения поле.

Далее необходимо указать доступ к проекту. По умолчанию выбран "Приватный проект", то есть созданный проект не будет виден другим пользователям, пока они не станут участниками этого проекта. Вы можете выбрать "Публичный проект", тогда все пользователи смогут видеть ваш проект при поиске, при переходе по прямой ссылке и на главной странице.

1. Страница проекта

При нажатии на название проекта на главной странице, на странице пользователя или в списке проектов отображается страница этого проекта. На этой странице есть три вкладки: "Проект", "Участники" и "Настройки". Вкладка "Настройки" доступна только модераторам и владельцу проекта.

На вкладке "Проект" доступна информация о проекте: название, прогресс, язык оригинала и перевода, описание, обложка, модераторы и список разделов. Раздел — это блок текста, например, глава книги, загруженная на сайт для перевода. Вы можете загрузить новый раздел, нажав на кнопку "Добавить раздел".

На вкладке "Участники" отображаются участники этого проекта и их роли.

Участники могут иметь следующие роли:

* Владелец — пользователь, который создал проект. Имеет полный доступ к управлению проектом и работе над переводом.
* Переводчик — роль, которую автоматически получает пользователь, ставший участником проекта. Может добавлять и изменять свои варианты перевода и голосовать за лучший перевод.
* Модератор — роль, назначаемая владельцем и другими модераторами. Помимо возможностей переводчика имеет доступ к изменению ролей участников (кроме владельца и других модераторов), может изменять настройки проекта.
* Редактор — роль, назначаемая владельцем и модераторами. Помимо возможностей переводчика может изменять чужие варианты перевода и имеет больший вес при голосовании за лучший вариант перевода.

Модераторам и Владельцу проекта доступно также поле "Пригласить участника". При введении имени пользователя и нажатии кнопки "Пригласить" указанный пользователь получает приглашение в ваш проект, которое он может принять или отклонить.

На вкладке "Настройки" пользователи с соответствующими правами могут изменять название, описание, обложку проекта, уникальную ссылку на проект, язык оригинала и язык перевода, настраивать доступ к проекту, его категорию и статус. Владелец может передать владение проектом другому пользователю, введя его имя пользователя в поле "Владелец проекта".

1. Работа в редакторе отрывков

После того, как вы загрузили и разбили на отрывки текст поризведения, можно начать работу над его переводом. При нажатии на название раздела на странице проекта открывается редактор отрывков, в который загружается этот раздел, разбитый на отрывки. При наведении на кнопки в редакторе вы будете получать всплывающие подсказки о работе этих кнопок.

Работа над переводом осуществляется следующим образом:

1. Участник выбирает отрывок, нажав на строку-отрывок в любом месте.
2. Участник выбирает в окне справа текстовое поле, в котором вводит свой вариант перевода отрывка.
3. Участник нажимает на кнопку "Добавить перевод", и введенный им текст отображается под текстовым полем в списке вариантов перевода. Имя пользователя добавившего перевод участника отображается рядом с вариантом перевода.
4. Последний добавленный вариант перевода отображается в строках отрывков как текущий вариант перевода текста.
   1. Программная документация

Подробная документация приведена в описании модулей программного кода.

Структура кода:

DesmanTranslate

├── Controllers

│   ├── APIController.js <-- Контроллер для API запросов

│   ├── Authorization.js <-- Контроллер для обработки авторизации и регистрации

│   ├── Middleware.js <-- Контроллер для промежуточных обработчиков

│   ├── PageController.js <-- Контроллер для запросов страниц

│   └── DBUtils.js <-- Вспомогательные функции для работы с БД

├── Handlers <-- Хэндлы для соответствующих сущностей

│   ├── Invite.js

│   ├── OrigString.js

│   ├── Project.js

│   ├── Section.js

│   ├── Translation.js

│   └── User.js

├── Pages <-- Компоненты для генерации соответствующих страниц

│   ├── Editor.js

│   ├── Footer.js

│   ├── Home.js

│   ├── Login.js

│   ├── NavBar.js

│   ├── Project.js

│   ├── ProjectSettings.js

│   ├── Register.js

│   ├── Search.js

│   ├── SectionEditor.js

│   └── User.js

├── Routers

│   ├── APIRouter.js <-- Маршрутизатор API запросов

│   └── PageRouter.js <-- Маршрутизатор запросов страниц

├── index.js

# Реквизиты и подписи сторон

Заказчик \_\_\_\_\_\_\_\_\_\_\_\_ / М. А. Колчин

Исполнитель \_\_\_\_\_\_\_\_\_\_\_\_ / С. Э. Зименкова