Петрозаводский государственный университет
Институт математики и информационных технологий
Кафедра Информатики и Математического Обеспечения

Отчет о научно-исследовательской работе

Разработка клиентской части веб-сервиса для коллективных переводов

Выполнила:
студентка 3 курса группы 22305

С. Э. Зименкова _____
подпись

Научный руководитель:
Д. Б. Чистяков

_____
подпись

Итоговая оценка

_____
оценка

Петрозаводск — 2024

Содержание

Введение

Перевод больших массивов текста зачастую является трудоемкой задачей, требующей большого количества времени. Задача усложняется, если необходимо перевести художественное произведение, научную работу, текст выступления на конференции или справочные материалы, содержащие большое количество специальной лексики. Процесс перевода становится более эффективным, если он выполняется коллективом профессионалов, выполняющих разные роли: помимо самих переводчиков в работе принимают участие редакторы и руководители проекта, которым нужно разделять обязанности между работниками, согласовывать процесс и утверждать прошедшие редакцию главы.

Профессиональный перевод не является единственной возможной формой работы над текстом на иностранном языке. Не все заказчики могут позволить себе оплатить профессиональный перевод значительного по объему текста. Примером подобных текстов могут выступать интерфейс ПО и компьютерных игр, узконаправленная литература, конспекты лекций и конференций, субтитры к видеоматериалам, находящимся в открытом доступе. Такую работу часто выполняют волонтеры, не все из которых обладают достаточными навыками для выполнения профессионального перевода. В проектах, связанных с любительским переводом, помимо синхронизации и распределения частей текста между участниками, необходимо проверять качество работы и выбирать наилучшие варианты перевода, так как волонтеры и любители не несут достаточной ответственности за свою работу и не могут гарантировать его верность.

Решением проблемы распределения, синхронизации и утверждения перевода для команд переводчиков любой степени профессионализма является удобный инструмент, позволяющий пользователям совместно работать над переводом.

Разработка веб-сервиса разделена на три этапа:

- разработка структуры (веб-API, база данных) и веб-страниц;
- установка связи между back-end и front-end составляющими проекта, разработка инструмента-редактора;
- доработка и улучшение сервиса, расширение функционала.

Цель данной работы — разработать клиентскую часть веб-сервиса для коллективных переводов. На предыдущем этапе уже были реализованы HTML-шаблоны с подключенными стилями, поэтому на этом этапе необходимо внести улучшения в существующие шаблоны, сделать их динамическими и осуществить взаимодействие клиентской части с веб-API.

Для осуществления поставленной цели необходимо выполнить следующие задачи:

- Сформулировать требования к клиентской части веб-сервиса;
- Доработать макет пользовательского интерфейса;
- Разработать и интегрировать клиентскую часть веб-сервиса в информационную систему.

1 Анализ предметной области

1.1 Описание требований к разрабатываемому сервису

Разрабатываемый сервис для удобства получил временное название "Desman Translate".

1.1.1 Глоссарий

- **Текст** — содержание книг, субтитров, подписей в компьютерных программах и др.
- **Раздел** — набор отрывков текста, объединенных в массив по усмотрению пользователя для его удобства.
- **Участник** — пользователь, имеющий доступ к работе над текстом в проекте.

1.1.2 Описание функционала разрабатываемого сервиса

Desman Translate — веб-сервис, позволяющий пользователям совместно работать над проектами по переводу текстов. Сервис реализован в виде веб-сайта.

Сервис рассчитан на два типа переводчиков:

- Переводчики-любители, которым нужно найти команду, т. к. они не могут перевести некоторое произведение в одиночку. Для удобства таких пользователей сервис должен предоставлять возможность для работы большого количества людей над одним и тем же текстом, а также поиска существующих проектов, где пользователи могли бы предоставить помощь с переводом, и популяризации своих проектов, чтобы найти других переводчиков, которые готовы помочь.
- Переводчики-профессионалы, которым нужен инструмент для совместной асинхронной работы над переводом, а также возможность распределения работы между членами команды, рецензирования готового перевода. Профессиональным командам также важна возможность скрыть свою работу от публики и не допустить других людей до работы над текстом, т. к. они могут работать над произведением или программой, защищенными авторским правом.

Проект — это созданная пользователем страница сайта, которая определяет процесс работы над некоторым произведением. Проект имеет название, описание, раздел, участников, статус (в работе, заморожен, закрыт), видимость (открытый, скрытый). Статус

проекта влияет на представление и доступ к работе над текстом. Замороженный проект — незавершенный перевод, над которым временно прекращена работа; участники не могут работать с текстом, пока владелец проекта не откроет его вновь. Закрытый проект — завершенный перевод, в котором больше нельзя загружать разделы и работать над текстом. Проект также может быть публичным и приватным. Эта возможность добавлена для того, чтобы проект мог отображаться или не отображаться в поиске и быть видимым для пользователей, не являющихся участниками. Владелец проекта может скрыть свой проект от других пользователей по своему усмотрению (например, если переводимый текст защищен авторским правом и не должен быть в открытом доступе). Для проекта можно настроить и другие параметры. Например, для каждого типа пользователей можно отдельно настроить права доступа к проекту (например, могут ли пользователи, не являющиеся участниками, читать текст, загруженный в проект; могут ли пользователи присоединяться к проекту без приглашения/одобрения).

Работа над переводом осуществляется следующим образом:

- Пользователь, имеющий соответствующие права доступа, загружает в проект текст.
- Текст автоматически или по усмотрению пользователя разбивается на отрывки. Отрывком может быть строка, абзац, страница и любое другое количество текста.
- Пользователи могут добавлять для отрывков свои варианты перевода и голосовать за лучший вариант.
- Лучший по результатам голосования вариант перевода отображается вверху списка и считается итоговым переводом.
- Варианты переводов объединяются в текст при экспорте перевода. Пользователь может скачать переведенный текст и использовать его далее по своему усмотрению. Пользователь может экспортировать объединение лучших вариантов перевода или объединение вариантов перевода от отдельного пользователя. Таким образом может быть получен лучший текст по результатам голосования, а также несколько переводов от разных переводчиков.

Есть несколько типов пользователей: незарегистрированный пользователь и зарегистрированный пользователь. Незарегистрированный пользователь может зарегистрироваться или войти в аккаунт, просматривать существующие проекты на сайте.

Зарегистрированный пользователь может иметь разные роли в разных проектах. Зарегистрированный пользователь может просматривать существующие проекты, присоединяться к существующим проектам, создавать свои проекты (становясь при этом владельцем созданного проекта), модерировать свои проекты (в том числе назначать роли участников проекта), загружать тексты, разбивать тексты на разделы и отрывки, работать над переводом отрывков. Участник проекта может иметь следующие роли в проекте:

- Переводчик имеет возможность добавлять варианты перевода отрывков, голосовать за лучший вариант перевода.
- Редактор может все, что может переводчик, а также может редактировать существующие варианты перевода и имеет больший вес голоса при выборе лучшего варианта
- перевода.
- Модератор может все, что может переводчик, а также может назначать роли пользователей (исключение: не может понизить роль владельца и других модераторов), редактировать информацию о проекте, приглашать пользователей стать участниками проекта.
- Владелец может все, что могут редактор и модератор, но также может загружать и удалять разделы, понижать роли модераторов, менять права доступа к проекту, менять видимость проекта, менять статус проекта и другие параметры.
- Все перечисленные роли могут быть настроены владельцем, а отдельному пользователю можно выдать дополнительные права. Пользователь может иметь несколько ролей одновременно (например, быть и редактором, и модератором).

Зарегистрированный пользователь может подать заявку на присоединение к публичному проекту. Владелец и модераторы проекта могут одобрить или отклонить заявку; если заявка была одобрена, пользователь становится участником проекта. Владелец и модераторы проекта могут пригласить пользователя в проект. Пользователь может принять или отклонить приглашение; если пользователь принял приглашение, он становится участником проекта.

Сервис также предоставляет доступ для удобного поиска существующих проектов. Для этого проектам присваиваются такие параметры как язык оригинала, язык перевода, тип (книга, субтитры, программа...), жанр (жанры книг, жанры фильмов и сериалов, типы

программ) и популярность (например, количество посещений). Сервис имеет раздел поиска проектов, в котором пользователь может фильтровать проекты по названным параметрам или искать по названию на языке оригинала и языке перевода.

Пользователь также имеет свою собственную страницу на сайте. На этой странице пользователь может указать информацию о себе (например, уровень знания языков и область знаний/интересов), аватар (изображение, отображаемое рядом с именем пользователя), настроить свои контактные данные, имя пользователя и пароль, менять параметры пользователя, удалить свой аккаунт. Другие пользователи видят на странице этого пользователя информацию, которая поможет им выбрать приглашаемого переводчика или определить, стоит ли одобрить его заявку на вступление. Эта информация содержит имя пользователя, его аватар, его описание себя и проекты, в которых этот пользователь участвует.

1.1.3 Требования к разделам, доступным пользователям на сайте

1. Панель навигации
   (a) Панель навигации должна быть доступна в шапке каждой страницы веб-сервиса, кроме текстового редактора.
   (b) Панель навигации предоставляет пользователю возможность переходить на следующие страницы: Главная страница, Список проектов пользователя, Личный кабинет, Поиск проектов, Авторизация, Регистрация.
2. Список проектов пользователя
   (a) Список проектов пользователя предоставляет пользователю список проектов, в которых он является участником или владельцем, краткую информацию о них и роль пользователя в этом проекте.
   (b) Список проектов пользователя должен позволять пользователю перейти на страницу выбранного проекта.
   (c) Со страницы списка проектов можно перейти на страницу создания проекта.
   (d) На странице списка проектов пользователя есть список приглашений в проекты, которые пользователь может принять или отклонить.
3. Страница регистрации

(a) Пользователь вводит электронную почту, логин и пароль. При нажатии на кнопку "Зарегистрироваться"создается новая учетная запись.

4. Страница авторизации

   (a) Пользователь вводит логин, пароль и авторизуется.

   (b) Пользователь может сменить пароль, если он его забыл.

5. Главная страница

   (a) Главная страница должна предоставлять возможность пользователю сервиса видеть свои последние проекты и ведущиеся на данный момент популярные переводы.

   (b) Главная страница должна предоставлять возможность осуществлять быстрый поиск проектов по названию или ID.

   (c) Главная страница должна предоставлять новому пользователю возможность получить информацию о веб-сервисе.

   (d) С главной страницы можно перейти на страницу создания проекта.

6. Личный кабинет

   (a) Личный кабинет должен предоставлять возможность пользователю редактировать информацию о своём профиле, такую как отображаемое имя (nickname), аватар и информацию о себе.

   (b) Личный кабинет должен предоставлять возможность пользователю редактировать для себя настройки интерфейса сервиса (внешний вид сайта).

   (c) Личный кабинет должен предоставлять возможность пользователю изменять и восстанавливать пароль от аккаунта в случае потери.

   (d) В личном кабинете указаны все проекты, в которых участвует пользователь.

   (e) Из личного кабинета можно перейти на страницу создания проекта.

7. Страница проекта

   (a) На странице проекта для удобства содержимое разделено на вкладки.

   (b) На вкладке "Проект"пользователю должна быть доступна информацию о проекте: название, обложка, описание, список разделов.

(c) На вкладке "Проект"владелец проекта может загружать разделы при помощи файлов, в т. ч. с различным форматированием, таким как JSON или CSV.

(d) На вкладке "Проект"пользователь, не являющийся участником, может нажать на кнопку "Присоединиться к проекту".

(e) На вкладке "Проект"участник проекта может зайти в редактор отрывков.

(f) На вкладке "Участники"участникам проекта доступен список участников, модераторы и владелец могут удалять и приглашать участников.

(g) На вкладке "Настройки"модераторы и владелец проекта могут менять информацию о проекте, владелец может удалить свой проект.

8. Меню управления проектом

(a) Модератор должен иметь возможность изменять такие редактируемые данные как информация о проекте, название, обложка, разделы проекта, участники проекта, язык оригинала, язык перевода и параметры проекта.

(b) Меню проекта должно предоставлять возможность имеющим права участникам скачивать полученный перевод в запрашиваемом формате (JSON, CSV...) и с заданными фильтрами (лучший перевод, перевод конкретного пользователя).

(c) Меню проекта должно предоставлять возможность имеющим права участникам приглашать новых пользователей участвовать в данном проекте.

(d) Модератор может менять роли участников.

9. Текстовый редактор

(a) Текстовый редактор должен предоставлять возможность переводчику добавлять новые переводы данных в разделе отрывки.

(b) Текстовый редактор должен предоставлять возможность пользователю-редактору изменять и удалять отрывки других участников.

(c) Участник проекта может голосовать за отрывки других участников.

10. Поиск проектов

(a) Поиск проектов должен позволять фильтровать проекты по тем или иным признакам (популярность, жанр, язык оригинала и перевода), а также выделять отдельно доступные для вступления проекты.

(b) Поиск проектов должен предоставлять возможность найти нужный проект по названию или ID.

11. Страница создания проектов

(a) Страница создания проектов должна позволять пользователю создать новый проект.

(b) Пользователь должен иметь возможность указать следующую информацию о проекте: Название проекта, Уникальная ссылка, Язык оригинала, Язык перевода, Обложка проекта, Доступ к проекту.

1.1.4 Модели требований

Модель предметной области

Объекты предметной области

- Неавторизованный пользователь — посетитель сайта, не прошедший этап авторизации.
- Авторизованный пользователь — посетитель сайта, прошедший регистрацию.
- Главная страница — страница сайта, которая содержит недавние и популярные проекты.
- Проект — совокупность пользователей, участвующих в переводе некоторого произведения, и набора отрывков текста, которые относятся к этому произведению.
- Участник — пользователь, имеющий доступ к работе над текстом в проекте.
- Текст — содержимое для перевода, загружается на страницу проекта.
- Отрывок — небольшая часть текста, доступная для перевода по отдельности.
- Свойства и информация о проекте — параметры проекта, которые может просмотреть любой пользователь, редактируются владельцем.
- Вариант перевода — относятся к отрывкам, за них можно голосовать и редактировать в соответствии с ролью.
- Популярные проекты — проекты с самым большим количеством просмотров.

- Недавние проекты — проекты, упорядоченные по дате создания, начиная с новых.
- Модуль регистрации — модуль, отвечающий за создание новых учетных записей для неавторизованных пользователей.
- Модуль авторизации — модуль, дающий неавторизованным пользователям возможность войти в свою учетную запись.
- Окно смены пароля — модуль, позволяющий восстановить пароль пользователю.
- Личный кабинет — страница с информацией о пользователе: имя, логин, описание, а также все проекты, в которых он участвует.
- Настройки интерфейса — смена цвета темы, шрифта.
- Роль проекта: переводчик — имеет возможность добавлять варианты перевода отрывков, голосовать за лучший вариант перевода.
- Роль проекта: редактор — может редактировать существующие варианты перевода и имеет больший вес голоса при выборе лучшего варианта перевода.
- Роль проекта: модератор — может назначать роли пользователей (исключение: не может понизить роль владельца и других модераторов), редактировать информацию о проекте, приглашать пользователей стать участниками проекта.
- Роль проекта: владелец — может загружать и удалять разделы, понижать роли модераторов, менять права доступа к проекту, менять видимость проекта, менять статус проекта и другие параметры.

Далее представлены диаграммы процессов предметной области: навигация по вебсайту и структура проекта.

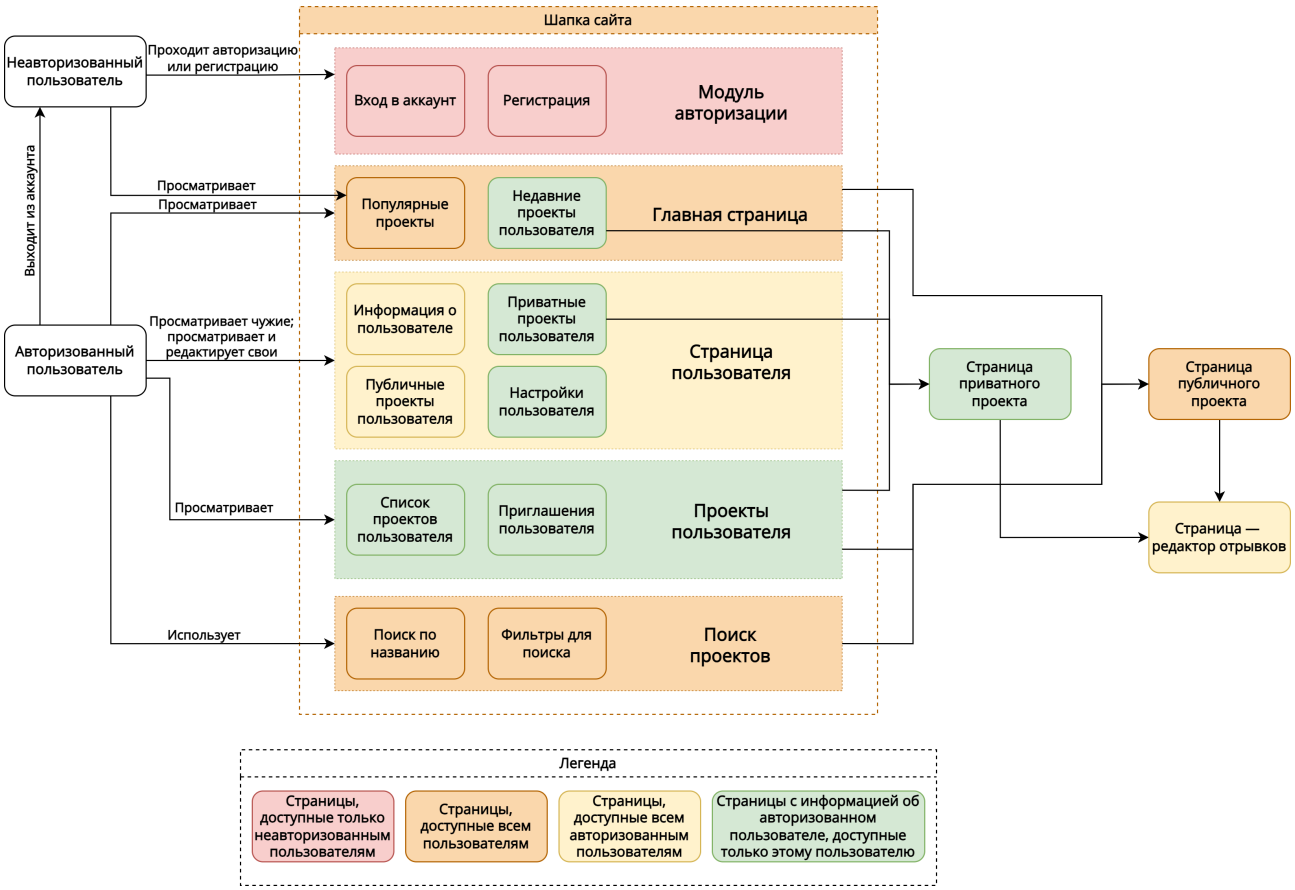

Рис. 1: Диаграмма процессов предметной области: навигация по веб-сайту.

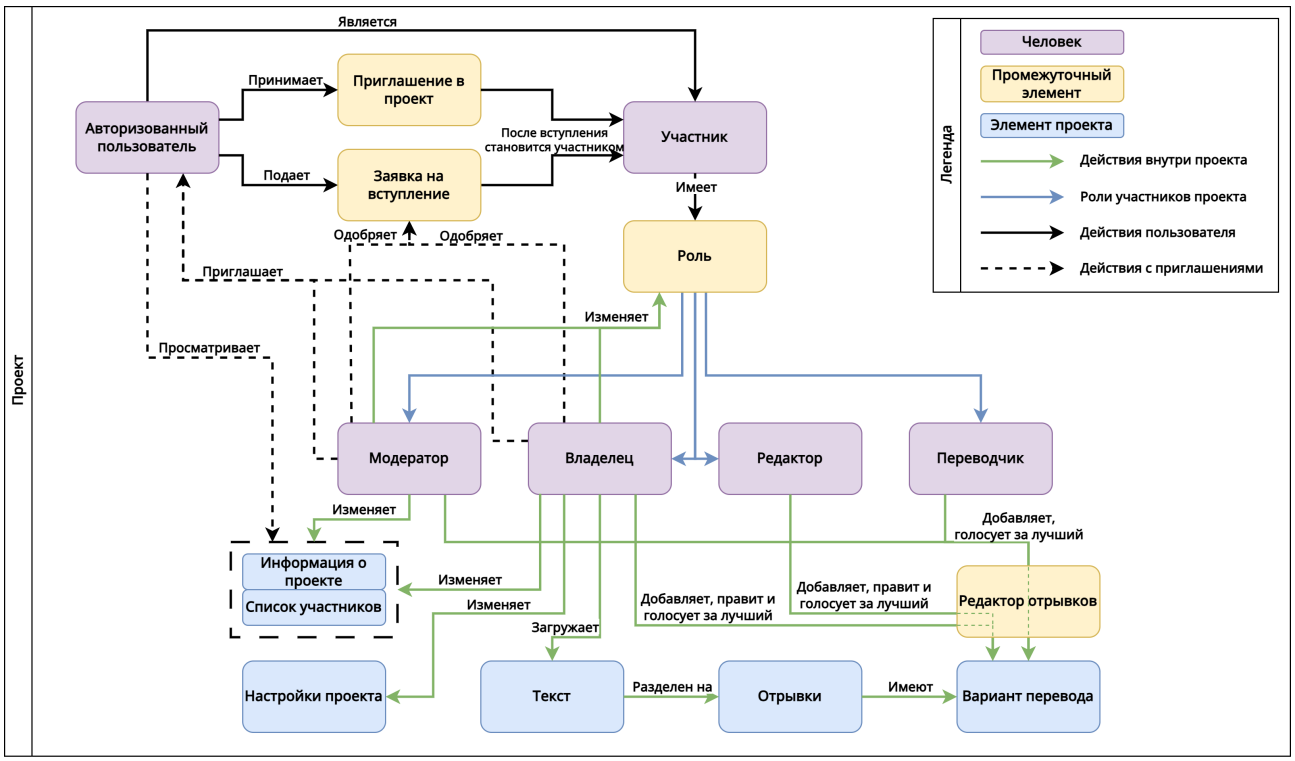

Рис. 2: Диаграмма процессов предметной области: структура проекта.

Модель предметной области

Список пользователей

- Неавторизованный пользователь.
- Авторизованный пользователь.

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.

Интересы пользователя

- Неавторизованный пользователь:
  - Доступная информация о назначении сайта и его использовании другими пользователями.
  - Информирование о необходимости в авторизации.
  - Удобный интерфейс для авторизации и регистрации.
- Авторизованный пользователь:
  - Удобные средства для поиска, создания проектов, возможности к ним присоединиться.
  - Удобные средства для работы в проекте (см. модель участников проекта).

Модель участников проекта

- Авторизованные пользователи, присоединившиеся к проекту, получают одну из ролей проекта. Все участники проекта имеют права переводчика, отличная от переводчика роль добавляет участнику другие права.

Список ролей

- Переводчик
- Редактор
- Модератор
- Владелец

Права роли

Переводчик:

- Может добавлять и редактировать свои варианты перевода отрывков.
- Может голосовать за лучший вариант отрывка.

Редактор:

- Имеет все права переводчика.
- Может редактировать чужие варианты перевода.

Модератор:

- Имеет все права переводчика.
- Может назначать роли участников (не может изменить роли модераторов и владельца).
- Может приглашать новых участников, одобрять заявки на вступление в проект.
- Может редактировать информацию о проекте.

Владелец:

- Имеет все права переводчика, редактора и модератора.
- Может создавать и удалять разделы.
- Может изменять роли модераторов.
- Может менять статус проекта и его видимость.

Интересы роли

Переводчик:

- Удобный интерфейс для просмотра текста и вариантов перевода.
- Простая система для голосования за лучший перевод.
- Простой и удобный интерфейс для добавления вариантов перевода.

Редактор:

- Интересы роли переводчика.
- Простой и удобный интерфейс для добавления и изменения вариантов перевода.

Модератор:

- Интересы роли переводчика.
- Функциональный интерфейс для изменения ролей.
- Возможность быстро принять и пригласить новых участников.

Владелец:

- Интересы ролей переводчика, редактора и модератора.
- Возможность создавать новые разделы.
- Интерфейс для работы со свойствами проекта.

1.1.5 Высокоуровневая архитектура

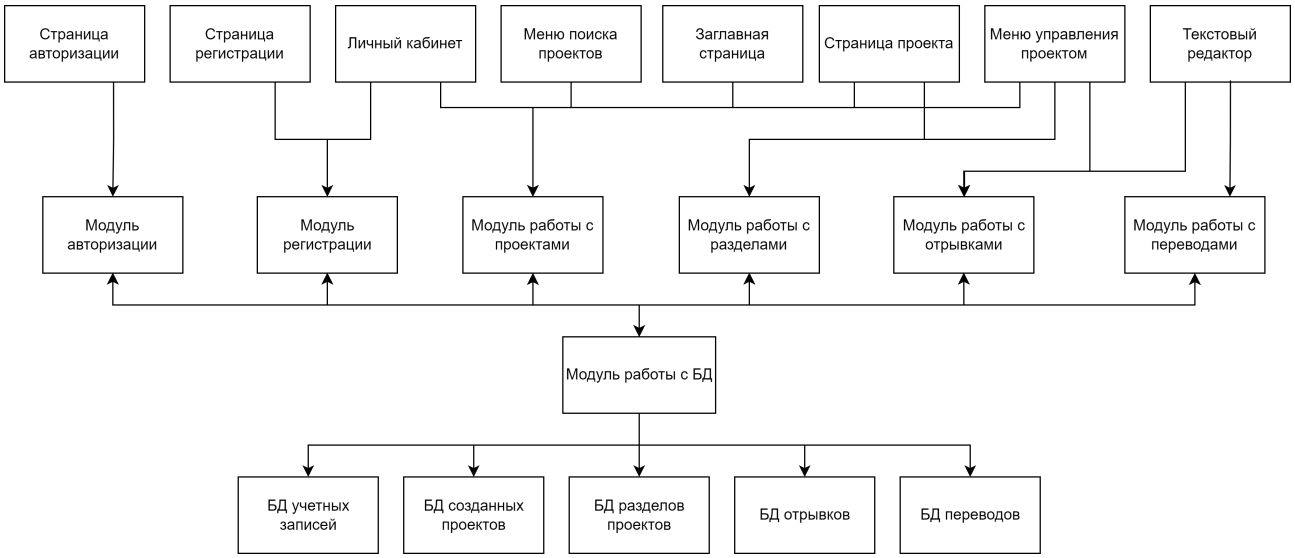

Рис. 3: Диаграмма высокоуровневой архитектуры.

1.2 Вывод главы

После подробного анализа требований к разрабатываемому сервису необходимо улучшить проекты веб-страниц и реализовать клиентскую часть в соответствии с требованиями и моделями предметной области. Также необходимо спроектировать и реализовать текстовый редактор, который не был спроектирован прежде.

2 Описание инструментов и технологий

2.1 Выбор инструментов для реализации клиентской части

HTML (HyperText Markup Language) — язык разметки, используемый для структурирования и отображения веб-страницы и ее контента. Например, контент может быть структурирован внутри множества параграфов, маркированных списков или с использованием изображений и таблиц данных. HTML не является языком программирования, и используется для того, чтобы сообщить браузеру, как необходимо отображать открытую веб-страницу. HTML-разметка позволяет создавать сколько угодно простые или сложные дизайны пользовательских интерфейсов для любых веб-сайтов. [2]

CSS (Cascading Style Sheets) — язык таблицы стилей, используемый для применения стилей к элементам в документах HTML. CSS позволяет применять стили к элементам выборочно, используя классы, теги, идентификаторы и так далее. [1]

Для реализации клиентской части можно использовать несколько доступных инструментов.

2.1.1 HTML+JavaScript без использования фреймворков

Преимущества:

+ нет необходимости в изучении дополнительных фреймворков;
+ большая свобода в реализации, поскольку нет ограничений в виде возможностей фреймворка.

Недостатки:

- низкая скорость реализации;
- значительная трудность реализации;
- более низкое качество итогового продукта в силу отсутствия опыта разработки.

Вывод: реализация клиентской части сервиса без использования фреймворков позволит начать разработку без изучения соответствующих инструментов, однако сам этап разработки займет больше времени и усилий, будет труднее в дальнейшем поддержании

кода, а итоговый продукт будет ниже качеством, поскольку реализуется неопытными разработчиками.

2.1.2 React

React — JavaScript-библиотека для разработки пользовательских интерфейсов. Основана на использовании индивидуальных компонентов, языке разметки JSX. Компонент — независимый модуль исходного кода, предназначенный для повторного использования. Преимущества:

+ низкий порог вхождения;
+ компонентно-ориентированная библиотека с большой базой готовых компонентов;
+ рендеринг на стороне сервера, а не клиента;
+ нисходящий поток данных;
+ виртуальная бъектная модель документа.

Недостатки:

- необходимо перевести HTML разметку в JSX разметку;
- плохая документация;
- front-end библиотека, отвечающая только за представление, т. е. back-end должен быть реализован с помощью другого инструмента.

Вывод: реализация клиентской части сервиса с помощью React будет быстрой, однако возникнет сложность при переводе HTML-шаблонов в JSX-разметку, а back-end разработка должна будет произведена с помощью другого инструмента.

2.1.3 Vue.js

Vue.js — JavaScript фреймворк, который легко надстраивается над существующим HTML и JavaScript кодом. Хорошо подходит для создания адаптируемых пользовательских интерфейсов и сложных одностраничных приложений. Преимущества:

+ позволяет оптимизировать HTML-разметку с помощью компонентов;

+ легкая интеграция в существующую инфраструктуру;

+ легкость в масштабировании;

+ малый размер самого фреймворка.

Недостатки:

- нехватка документации на английском языке;

- нехватка ресурсов — очень малый процент разработчиков используют Vue.js для крупных проектов;

- проблемы при интеграции в большие проекты, при этом отсутствует опыт возможных решений подобной проблемы.

Вывод: Vue.js в силу своего небольшого размера позволяет быстро реализовать сложное одностраничное приложение или улучшить существующий проект, однако имеет очень маленькую аудиторию и недостаточную документацию на английском языке (фреймворк разработан китайскими компаниями), что повышает вероятность появления тяжело разрешаемых проблем при масштабировании.

2.1.4 Vue.js

Angular — платформа для разработки веб-приложений, основанная на языке TypeScript. Преимущества:

+ очень подробная документация;

+ двусторонняя привязка данных — минимальный риск ошибок приложения;

+ ускоренная компиляция;

+ модель Model-View-ViewModel, пользованяющая разработчикам раздельно работать в одном разделе приложения с использованием одного и того же набора данных.

Недостатки:

- необходимость изучения сложного синтаксиса TypeScript;

- проблемы с миграцией при переходе между версиями фреймворка — поддерживать приложение на протяжении многих лет будет проблематично.

Вывод: Angular является очень популярной и эффективной платформой, однако требует изучения TypeScript, что значительно замедлит процесс разработки, поскольку уже реализованные шаблоны и веб-API станут непригодны для разработки.

2.2 Вывод

Исходя из преимуществ и недостатков рассмотренных инструментов, библиотек и фреймворков, было принято решение использовать React.js, поскольку эта библиотека на данный момент является наиболее популярной, то есть имеет низкий порог вхождения и подробную документацию, что значительно ускорит процесс разработки. Помимо этого React.js поддерживает Bootstrap в виде библиотеки компонент React Bootstrap, что позволит легко и быстро адаптировать уже существующие шаблоны для разработки клиентской части. Ориентированность React на front-end составляющую не является существенным недостатком для этого проекта, поскольку back-end разработка проводится с помощью платформы Node.js и будет легко интегрирована с front-end составляющей с помощтю веб-API.

3 Разработка клиентской части приложения

Листинги кода front-end составляющей доступны по ссылке:
https://github.com/ipaingo/Desman-Translate-frontend

Для разработки клиентской части было необходимо выполнить несколько этапов.

3.1 Улучшение и перевод существующих HTML-шаблонов

Поскольку React.js использует язык разметки JSX, а React Bootstrap, который крайне удобно использовать вместе с React.js предлагает не библиотеку CSS-файлов, а готовых React-компонентов, было необходимо перевести HTML-шаблоны в JSX-разметку. Для перевода между языками разметок использовались онлайн-инструменты, преобразование полученных блоков разметки для использования с React Bootstrap было выполнено вручную, поскольку являлось простой и быстрой в выполнении задачей.

Рис. 4: Пример изменения разметки после перевода.

3.2 Установка React.js

Для разработки с помощью React.js необходимо установить его локально на свой рабочий компьютер. Установка готового приложения на сервер является частью back-end разработки проекта, поэтому не будет рассмотрена в этой работе.

1. Установить Node.js на рабочий компьютер.

2. Открыть терминал командной строки в директории, где должен быть установлен React.

3. Ввести команду `npm install react react-dom`.

4. При необходимости использовать некоторую библиотеку для работы с React необходимо в терминале ввести команду `npm install <название библиотеки>`.

Установщик npm формирует файл зависимостей, поэтому при установке готового проекта на React на другой рабочий компьютер необходимо в терминале выполнить команду `npm install`. Тогда все необходимые библиотеки будут установлены автоматически в соответствии с файлом зависимостей.

3.3 Запросы к API и отрисовка страницы в зависимости от ответа

Для взаимодействия с API были реализованы несколько функций, совершающих fetch-запросы и возвращающих ответы в зависимости от введенных параметров.

```
export async function fetchSomeAPI(link, method = "GET", body = {}) {
    let options = { method: method, }
    if (method == 'POST' || method == 'PATCH') {
        options.body = JSON.stringify(body)
        options.headers = { 'Content-Type': 'application/json; charset=UTF-8
            ', }
    }

    const response = await fetch(link, options)
    if (!response.ok) {
        throw { status: response.status, errors: JSON.parse(await response.
            text() || "{}")?.errors }
    }
    try {
        let data = await response.json()
        return data
    } catch (err) {

    }
}
```

Листинг 1: Вспомогательная функция, использующаяся для осуществления fetch-запросов с параметрами.

Для аутентификации пользователя используется компонент AuthProvider, функция ReAuth которого совершает запрос к API с попыткой получить пользователя, возвращает информацию о том, что пользователь не прошел аутентификацию, если проверка была неуспешной, возвращает информацию о пользователе в случае успеха.

```
function AuthProvider(props) {
    const [user, setUser] = useState(undefined);
    const [gotUser, setGotUser] = useState(false);

    async function ReAuth() {
        console.log("okay, try to get me")
        let rsp = await fetch("/api/whoami",
        {
            method: "GET",
            headers: {
                'Content-Type': 'application/json; charset=UTF-8',
            },
        })
        console.log(rsp)
        if (!rsp.ok) {
            console.log("what")
            setUser(undefined)
            return
        }

        try {
            rsp = await rsp.json()
            console.log("user")
            console.log(rsp)
        } catch (err) {
            rsp = null
            console.log("no user :(")
        }

        setUser(rsp)
    }

    const value = {
        user,
        ReAuth,
        gotUser,
```

```
37      };
38
39
40      useEffect(() => {
41          ReAuth();
42      }, [])
43
44      useEffect(() => {
45          setGotUser(user !== undefined)
46      }, [user])
47
48      return <AuthContext.Provider value={value} {...props} />;
49  }
```

Листинг 2: Компонент AuthProvider, необходимый для аутентификации пользователя.

Функции с fetch-запросами вызываются в коде для получения необходимой для отрисовки информации.

```
1   function Home() {
2
3       const { user } = useContext(AuthContext);
4
5       const [recentProjects, setRecentProjects] = useState([]);
6
7       const [popularProjects, setPopularProjects] = useState([]);
8
9       async function GetRecentProjects() {
10          if (!user)
11              return
12
13          try {
14              let projects = (await fetchUser(user.id, true)).projects
15              projects.sort((a, b) => new Date(b.created_at) - new Date(a.
                    created_at))
16              setRecentProjects(projects)
17          } catch (err) {
18              console.log(err)
19          }
20      }
21
22
```

```
    async function GetPopularProjects() {
        try {
            let projects = await fetchProjects(4)
            projects.sort((a, b) => new Date(b.created_at) - new Date(a.
                created_at))
            setPopularProjects(projects)
        } catch (err) {
            console.log(err)
        }
    }


    useEffect(() => {
        GetRecentProjects();
    }, [user]);

    useEffect(() => {
        GetPopularProjects();
    }, []);


    let navigate = useNavigate();
    const routeChange = () => {
        let path = '/create';
        navigate(path);
    }
<...>
```

Листинг 3: Код на JavaScript, получающий необходимую информацию для отрисовки главной страницы.

React Bootsrap предоставляет разработчику большую библиотеку готовых компонент, которые были использованы при написании кода. Далее приведен пример JSX-разметки со вставками кода на JavaScript, отрисовывающий главную страницу веб-сайта.

```
<Header />
<Container className="mt-5">
    <Row>
        <Col sm={6} className="text-left pe-5">
            <h2 className="mb-4">Recent projects:</h2>
            {recentProjects.map((project, i) =>
                <Container className="text-left pb-2" key={project.id}>
```

```
8                   <Stack direction="horizontal" gap="10" className="border
                        rounded p-3 mt-1">
9                       <img
10                          width={60}
11                          height={60}
12                          src={placeholder}
13                          alt="thumbnail"
14                      />
15                      <Container className="text-left text-break">
16                          <Link to={"/projects/" + project.handle}
                                className="link-primary">
17                              {project.name}
18                          </Link>
19                          <br /> {project.description}
20                      </Container>
21                  </Stack>
22              </Container>
23          )}
24
25          <h2 className="my-4">Trending projects:</h2>
26          {popularProjects.map((project, i) =>
27              <Container className="text-left pb-2" key={project.id}>
28                  <Stack direction="horizontal" gap="10" className="border
                        rounded p-3 mt-1">
29                      <img
30                          width={60}
31                          height={60}
32                          src={placeholder}
33                          alt="thumbnail"/>
34                      <Container className="text-left text-break">
35                          <Link to={"/projects/" + project.handle}
                                className="link-primary">
36                              {project.name}
37                          </Link>
38                          <br /> {project.description}
39                      </Container>
40                  </Stack>
41              </Container>
42          )}
43
```

```
44
45          </Col>
46          <Col className="border-top border-start rounded py-1 mt-3 ps-3">
47              <h5 className="py-2 border-bottom" >
48                  What is Desman Translate?
49              </h5>
50              <p>
51                  <...>
52              </p>
53              <h5 className="py-2 border-bottom">
54                  How does this work?
55              </h5>
56              <p>
57                  <...>
58              </p>
59              <p>
60                  <...>
61              </p>
62              <p>Have a lot of fun...</p>
63              {user &&
64                  <Button variant="primary"
65                      onClick={routeChange}>
66                      Create a project
67                  </Button>
68              }
69          </Col>
70      </Row>
71  </Container>
72  <Footer />
```

Листинг 4: Код на JSX, отрисовывающий главную страницу.

Результат работы кода, описанного выше:

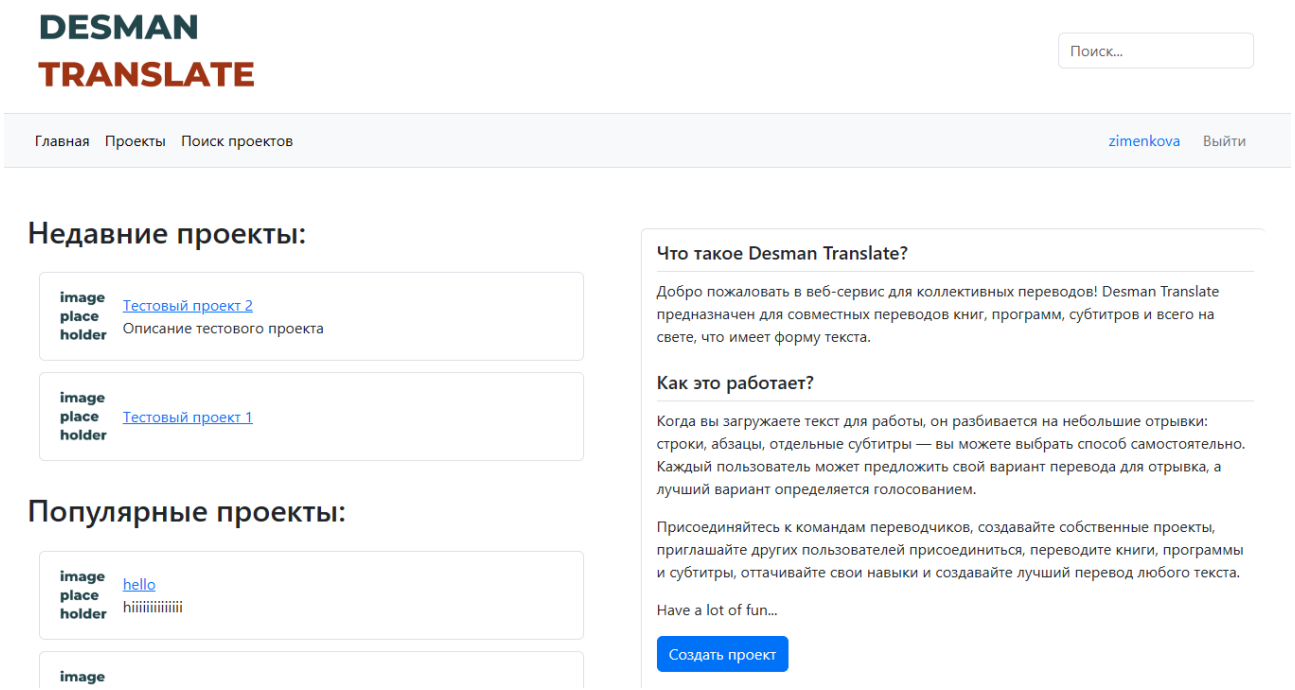

Рис. 5: Главная страница сайта.

3.4 Сборка и установка сервиса

3.4.1 Установка сервера

1. Перейти по ссылке на страницу репозитория на сайте GitHub.com: https://github.com/Nepr
2. Установить базу данных MongoD: https://www.mongodb.com/try/download/community.
3. Скачать и разархивировать проект или клонировать git-репозиторий на сервере или в локальном хранилище.
4. Настроить конфигурационный файл .env: `JWT_SECRET` - параметр для шифровки cookie-файлов.
5. Открыть терминал, перейти в папку скачанного проекта, убедиться, что пользователь имеет установщик пакетов npm и node. Выполнить команду `npm install`.
6. В терминале выполнить команду `node index.js` - сервер запущен.

3.4.2 Установка клиента

1. Перейти по ссылке на страницу репозитория в GutHub.com: https://github.com/ipaingo/Des Translate-frontend.

2. Скачать и разархивировать проект или клонировать git-репозиторий на сервере или в локальном хранилище.

3. Открыть терминал, перейти в папку скачанного проекта, убедиться, что пользователь имеет установщик пакетов npm и node. 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.

Заключение

Список литературы

1. MDN. Основы CSS.
   https://developer.mozilla.org/ru/docs/learn/getting_started_with_the_web/css_basics, 2023.

2. MDN. Основы HTML.
   https://developer.mozilla.org/ru/docs/learn/getting_started_with_the_web/html_basics, 2022.

Приложение