Петрозаводский государственный университет

Институт математики и информационных технологий

**Техническое задание**

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

**«Desman Translate»**

Заказчик

Колчин М.А. 22306

Исполнитель

Зименкова С.Э. 22305

# Термины, используемые в техническом задании

API – сервер с набором функций и описанием их применения.

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

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. Заявка прилагается к техническому заданию в виде docx файла.

* 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.

* + 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: Google Chrome, Yandex, Opera, Safari, Mozilla, Edge.

* + 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.

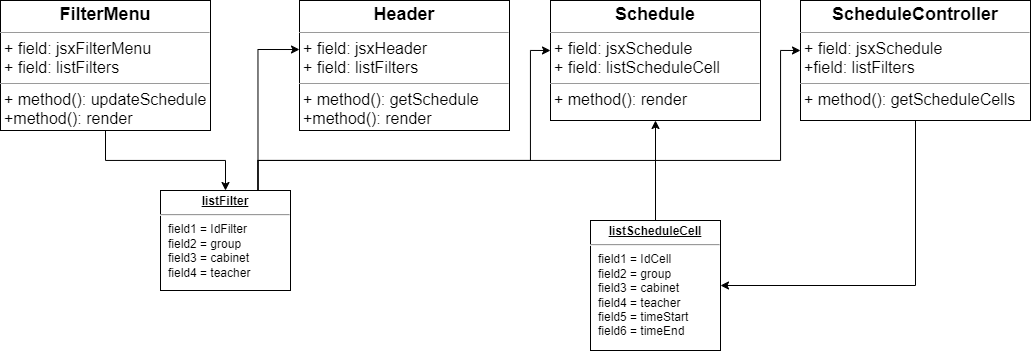
* + - 1. Права пользователей

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



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

Классы FilterMenu, Header, Schedule имеют jsx поля (jsxFilterMenu, jsxHeader, jsxSchedule соответственно), в которых хранится html-содержимое для меню фильтров, шапки (заголовка) и расписания. Функция render в каждом из полей возвращает эту jsx-переменную, и она отрисовывается на основном файле.

FilterMenu. Содержит метод updateSchedule(), который обновляет глобальную переменную listFilter – это массив, в котором хранятся фильтры для каждого расписания, которые выбрал пользователь, вызывается после нажатия пользователем кнопки «Принять».

Header. Метод getSchedule() обновляет заголовки фильтров для быстрого отображения выбранных пользователем фильтров в шапке сайта. Он вызывается каждый раз, когда обновляется глобальная переменная listFilters – данные из этой же переменной передаются на вход getSchedule().

Schedule. Через render() отрисовываются ячейки расписания, данные для которых взяты из глобальной переменной listScheduleCell.

ScheduleConroller. Метод getScheduleCells() принимает на вход переменную listFilter и вызывается каждый раз, когда listFilter обновляется. Метод обрабатывает список фильтров, обращается к API расписания и формирует список занятий listScheduleCell.

* + - 1. 2.4 По каждой подсистеме – перечень функций, задач, доп. требований.

|  |  |  |  |
| --- | --- | --- | --- |
| Подсистема | Перечень функций | Кому доступна | Изменяемые объекты |
| Шапка | 1.Вызвать окно фильтров.  2.Увидеть текущие фильтры.  3.Увидеть название сайта. | Все пользователи | Не изменяет никакие данные |
| Меню фильтров | 1.Добавить новое расписание  2.Изменить фильтры в расписании.  3.Применить фильтры. | Все пользователи | ListFilter – список фильтров |
| Контроллер расписания | 1.Обработать фильтры. | Разработчик сайта. | listScheduleCell – список занятий для фильтров. |
| Тело расписания | 1.Увидеть выбранное расписание. | Все пользователи | Не изменяет никакие данные |

* + 1. 2.5 Перечень сценариев по реализации наиболее важных и сложных процессов в системе – диаграммы пригодности и использования

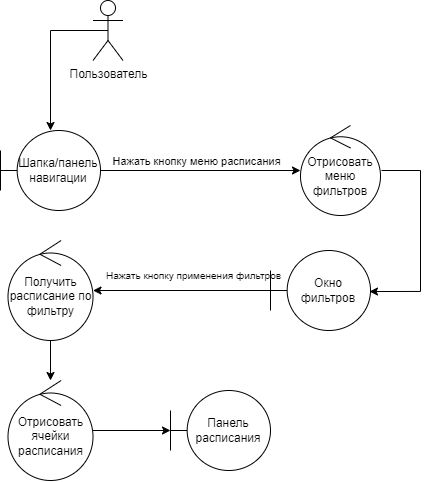


Рисунок 1 Диаграмма пригодности

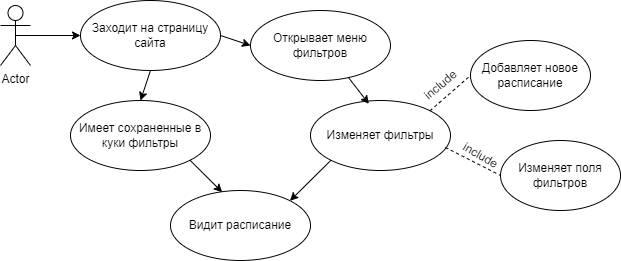


Рисунок 2 Диаграмма использования

* + 1. 2.6 Структура сайта, системы, дерево разделов и подразделов

Web-ресурс состоит лишь из одной страницы, с которой взаимодействует пользователь и основывается на работе с виджетами, расположенными на начальной странице.

Общая структура сайта выглядит следующим образом:

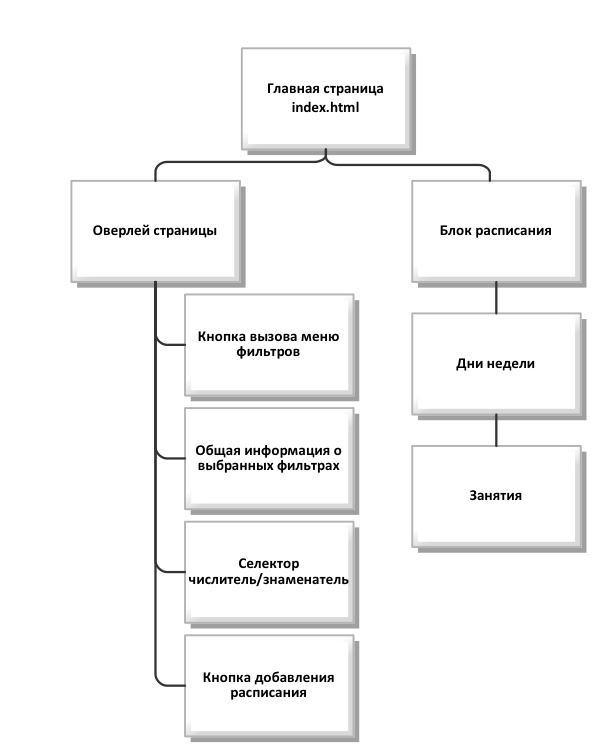


Рисунок 3 Общая структура страницы

1. Оверлей страницы – полоса вверху экрана, на которой находятся кнопка фильтров, информация о выбранных фильтрах в виде текста и выбор числителя или знаменателя.
   1. Селектор выбора числителя/знаменателя расположен в оверлее, а не в окне фильтров из-за того, что это самое часто используемое свойство.
   2. Кнопка добавления расписания – добавляет новые опции в меню фильтров и выделяет место на экране под расписание (новое расписание располагается слева от уже существующих и занимает столько же места по горизонтали, то есть все расписания меняют ширину).
2. Блок расписания более подробно описан на рисунке 4.

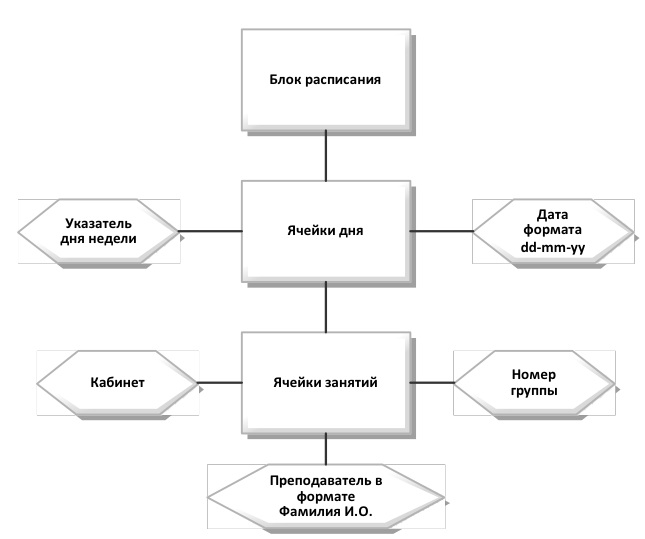


Рисунок 4 Структура блока расписания

1. Расписание состоит из шести ячеек дней, в верху которых указываются дата и название дня недели.
2. В ячейках дней находятся ячейки занятий. В них расположена текстовая информация о кабинете, в котором проводится занятие, номере группы и преподавателе.

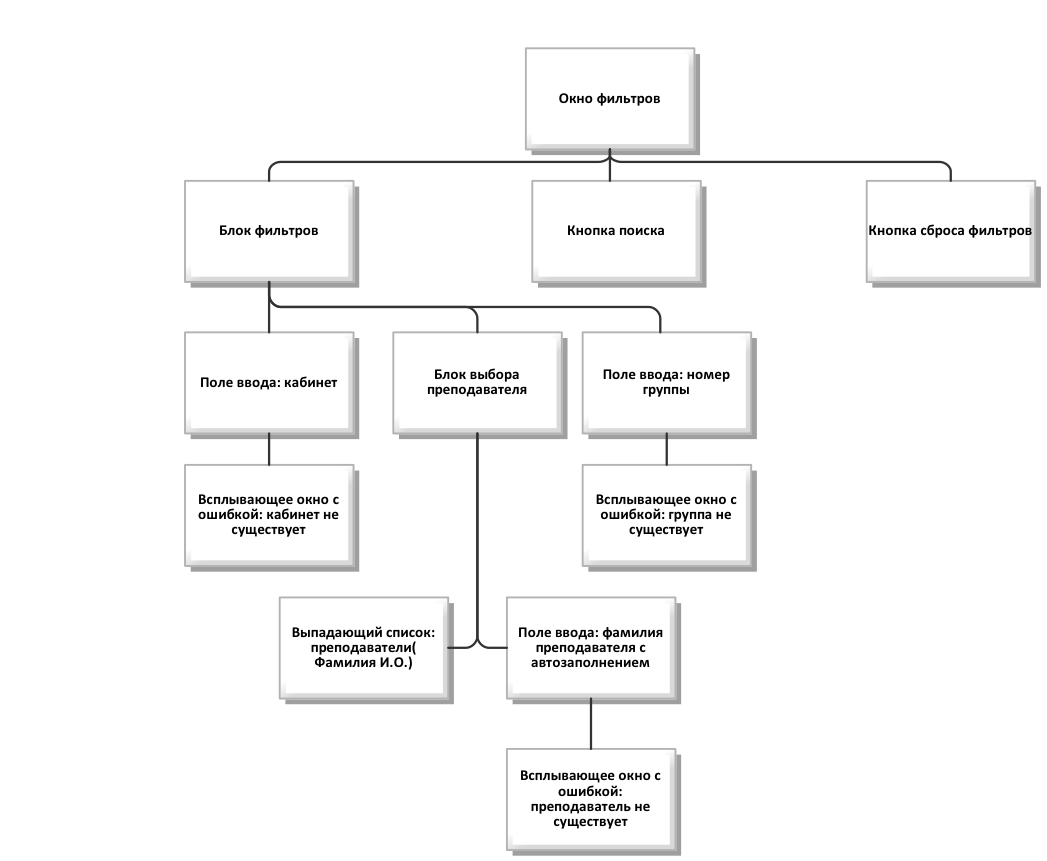


Рисунок 5 Структура окна фильтров.

Окно фильтров вызывается из оверлея нажатием кнопки фильтров. После нажатия появляется поле в правой части страницы, занимающее половину экрана. В нем находятся следующие элементы:

1. Блок фильтров. Через него пользователь выбирает желаемые опции для фильтрации расписания: кабинет, преподаватель, номер группы.
2. Кнопка поиска. Как только пользователь ввел нужные фильтры, он нажимает эту кнопку, после чего окно фильтров сворачивается и в блоке расписания появляется информация.

Если какие-то текстовые фильтры введены неправильно, появляется всплывающее сообщение с ошибкой, в которой указана причина неуспеха.

1. Кнопка сброса фильтров. Очищает все выбранные пункты фильтров.

Из окна фильтров можно выйти через нажатие кнопки поиска или кнопки фильтров на оверлее (через которую он открыл это окно).

* 1. Перечень прототипов и экранных схем – прототипы

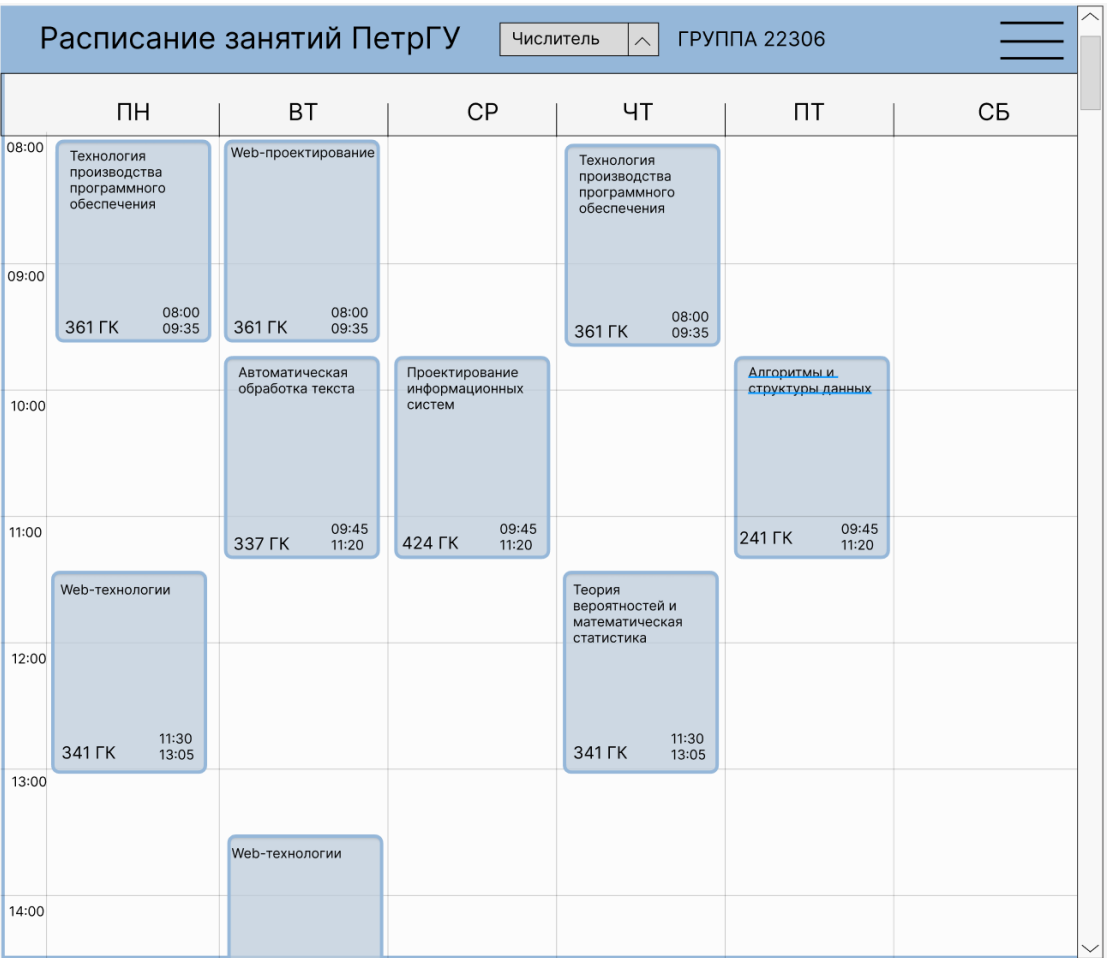


Рисунок 6 Пример расписания

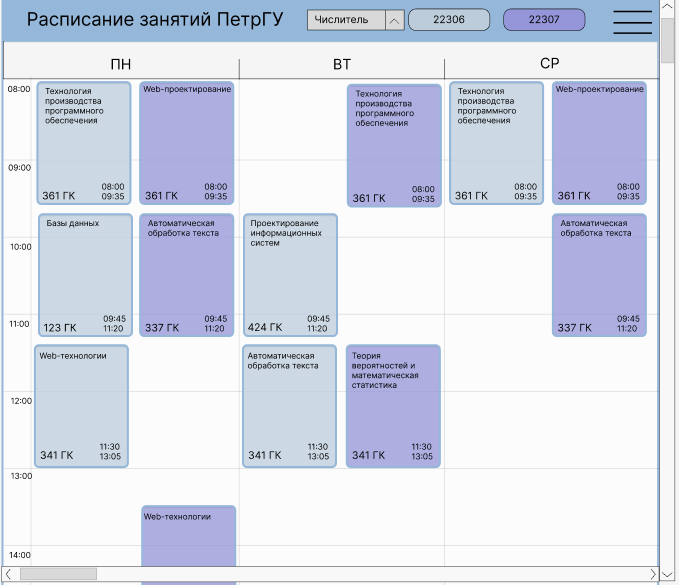


Рисунок 7 Пример двух расписаний с фильтрами по группам

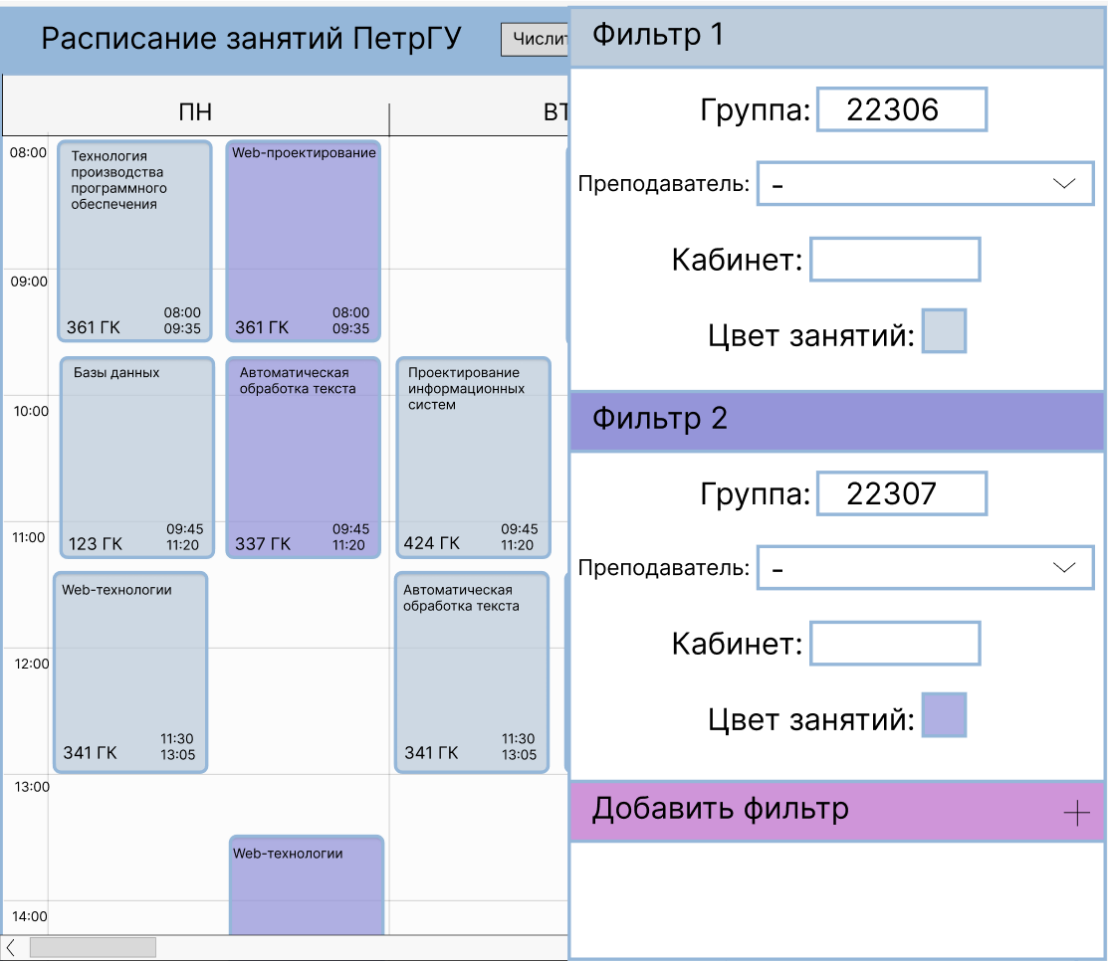


Рисунок 8 Окно фильтров

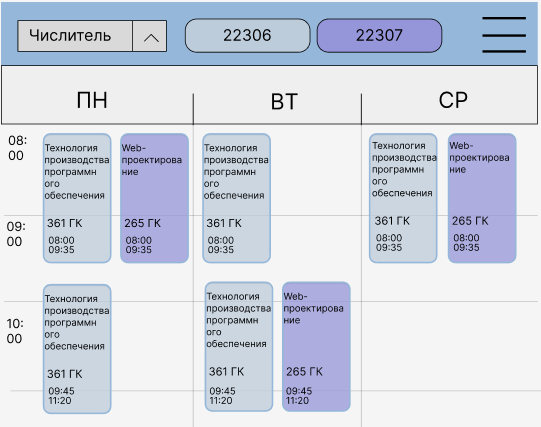
Дизайн сайта для экрана в 680px

Рисунок 9 Меню с 2 выбранными расписаниями

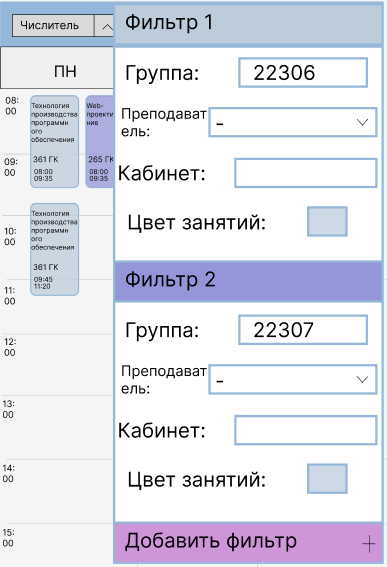


Рисунок 10 Открытое меню фильтров

Дизайн сайта для экрана в 320px

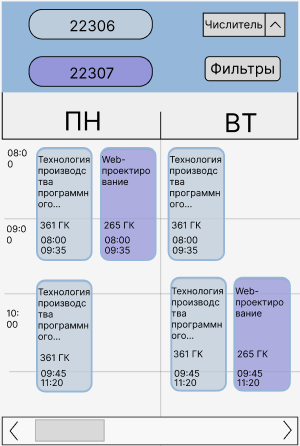


Рисунок 11 Меню с 2 выбранными фильтрами



Рисунок 12 Меню фильтров

* 1. Требования к видам обеспечения

Сайт может быть запущен на локальном сервере. В случае установки сайта на внешнем сервере, требования по мощности будут низкими.

* 1. Программное обеспечение

Сайт, развернутый на сервере, доступен любому пользователю, чье устройство обладает поддержкой одного из браузеров: Google Chrome, Yandex, Opera, Safari, FireFox, Edge.

* 1. Техническое обеспечение

Персональный компьютер пользователя.

* 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-е 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-е.

* 1. Математическое обеспечение

Математические методы и алгоритмы решения задач не используются на сайте.

* 1. Организационное обеспечение

Сайт предназначен для использования одним человеком. Но сайтом могут пользоваться одновременно несколько человек в одно время на разных устройствах.

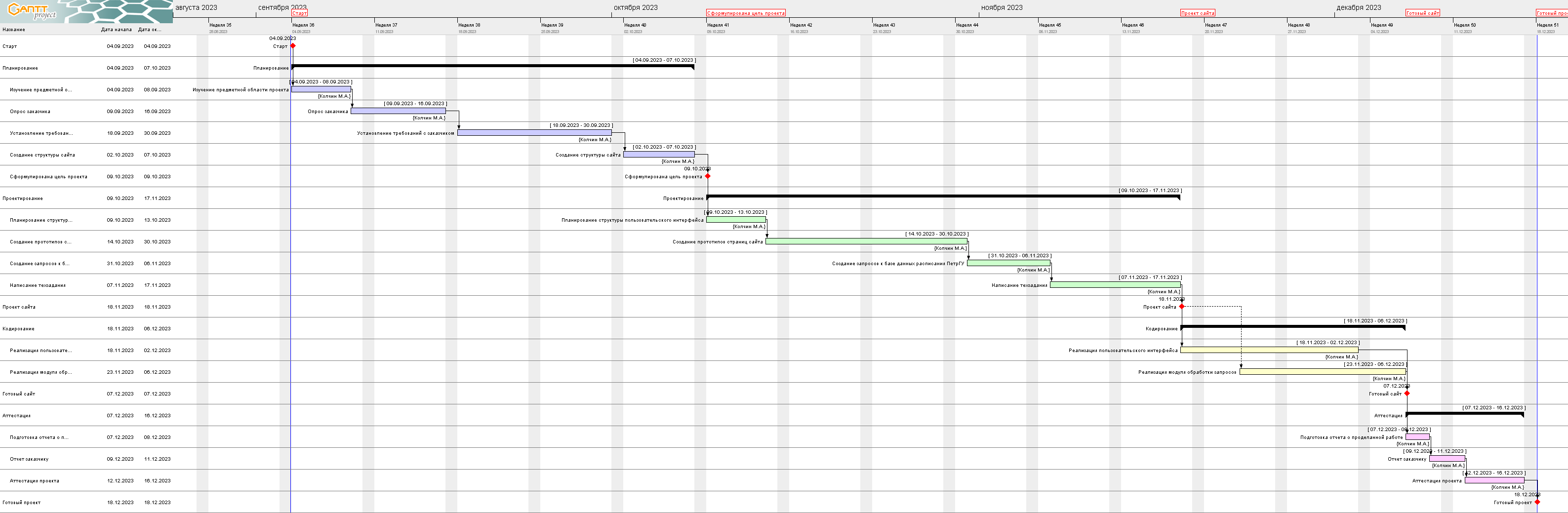
* 1. Информационное обеспечение (контент, наполнение)

Сайт предоставляет расписание университета ПетрГУ, отображение которого можно настраивать с помощью фильтров.

# 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 | 30 дней | Создан макет сайта. |
| Написание запросов к API расписания ПетрГУ | 20 дней | Реализован модуль кода, реализующий запросы по фильтрам к базе данным. |
| Реализация пользовательского интерфейса | 15 дней | Страница приобрела конечный внешний вид. |
| Компиляция приложения | 5 дней | Страница 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.

Сайт должен быть скомпилирован на сервере – приложение будет представлять статичную страницу, обращающуюся к API, но сама записей в БД не делает.

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

9.1 Документация по установке

* 1. Перейти по ссылке на страницу репозитория на сайте GitHub.com по ссылке (<https://github.com/KolchinMMM/PetrSU_Schedule>)
  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 (иначе – установить по ссылке(<https://www.npmjs.com/package/download)>)
  4. Если необходимо запустить сайт на локальном сервере – написать команду в терминале «npm start». Сайт будем 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.

9.2 Документация по администрированию

Сайт не требует администрирования.

9.3 Документация по использованию

1. На главной странице находятся шапка сайта с кнопкой меню фильтров, описанием принятых фильтров, названием сайта, и окно расписания, в котором будут отображаться ячейки с занятиями по выбранным фильтрам.
2. Чтобы попасть в меню фильтров, необходимо нажать на кнопку в виде трех горизонтальных черточек в шапке сайта – справа выедет меню фильтров и займет пол экрана.
3. В меню фильтров можно добавить новое расписание, нажав на кнопку «+» в поле «Добавить фильтр». Все фильтры расписания будут располагаться в этом меню друг за другом.
4. В каждом расписании в меню фильтров можно изменить настройки фильтров: название группы, имя преподавателя, кабинет, цвет ячеек расписания.
5. Чтобы принять измененные фильтры, необходимо нажать кнопку «Принять» - меню фильтров закроется, а в окне расписания отобразится обновленная информация.
6. В окне расписаний ячейки занятий распределены по дням недели и часам.

9.4 Программная документация (для доработки)

Программа состоит из следующих основных файлов:

1. public/index.js. Здесь находится тело сайта, в которое будет записываться весь интерфейс. Вносить изменения в этот файл не рекомендуется.
2. src/App.js. В находится главная сайта, в которой объявляются главные переменные и вызываются методы отрисовки остальных классов.
3. src/pages/Header.js. В файле находится класс Header. Он отвечает за отрисовку шапки сайта.
4. src/pages/FilterMenu.js. В файле находится класс FilterMenu. Он отвечает за отрисовку меню фильтров и вызывает контроллеры из класса ScheduleController, передавая им данные из полей ввода.
5. src/pages/Schedule.js. В файле находится класс Schedule. Он отвечает за отрисовку окна расписания.
6. src/ScheduleController.js. В файле находится класс ScheduleController.

# Реквизиты и подписи сторон

Заказчик Зименкова С. Э

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_

Исполнитель Колчин М.А.

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_

Дата: