МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«Новосибирский национальный исследовательский государственный университет»

(Новосибирский государственный университет, НГУ)

Структурное подразделение Новосибирского государственного университета –

Высший колледж информатики Университета (ВКИ НГУ)

КАФЕДРА ИНФОРМАТИКИ

**СОЗДАНИЕ ВЕБ-СЕРВИСА ДЛЯ МОНИТОРИНГА ИСПОЛНЕНИЯ ДОЛГОСРОЧНЫХ ПЛАНОВ РАБОТ СЕРВИСНЫХ БРИГАД ПРИ РАЗРАБОТКЕ НЕФТЯНЫХ МЕСТОРОЖДЕНИЙ**

Квалификация программист

|  |  |
| --- | --- |
| Руководитель  руководитель отдела ПО ООО «ННТЦ» | Сердюк К.С.  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2020 г. |
| Студент 4 курса  603 с | Чусовлянов И.В.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2020 г. |

Новосибирск

2020

СОДЕРЖАНИЕ

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ УСЛОВНЫХ ОБОЗНАЧЕНИЙ 4](#_Toc42046788)

[ВВЕДЕНИЕ 5](#_Toc42046789)

[1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 7](#_Toc42046790)

[2 ПОСТАНОВКА ЗАДАЧИ 9](#_Toc42046791)

[3 АНАЛОГИ 10](#_Toc42046792)

[3.1 Библиотека JSGantt 11](#_Toc42046793)

[3.2 Библиотека Google Charts 12](#_Toc42046794)

[3.3 Вывод 13](#_Toc42046795)

[4 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ПРОГРАММНОМУ СРЕДСТВУ 14](#_Toc42046796)

[4.1 Возможности для работы с календарным планом 14](#_Toc42046797)

[5 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ПРОГРАММНОМУ СРЕДСТВУ 21](#_Toc42046798)

[5.1 Требования к программному обеспечению 21](#_Toc42046799)

[5.2 Требования к аппаратному обеспечению 21](#_Toc42046800)

[5.3 Требования к надёжности 21](#_Toc42046801)

[5.4 Требования к безопасности 22](#_Toc42046802)

[6 ХАРАКТЕРИСТИКА ВЫБРАННЫХ ПРОГРАММНЫХ СРЕД И СРЕДСТВ 23](#_Toc42046803)

[7 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ 26](#_Toc42046804)

[7.1 Входные и выходные данные для сервиса загрузки 26](#_Toc42046805)

[7.2 Входные данные для веб-сервиса 28](#_Toc42046806)

[8 СХЕМА ФУНКЦИОНИРОВАНИЯ ПРОГРАММНОГО СРЕДСТВА 30](#_Toc42046807)

[8.1 Схема функционирования веб-сервиса 30](#_Toc42046808)

[8.2 Схема функционирования загрузчика данных 34](#_Toc42046809)

[9 ОПТИМИЗАЦИЯ ПРОГРАММНОГО СРЕДСТВА 35](#_Toc42046810)

[10 ОТЛАДКА И ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА 36](#_Toc42046811)

[10.1 Функциональное тестирование 36](#_Toc42046812)

[10.2 Тестирование удобства использования 36](#_Toc42046813)

[10.3 Тестирование совместимости 38](#_Toc42046814)

[10.4 Тестирование производительности 38](#_Toc42046815)

[11 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 39](#_Toc42046816)

[ЗАКЛЮЧЕНИЕ 42](#_Toc42046817)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 43](#_Toc42046818)

[ПРИЛОЖЕНИЕ А 45](#_Toc42046819)

[ПРИЛОЖЕНИЕ Б 47](#_Toc42046820)

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

В работе используются следующие сокращения:

* LTDS (Long Term Drilling Sequence) – название веб-сервиса для составления долгосрочного плана бурения нефтегазовых скважин.
* Диаграмма Ганта – тип столбчатых диаграмм (гистограмм), который используется для иллюстрации плана, графика работ по проекту.
* Tooltip – интерактивный элемент, при наведении на который появляется подсказка с текстом.
* Angular – открытая и свободная платформа для разработки веб-приложений, написанная на языке TypeScript, разрабатываемая командой из компании Google [1].
* Material Design – стиль графического дизайна интерфейсов программного обеспечения и приложений, разработанный компанией Google [2].
* SVG – язык разметки масштабируемой векторной графики, созданный Консорциумом Всемирной паутины (W3C) [3].
* TypeScript – язык программирования, представленный Microsoft в 2012 году и позиционируемый как средство разработки веб-приложений [4].
* JetBrains WebStorm – интегрированная среда разработки на JavaScript, CSS и HTML от компании JetBrains [5].
* Node.js – программная платформа, основанная на движке V8, превращающая JavaScript из узкоспециализированного языка в язык общего назначения [6].
* CSS – формальный язык описания внешнего вида документа, написанного с использованием языка разметки.
* HTML – стандартизированный язык разметки документов во Всемирной паутине.
* GraphQL – язык запросов с открытым исходным кодом, разработанный Facebook. Он создавался как более эффективная альтернатива REST для разработки и использования программных интерфейсов приложений [13].

ВВЕДЕНИЕ

Тема: Создание веб-сервиса для мониторинга долгосрочных планов работ сервисных бригад при разработке нефтяных месторождений.

Нефтяные и газовые добывающие компании, как правило в своем активе имеют несколько тысяч скважин и постоянно ведут работы по бурению новых и ремонту старых скважин. Работы выполняются десятками различных бригад и требуют дорогостоящего специализированного оборудования, из-за использования которого могут возникать технические ограничения. Порядок выполнения работ существенно влияет на объем добычи углеводородов и то, как нужно изменять инфраструктуру дочернего общества, чтобы обеспечить работу вновь введенных скважин. Таким образом, в зависимости от целей нефтегазовой компании возникают различные варианты параллельно отслеживаемых планов, отличающихся друг от друга десятками целевых показателей, таких как объём затрат, количество пробуренных метров, объём добычи и т.д. Сейчас составление планов выполняется вручную и занимает много времени. Так же над составлением плана работают несколько специалистов, такие как геолог, экономист, мастер по бурению, которые вносят правки и обмениваются вариантами по электронной почте. Автоматизация этого процесса позволила бы добиться высоких показателей оптимальности плана.

Для этой цели компанией ООО «ННТЦ» разрабатывается система нацелена на оптимизацию логистики проводимых работ по разработке нефтяных и газовых месторождений. Данная работа является ее частью, в рамках которой создается веб-сервис, позволяющий проводить мониторинг исполнения долгосрочных планов. Процесс мониторинга предполагает возможность сравнения плановых и фактических выполненных работ за определенную дату для каждой бригады и для каждой скважины на диаграмме Ганта и сравнение ключевых целевых показателей. Для достижения этой цели разрабатывается механизм загрузки данных на имеющийся сервер и мониторинг этих данных. Новизна веб-сервиса в том, что он позволяет быстро визуализировать сразу несколько вариантов планов, интегрировать аналитику и мониторинг.

Сервис предназначен для геологов, мастеров по бурению и экономистов.

Тема является актуальной, так как на фоне нестабильных цен на энергоресурсы нефтяным компаниям необходимо повышать свою эффективность. А оптимизация плана бурения дает быстрый и хорошо оцениваемый результат в отличие от многих других методов повышения эффективности.

Были проделаны следующие работы:

* Изучена новая предметная область бурения и освоения нефтяных скважин.
* Найдены и рассмотрены существующие библиотеки для визуализации планов Gantt, Google Charts.
* Составлены функциональные и нефункциональные требования.
* Изучены новые технологии и программные средства:
  1. Фреймворк Angular;
  2. Язык разметки векторной графики SVG;
  3. Стиль графического дизайна Material Design.
* Разработаны и реализованы: загрузчик, компонент для визуализации планов в виде расширенной диаграммы Ганта, одностраничное приложение для мониторинга.

Используемые программные средства, инструменты разработки и технологии:

* HTML;
* CSS;
* TypeScript;
* JetBrains WebStorm;
* Angular;
* Node.js;
* SVG;
* Material Design.

1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Основной задачей составления планов разработки нефтяных скважин, является планирование темпов роста добычи нефти в определенный период времени, а также ввод месторождений в разработку своевременно. Очередность и размещение строительства скважин должны планироваться так, чтобы буровые работы были максимально эффективным.

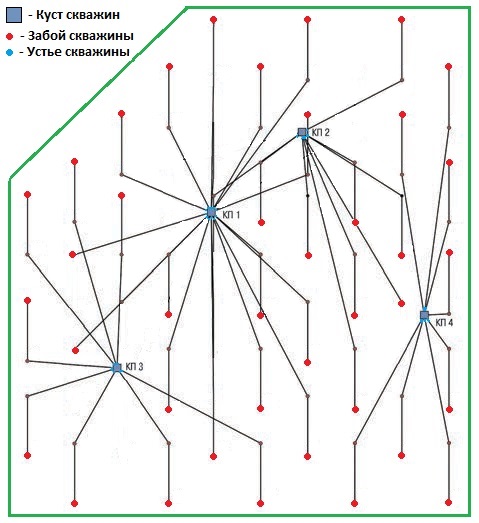


Рисунок 1 – Кустовое бурение

Бурение скважин происходит кустовым способом (рисунок 1), это когда скважины находятся на одной площадке, а их устья расположены на расстоянии в несколько метров друг от друга, и бурятся так, что их забои могут располагаться на расстоянии до нескольких десятков километров друг от друга. У такого бурения может возникнуть ряд ограничений, очередность бурения скважин должна соответствовать определенным требованиям, а переезд на соседние кусты занимает много времени. Эти ограничения усложняют задачу составления планов. После введения в эксплантацию скважины подключаются к единой сети трубопровода и нефти отправляется в необходимое место. Так как пропускная способность трубопровода ограничена, необходимо заранее обеспечить необходимую пропускную способность или отложить бурение куста, пока она не будет увеличена для пропуска необходимого количества нефти.

Именно поэтому необходимо точно знать количество нефти, которое будет добываться с куста и распределять бригады для бурения так, чтобы после введения куста в эксплуатацию не было перегрузок на трубопровод, а также чтобы бригады после выполнения бурения не простаивали. Возможность визуализировать планирование работ сервисных бригад и предоставляет веб-сервис «LTDS».

Продукт позволяет наглядно увидеть составленный план и ряд показателей для бригад в графическом виде, что дает возможность оценить его эффективность. Благодаря этому можно выбрать самый оптимальный план, тем самым снизить риск и повысить надежность планирования.

2 ПОСТАНОВКА ЗАДАЧИ

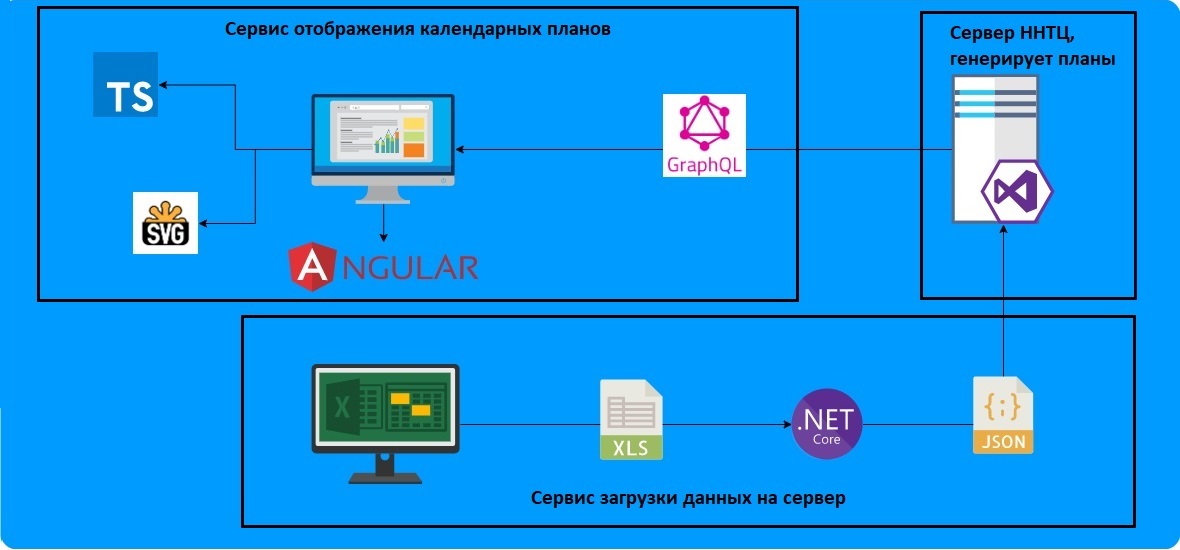


Рисунок 2 – Схема разработки

Задача разработать загрузчик данных на сервер, веб-сервис, который позволит производить мониторинг и анализ составленных планов (рисунок 2):

* Реализация основных функций сайта;

1. Отображение календарного графика для планов в разных представлениях.
2. Просмотр графиков показателей.

* Оформление, основанное на стиле Material Design.
* Адаптивная верстка страниц.
* Создание пользовательского интерфейса на фреймворке Angular.
* Тестирование сайта на совместимость и адаптивность с наиболее распространенными браузерами.

3 АНАЛОГИ

Важным этапом разработки является отображение календарного графика работ. Для этой цели необходимо использовать библиотеку для отображения диаграммы Ганта. Поэтому, как аналоги компоненты отображения мы будем рассматривать библиотеки для ее реализации.

Критериями выбора библиотек для рассмотрения являются:

* Отображения графика Ганта.
* Легкость по внедрению в проект.
* Возможность интерактивного взаимодействия.
* Возможность масштабирования и перемещения элементов.
* Отображение нескольких графиков на одной странице.
* Возможность коммерческого использования.

В результате анализа были найдены следующие библиотеки:

* JSGantt [7];
* Frappe Gantt [8];
* Google Charts [9];
* FusionCharts [10];
* AnyChart [11];
* jQueryGantt [12].

Все рассмотренные библиотеки можно разделить на две группы. Первая группа направлена только на реализацию диаграммы Ганта. А вторая группа включает в себя набор разнообразных графиков. Далее рассмотрим наиболее подходящие библиотеки из каждой группы.

## 3.1 Библиотека JSGantt

«JsGantt» входит в первую группу, она представлена на рисунке 3.

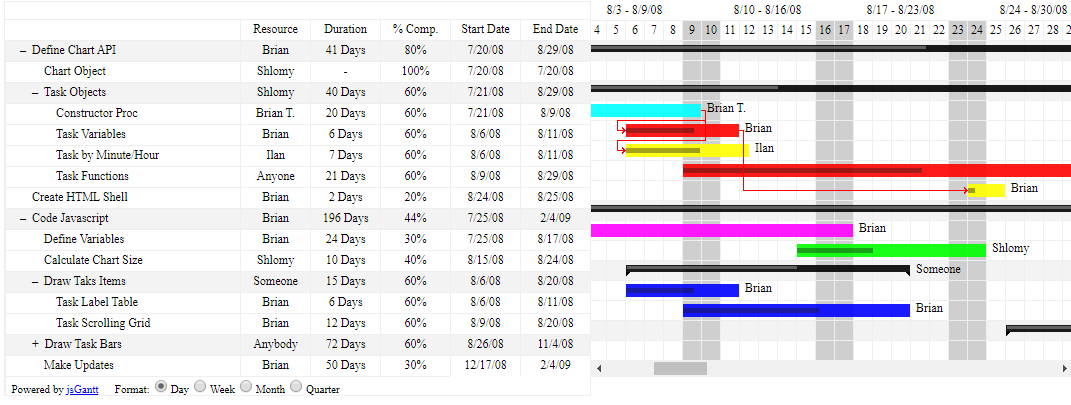


Рисунок 3 – Представление диаграммы Ганта в JsGantt

Плюсы:

* График Ганта;
* Обозначения элементов на оси Y представлены в виде древовидной структуры;
* Легко внедряется в проект;
* Открытый исходный код;
* Отображение нескольких графиков.

Минусы:

* Для отображения использует DOM, а именно table, что затрудняет реализацию масштабирования и перемещения элементов, либо делает её невозможной;
* Не совместима с другими библиотеками для реализации линейных графиков.

## 3.2 Библиотека Google Charts

«Google Charts» входит во вторую группу, она представлена на Рисунке 4.

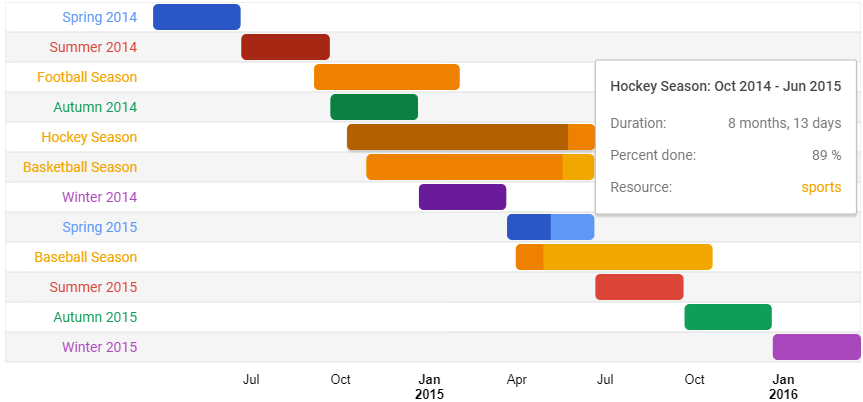


Рисунок 4 – Представление диаграммы Ганта в Google Charts

Плюсы:

* График Ганта;
* Легко внедряется в проект;
* Отображение подробной информации по элементам;
* Полноценная документация.

Минусы:

* Не работает без подключения к интернету;
* Нет возможности перемещать элементы;
* Бесплатная лицензия, но исходный код не открыт. Не позволяет размещать JS файлы Google на вашем сервере, поэтому может не подойти вам, если вы работаете с конфиденциальными данными.

## 3.3 Вывод

Таблица 1 – Плюсы и минусы использования сторонней библиотеки

| **Плюсы** | **Минусы** |
| --- | --- |
| Легко внедряется в проект | Сложная общая архитектура, поэтому тяжело разбираться в возникающих ошибках |
| Позволяет избежать многих непредвиденных проблем | Не может решить конкретную задачу самым оптимальным способом |
| Код оттестированный огромным количеством пользователей | При самостоятельном изменение библиотеки могут возникнуть проблемы с будущим обновлением версий |

Проанализировав плюсы и минусы использования сторонней библиотеки Таблица 1, было принято решение разрабатывать отображение календарного графика самостоятельно.

4 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ПРОГРАММНОМУ СРЕДСТВУ

В веб-сервисе существует следующие типы ролей:

* Пользователь.

Пользователь имеет возможность:

* Просматривать планы в разных представлениях (группировка по подрядчикам, кустам, месторождениям);
* Изменять масштаб и область просмотра календарного графика;
* Просматривать графики показателей;
* Просматривать работы в виде таблицы.

## 4.1 Возможности для работы с календарным планом

На рисунке 5 отображен календарный график.

### 

Рисунок 5 – Отображение календарного графика

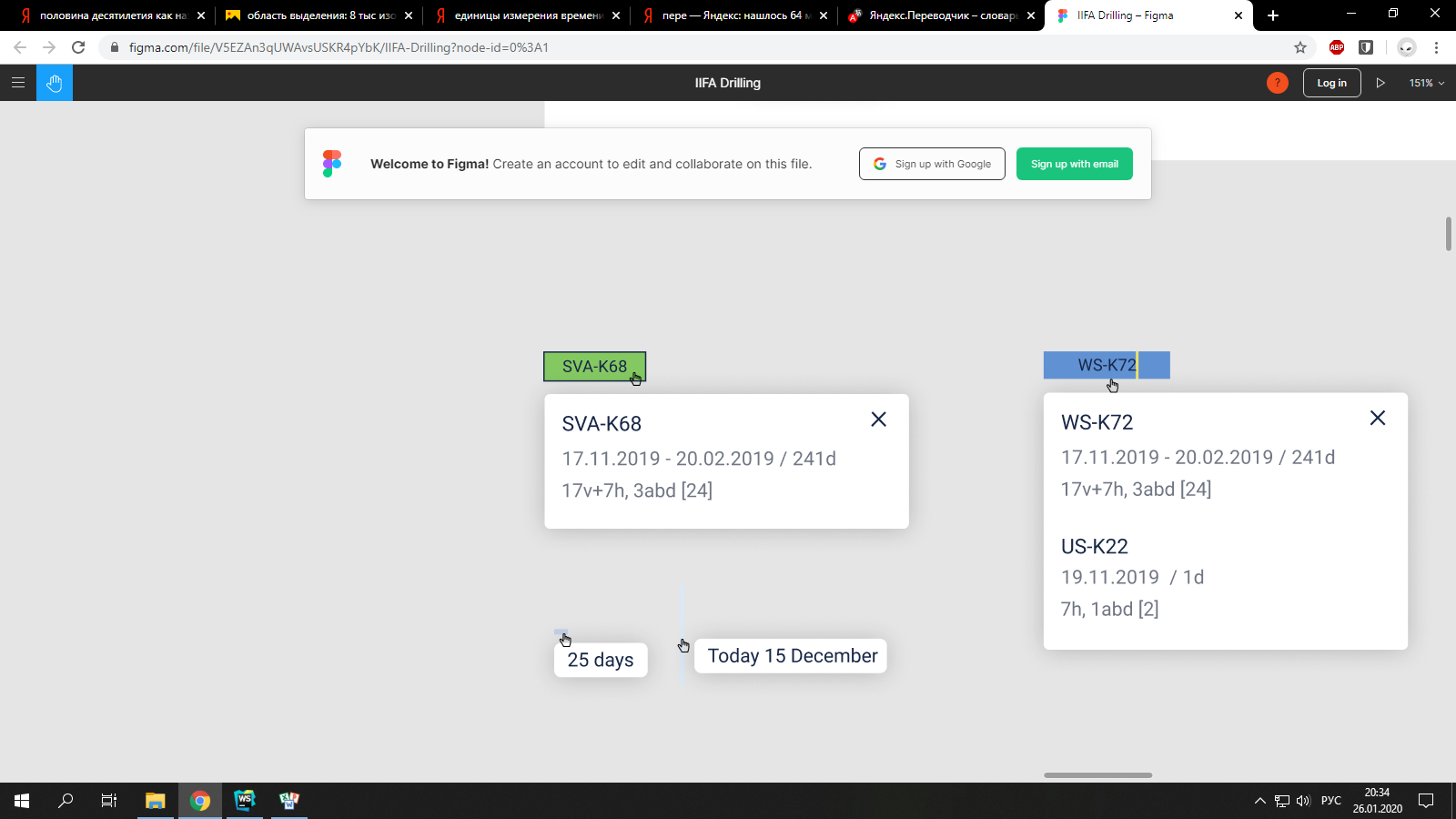


Рисунок 6 – Tooltip с информацией по кусту

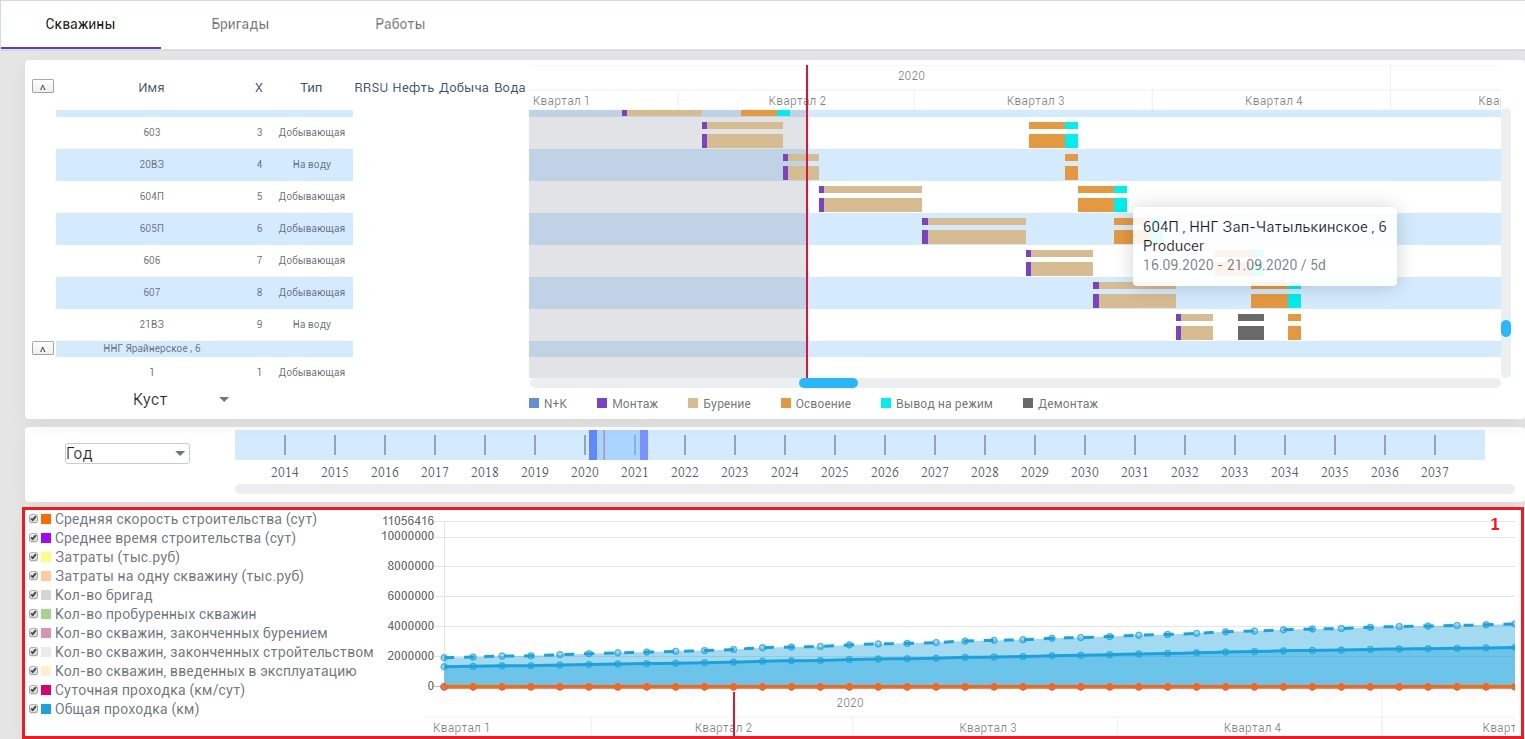
На графике отображаются следующие данные:

* По вертикали отображается список объектов. Объекты должны быть сгруппированы (рисунок 5 пункт 1).
* Блоками с названием на графике отображаются работы и их продолжительность по времени (рисунок 5 пункт 2). У каждого объекта отображается утвержденный план работ, его работы отображаются блоками, которые шире, чем у обычных планов. Текущая дата отображается красной вертикальной чертой.
* Работы выделяются определенным цветом в зависимости от типа, тип указывается в легенде (рисунок 5 пункт 3).
* Сверху должна отображаться шкала с датами согласно отметкам масштаба, (рисунок 5 пункт 4).
* При наведении на блок куста на графике должен отображаться tooltip (рисунок 6) с наименованием куста, датами начала и окончания работ в формате (дд.мм.гггг.) и продолжительностью работы.

Требования к функциональности:

* Система должна обеспечить переключение режимов отображения работ (рисунок 5 пункт 5).
* В режиме «Скважины» система должна отображать список объектов капитального строительства и работы, проводимые на этих объектах.
* В режиме «Бригады» система должна отображать список бригад и работы, проводимые этими бригадами на объектах капитального строительства.
* В режиме «Работы» система должна отображать работы и их параметры.
* Система должна отображать план работ в виде диаграммы Ганта в окне «Бригады», в окне «Скважины» и в виде текстовой таблицы в окне «Работы».
* Система на основе полученных данных должна автоматически отображать утвержденный план работ по ремонту и строительству капитальных объектов.

На рисунке 7 отображен график показателей.

Рисунок 7 – Отображение графика показателей

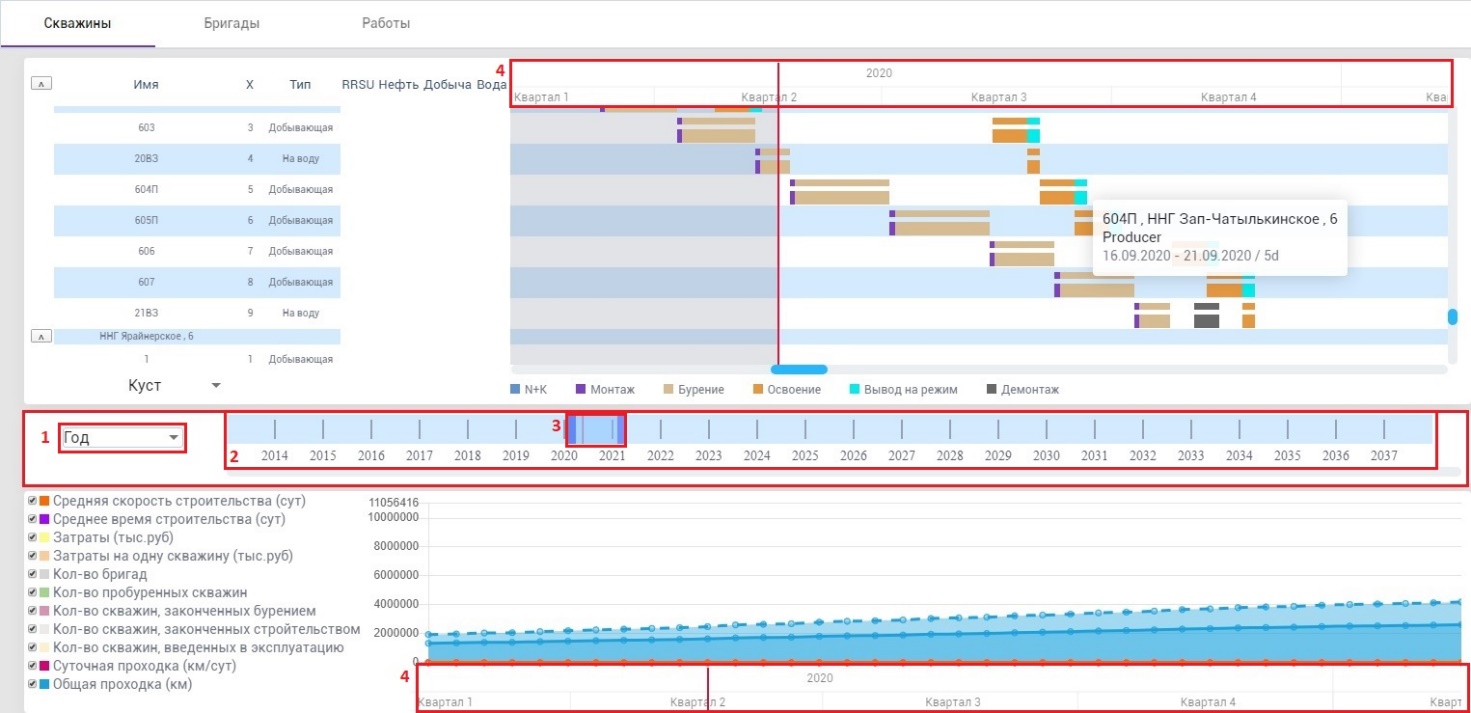
Необходимые показатели для отображения на графике (рисунок 7 пункт 1):

* Затраты (тыс. руб.);
* Затраты на одну скважину (тыс. руб.);
* Общая проходка (км);
* Суточная проходка (км/сутки);
* Среднее время строительства (сутки);
* Средняя скорость строительства (кол-во объектов/сутки);
* Количество бригад;
* Количество пробуренных скважин;
* Количество скважин, законченных бурением;
* Количество скважин, законченных строительством;
* Количество скважин, введенных в эксплуатацию.

Требования к функциональности:

* Система должна давать возможность пользователю включать и отключать интересующие его показатели.
* Система должна отображать на графике по вертикали значения показателей, по горизонтали отображать шкалу времени по фиксированным отметкам масштаба.
* Система должна отображать на графике показатели плана и утвержденного плана.
* Система должна давать возможность пользователю видеть информацию по показателям, при наведении мышкой на любое место на графике.

На рисунке 8 отображена настройка временного интервала и масштаба.

Рисунок 8 – Панель настройки временного интервала

Требования к функционалу:

* Система должна отображать в настройках временного интервала временной интервал, включающий все работы, которые есть в системе Рисунок 8 пункт 2.
* Система должна давать возможность пользователю задавать тип привязки курсора к временной шкале в настройках временного интервала. Может быть: по годам, по месяцам, по кварталам. Привязка границы ползунка на диапазоне временного интервала происходит по заданному критерию Рисунок 8 пункт 1.
* Система должна дублировать временную шкалу в календарном плане и на графике показателей данных Рисунок 8 пункт 4.
* Система должна при изменении масштаба в настройках временного интервала менять масштаб на временной шкале в календарном плане и временной шкале на графике показателей данных.
* Временная шкала в календарном плане и на графике показателей данных состоит из двух строк.
* Система должна давать возможность пользователю задать необходимый временной интервал, нажав на нужный на временном интервале, или перетащив ползунок на необходимый интервал времени.
* Система должна при настройке временного интервала отображать временную шкалу в календарном плане и на графике показателей данных в соответствии с таблицей 2.

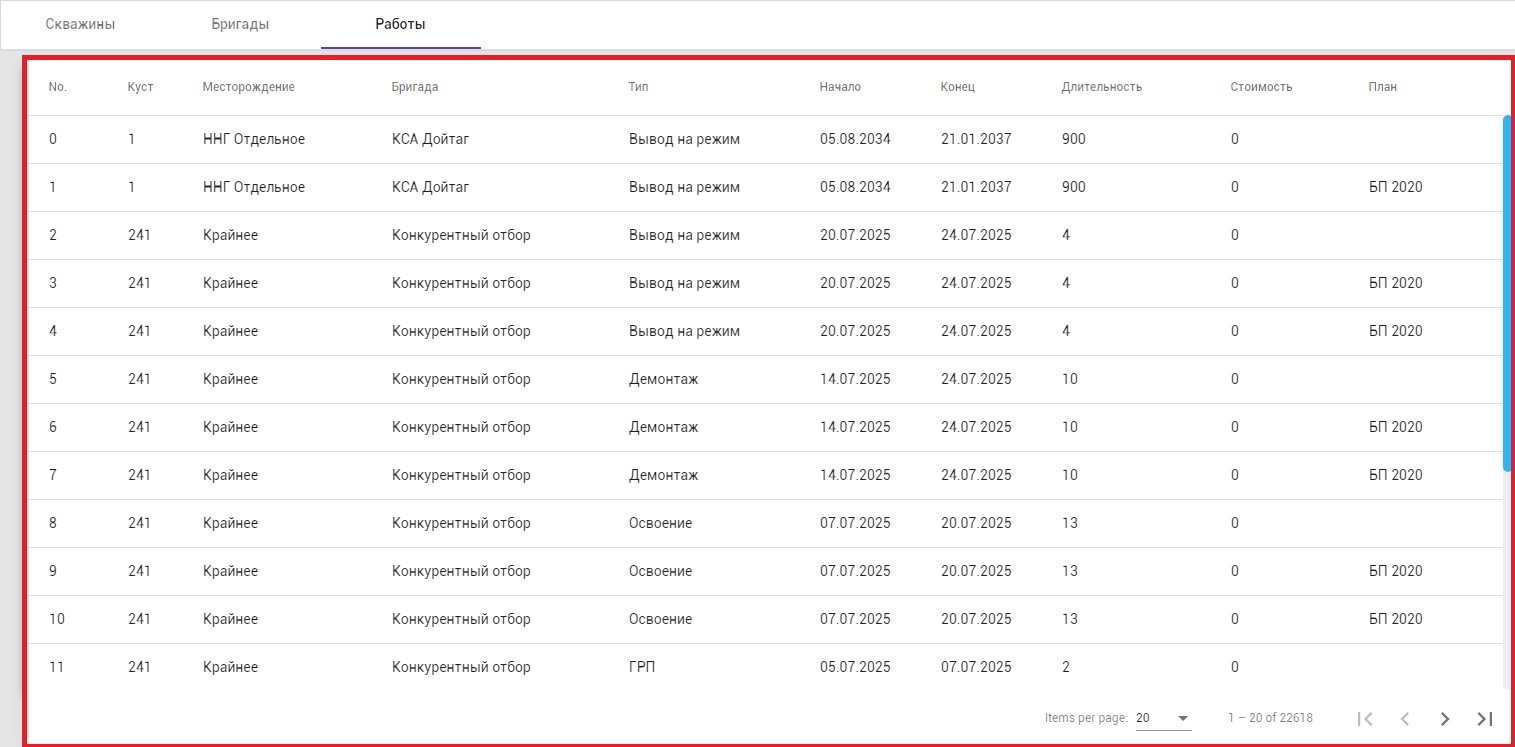
Таблица 2 – Фиксированные отметки масштаба

| Настройка временного интервала | **Отображение верхней строки шкалы** | **Отображение нижней строки шкалы** | **Комментарий** |
| --- | --- | --- | --- |
| От пяти лет и более | Год | *–* | В видимой зоне должен вмещаться период в пять лет |
| От года до пяти лет | Год | Квартал | В видимой зоне должны вмещаться 12 месяцев |
| От 3 месяцев до года | Квартал (Год) | Месяц | В видимой зоне должны вмещаться 3 месяца |
| От месяца до 3 месяцев | Месяц (Год) | День | В видимой зоне должен вмещаться 31 день |

«Настройка временного интервала» *–* это диапазон дат, отображаемых пользователю в календарном плане и графике показателей.

«Отображение верхней и нижней строки» *–* это то, что будет отображаться на временной шкале в календарном плане и графике показателей.

На рисунке 9 отображены работы в виде таблицы.

 Рисунок 9 – Отображение работ в виде таблицы

Требования к функциональности:

* Система должна отображать работы со всеми параметрами (рисунок 9).
* В система должна быть реализована пагинация, сортировка данных по всем параметрам.

Необходимые параметры для отображения, приведены в таблице 3.

Таблица 3 – Поля для отображения в таблице планов

| **Название поля** | **Содержание поля** |
| --- | --- |
| Куст | Куст, на котором выполняется работа |
| Месторождение | Месторождение, на котором выполняется работа |
| Бригада | Бригада, которая выполняет работу |
| Тип | Тип выполняемой работы |
| Начало | Дата начала выполнения работы |
| Конец | Дата завершения выполнения работы |
| Длительность | Длительность выполнения работы |
| Стоимость | Стоимость выполнения работы |
| План | План, к которому относится работа |

****5 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ПРОГРАММНОМУ СРЕДСТВУ****

## 5.1 Требования к программному обеспечению

Для обеспечения функционирования веб-сервиса необходимо предварительно установить на компьютере один из браузеров:

* Firefox версии не ниже 3;
* Opera версии не ниже 9;
* Chrome версии не ниже 4;
* Safari версии не ниже 3.2;
* Internet Explorer версии не ниже 9.

## 5.2 Требования к аппаратному обеспечению

Для обеспечения функционирования программного средства необходимы:

* количество свободного места на физическом накопителе не менее 1,5 ГБ;
* количество оперативной памяти не меньше 1 ГБ;
* процессор Intel Pentium 4 или более поздней версии с поддержкой SSE2;
* Windows 7, Windows 8.1, Windows 10 или более поздней версии;
* для правильного отображения необходимо разрешение от 1280×720 до 1920×1080.

## 5.3 Требования к надёжности

К программному средству предъявляются следующие требования в плане надёжности:

* время загрузки не должно превышать 4 секунды.

## 5.4 Требования к безопасности

К программному средству предъявляются следующие требования в плане безопасности:

* использование и своевременное обновление доверенного SSL-сертификата для обеспечения шифрования;
* обеспечение контроля уровня доступа пользователей в зависимости от их ролей.

6 ХАРАКТЕРИСТИКА ВЫБРАННЫХ ПРОГРАММНЫХ СРЕД И СРЕДСТВ

Для реализации веб-сервиса были выбраны следующие средства, технологии и инструменты разработки:

* TypeScript был выбран, так как он расширяет возможности JavaScript, используется для взаимодействия с пользователем.

Достоинства:

* 1. строгая типизация;
  2. читабельный код;
  3. обратная совместимость с JavaScript;
  4. реализует многие концепции ООП;
  5. широкая поддержка интеграционных сред разработки.

Недостатки:

* 1. не все браузеры поддерживают отладку;
  2. небольшое сообщество разработчиков.
* WebStorm был выбран, так как является полноценной средой разработки и включает в себя все необходимые функции.

Достоинства:

1. анализ кода «на лету»;
2. поддержка TypeScript;
3. интеграция с системами отслеживания ошибок
4. интеграция с системами управления версиями.

Недостатки:

1. высокие системные требования;
2. платный (есть возможность бесплатного использования для студентов).

* Angular был выбран, так как является полноценным Фреймворком для создания одностраничных приложений.

Достоинства:

1. Angular используется вместе с Typescript;
2. подробная документация;
3. внедрение зависимостей от компонентов, связанных с модулями и модульностью в целом;
4. структура и архитектура, специально созданные для большой масштабируемости проекта.

Недостатки:

1. разнообразие различных структур усложняет изучение;
2. проблемы с миграцией, которые могут возникнуть при переходе от старой версии к новой.

* SVG был выбран, так как данный векторный формат изображения масштабируется без потери качества.

Достоинства:

1. неограниченная масштабируемость;
2. быстро загружается;
3. легко модифицируется.

Недостатки:

1. не поддерживается старыми браузерами.

* Material Design был выбран, так как это графический язык и стиль в дизайне, созданный командой Google, чтобы помочь создавать удобные сайты и приложения.

Достоинства:

1. минималистический стиль;
2. понятен на интуитивном уровне;
3. имеется подробная документация.

Недостатки:

1. анимированные элементы требуют много ресурсов;
2. трудно различить интерактивные элементы.

* Node.js был выбран, так как выполняет роль веб-сервера.

Достоинства:

1. обеспечивает высокую производительность;
2. быстро развивается;
3. большое количество доступных библиотек.

Недостатки:

1. не поддерживает задачи с высокой загрузкой процессора;
2. недостаточно развитая документация.

7 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

## 7.1 Входные и выходные данные для сервиса загрузки

Входными данными для сервиса загрузки является Exсel таблица, содержащая работы со следующими параметрами:

* Rig – имя бригады;
* Pad – имя куста;
* WellName – имя скважины;
* Slot – номер скважины;
* StartDate – дата начала выполнения работы;
* FinishDate – дата завершения выполнения работы;
* Duration – продолжительность работы;
* Type – тип работы.

Выходными данными является данные отправляемые на сервер в формате JSON, которые имеют следующую структуру, изображенную на рисунке 10.

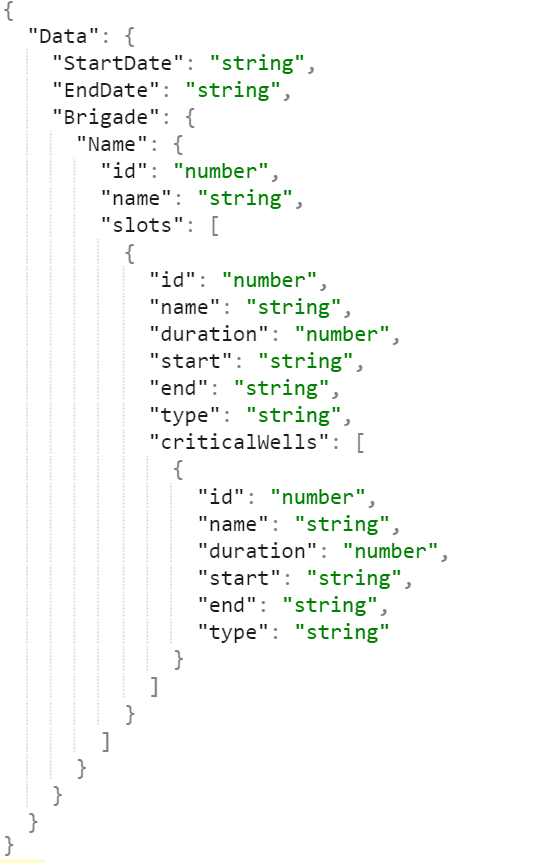


Рисунок 10 *–* Структура JSON

Где:

* Data – все передаваемые данные;
* StartDate – дата начала первой работы в данных;
* EndDate – дата конца последней работы данных;
* Brigade – массив содержащий бригады;
* Name – имя;
* Id – идентификатор объекта;
* Duration – продолжительность выполнения работ;
* Start – дата начала работы;
* End – дата завершения работы;
* CriticalWells – массив работ на скважинах требующие особого внимания.

## 7.2 Входные данные для веб-сервиса

Для взаимодействия с сервером используется язык запросов GraphQL, благодаря этому, структура входящих данных будет зависит от того, какую структуру данных мы зададим в запросе для получения работ, сгруппированных по бригадам, выполняется следующий запрос (рисунок 11).

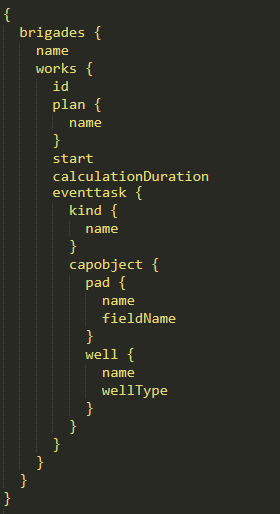


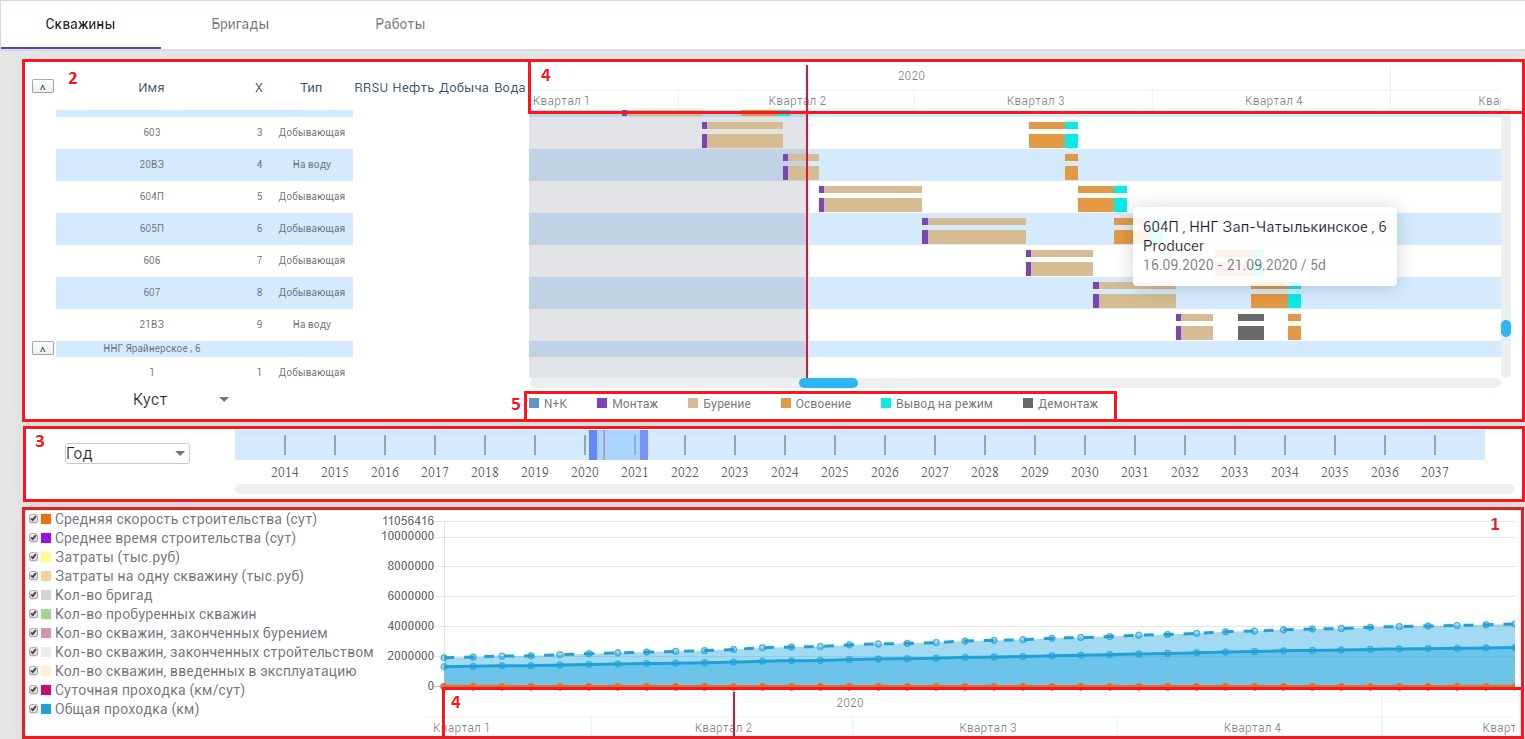
Рисунок 11 *–* Структура запроса

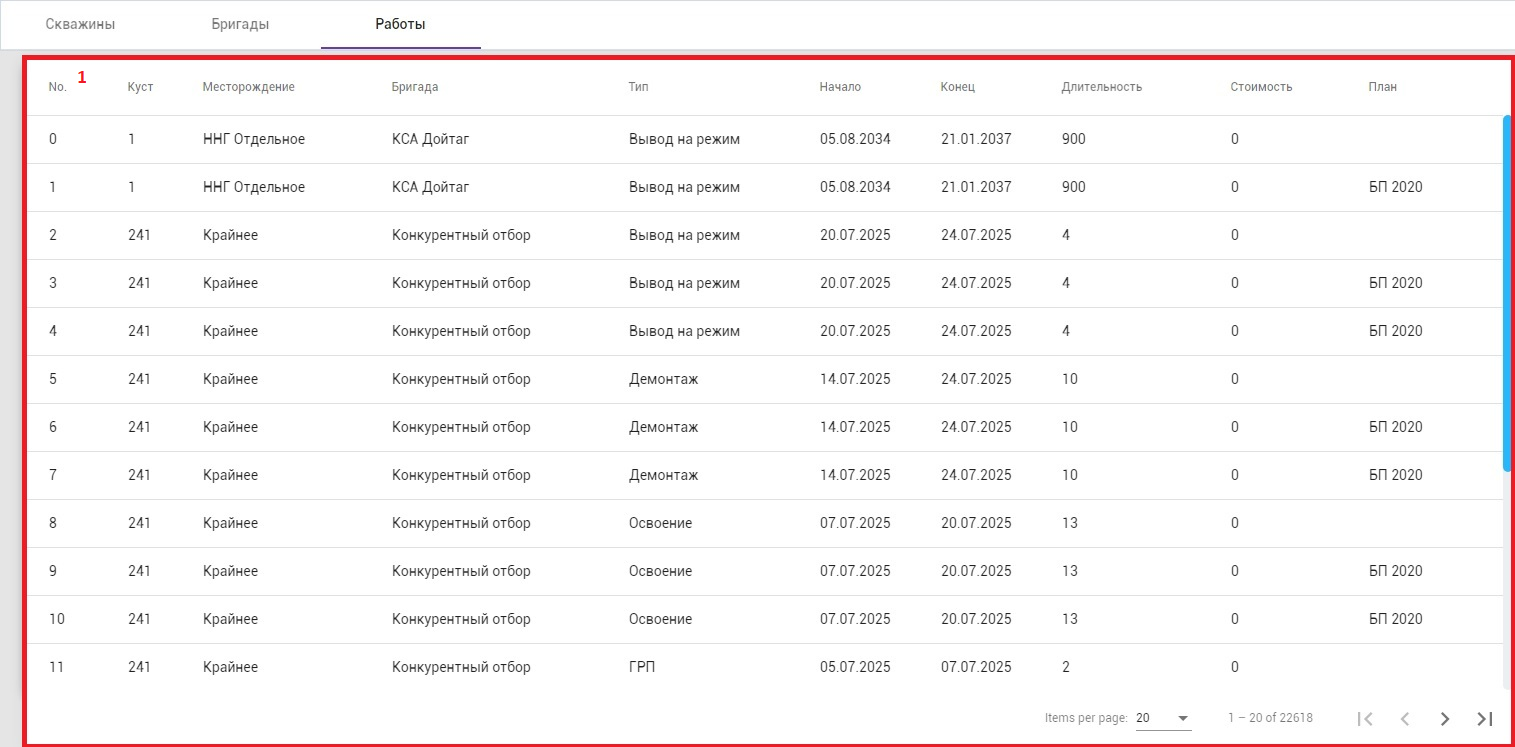
Где:

* brigades – массив содержащий бригады;
* works – массив содержащий работы;
* name – имя;
* plan – содержит информацию о плане;
* kind – содержит информацию о типе работы;
* pad – содержит информацию о кусте;
* well – содержит информацию о скважине;
* id – идентификатор объекта;
* start – дата начала выполнения работы;
* fieldsName – имя месторождения;
* wellType – тип скважины;
* calculationDuration – продолжительность выполнения работ.

8 СХЕМА ФУНКЦИОНИРОВАНИЯ ПРОГРАММНОГО СРЕДСТВА

## 8.1 Схема функционирования веб-сервиса

Рисунок 12 *–* Отображение компонент в пользовательском окне

Рисунок 13 *–* Отображение компоненты table-work в пользовательском окне

Сервис состоит из следующих модулей:

* App, это основной модуль, который запускается при запуске приложения, и через него используются остальные модули. Он состоит из app.component.css, app.component.html, app.component.ts.
* Time-line (рисунок 12 пункт 4), этот модуль отвечает за отображение временной линии. Он состоит из time-line.component.css, time-line.component.html, time-line.component.ts.
* Svg-drawing (рисунок 12 пункт 2), этот модуль отвечает за отображение календарного плана. Он состоит из svg-drawing.component.css, svg-drawing.component.html, svg-drawing.component.ts.
* Table-work (рисунок 13 пункт 1), этот модуль отвечает за отображение таблицы со всеми работами. Он состоит из table-work.component.css, table-work.component.html, table-work.component.ts.
* Legends (рисунок 12 пункт 5), этот модуль отвечает за отображение легенды для календарного графика. Он состоит из legends.component.css, legends.component.html, legends.component.ts.
* Brigades, этот модуль отвечает за подготовку данных для отображения на календарном плане сгруппированных по бригадам. Он состоит из brigades.component.css, brigades.component.html, brigades.component.ts.
* Facilites, этот модуль отвечает за подготовку данных для отображения на календарном плане сгруппированных по кустам и месторождениям. Он состоит из facilites.component.css, facilites.component.html, facilites.component.ts.
* Сhart (рисунок 12 пункт 1), этот модуль отвечает за отображения графиков. Он состоит из chart.component.css, chart.component.html, chart.component.ts.
* Provider, этот модуль отвечает за получения данных с сервера. Он состоит из provider.component.css, provider.component.html, provider.component.ts.
* Navigation-bar (рисунок 12 пункт 3), этот модуль отвечает за масштабирование и перемещение по датам. Он состоит из navigation-bar.component.css, navigation-bar.component.html, navigation-bar.component.ts.
* DataService, этот модуль отвечает за обмен данными между компонентами. Он состоит из файла data.service.ts.

Взаимодействие компонент изображено на рисунке 14.

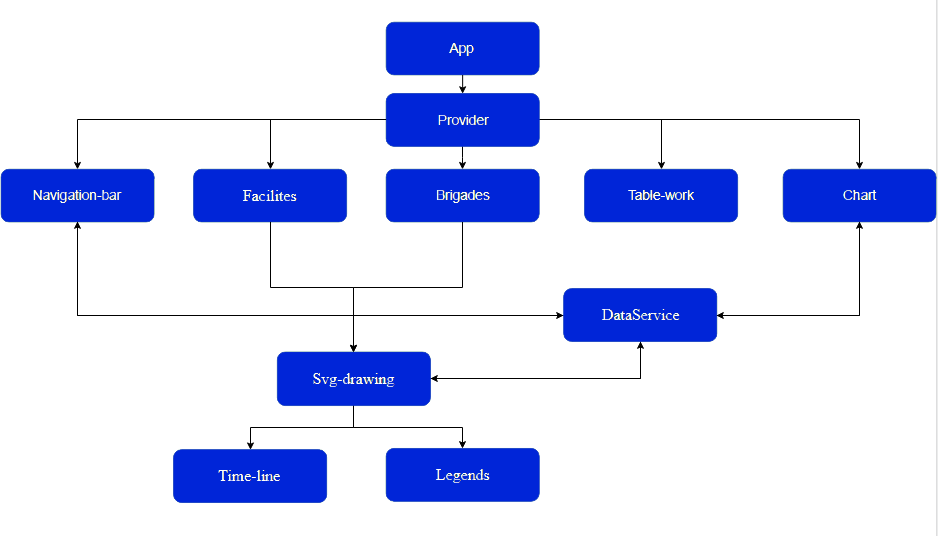
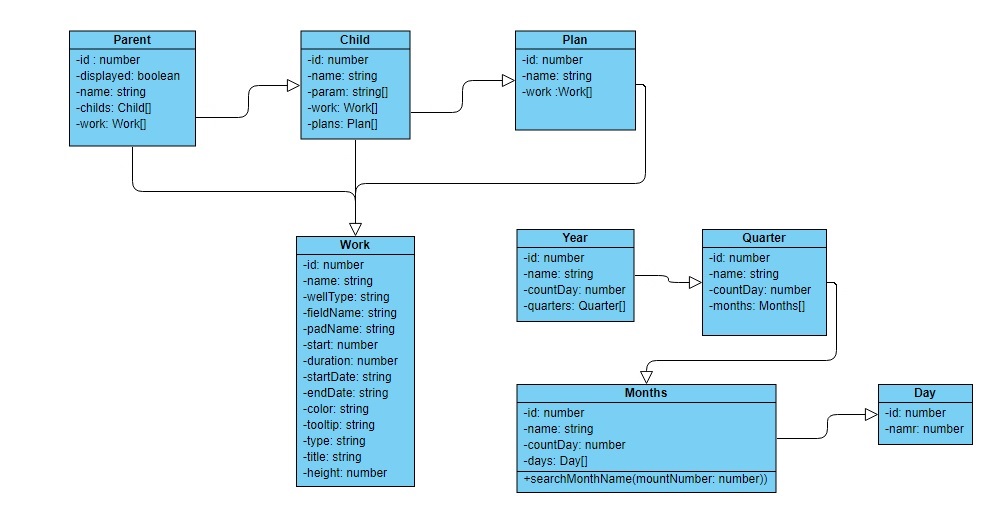
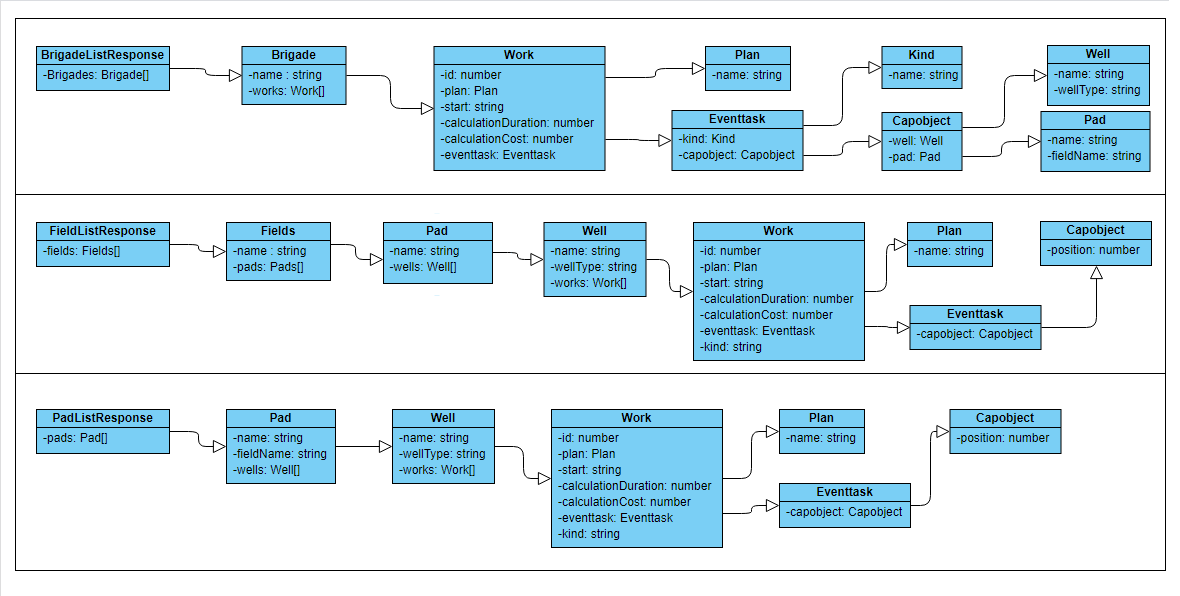


Рисунок 14 *–* Взаимодействие модулей

Веб-сервис состоит из классов, показанных на рисунке 15 и рисунке 16.

Рисунок 15 *–* Диаграмма классов веб-сервиса

Рисунок 16 *–* Диаграмма классов веб-сервиса

В веб-сервисе используются следующие методы:

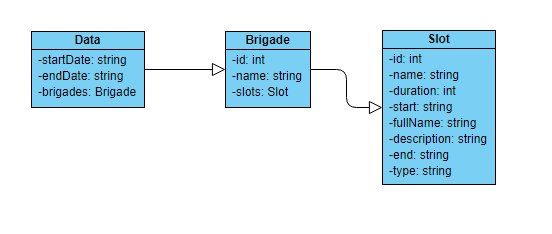
* ShowTooltip ($event, name, start, end, duration, type, description) метод выводит тултип;
* hideTooltip() метод прячет тултип;
* selectPad($event) метод выделяет объекты на графике;
* inputScale($event) метод изменяет масштаб;
* setDataTime() метод генерирует временную линию;
* jsonParse(json) метод получает данные из JSON-файла;
* downMouse($event) метод обрабатывает нажатие клавиши мыши;
* moveMouse($event) метод обрабатывает перемещение курсора мыши;
* jsonParseFact(json) метод получает данные из JSON-файла и добавляет их в фактические работы;
* jsonParsePlan(json) метод получает данные из JSON-файла и добавляет их в планируемые работы;
* setColor(type: string) метод возвращает цвет соответствующий типу работы;
* updateChart(scale: number, data: [], start: number, day: number) метод выполняет обновление графиков;
* selectChart($event) метод включает и отключает необходимые графики.

## 8.2 Схема функционирования загрузчика данных

Загрузчик состоит из модулей:

* WellController извлекает данные из Excel таблицы, обрабатывает их, и формирует из них JSON файл, для отправки на сервер.

Загрузчик состоит из классов, показанных на Рисунке 17.

Рисунок 17 *–* Диаграмма классов загрузчика данных

9 ОПТИМИЗАЦИЯ ПРОГРАММНОГО СРЕДСТВА

В процессе тестирования была обнаружена проблема медленной загрузки приложения, для решения этой проблемы были приняты следующие меры:

* Для оптимизации загрузки использовался метод ленивой загрузки, приложение было разделено на несколько модулей, при работе с которыми использовалась методика ленивой загрузки. При таком подходе для каждого модуля генерируется отдельный файл, который не будет загружаться, пока в нём не возникнет необходимость.
* Так же для оптимизации использовалась технология виртуальной прокрутки. Эта технология загружает элементы в DOM и выгружает их, основываясь на том, какая часть списка видна пользователю. Это ускоряет работу, поскольку полный список загружается не сразу, а только в соответствии с видимой на экране части.

В браузере Google Chrome полученный результат отображен в Таблице 4.

Таблица 4 – Результат оптимизации

| **Критерий оценки** | **До оптимизации** | **После оптимизации** |
| --- | --- | --- |
| Время полной загрузки приложения (мс) | 10516 | 3793 |
| Объем используемой оперативной памяти (мб) | 320 | 308 |

10 ОТЛАДКА И ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

Тестирование Веб-сервиса было разделено на этапы, такие как: функциональное тестирование, тестирование удобства использования, тестирование совместимости, тестирование производительности.

В процессе тестирования был найден и исправлен ряд проблем, таких как: медленная загрузка приложения, проблемы с отображением приложения в некоторых браузерах.

## 10.1 Функциональное тестирование

Цель функционального тестирования убедиться, что продукт соответствует функциональной спецификации.

Тестирование было выполнено методом системного тестирования. Для каждого требования были написаны тестовые случаи (Test Case), проверяющие выполнение данного требования.

Результат тестирования отображен в Таблице 5.

Таблица 5 – Результат функционального тестирования

| **Критерий тестирования** | **Результат** |
| --- | --- |
| Корректное отображение выбранного плана | План отображается корректно |
| Корректное изменение масштаба | Масштаб изменяется корректно |
| Корректное перемещение области просмотра (скроллинг) | Область просмотра изменяется корректно |
| Корректный выбор работ | Работы выбираются корректно |

## 10.2 Тестирование удобства использования

Тест удобства использования, используется для проверки программного продукта на соответствие с требованиями в плане удобств использования приложения. Тестирование выполнялось респондентами по тестовому сценарию.

В Таблице 6 показано сколько кликов необходимо было респондентам для выполнения определенной операции.

Таблица 6 – Количество кликов на операцию

| **Операция** | **Предполагаемое количество кликов** | **Количество кликов, сделанное пользователями** |
| --- | --- | --- |
| Переключение представления | 1 | 1 |
| Изменение масштаба | 2 | 2 |
| Выбор даты | 1 | 2 |
| Выбор объекта | 1 | 1 |

Результат тестирования отображен в Таблице 7.

Таблица 7 – Результат тестирования удобства использования

| **Критерий тестирования** | **Результат** |
| --- | --- |
| Простота и понятность в использовании сайта | Все страницы сайта понятны и просты в использовании |
| Удобство форм, кнопок и полей | Кнопки, формы и поля удобны для использования |
| Информативность, понятность, структурированность и логическая связность контента | Контент информативен, понятен, структурирован и логически связан |

## 10.3 Тестирование совместимости

Тестирование совместимости используется, чтобы убедиться, что приложение совместимо с другими элементами системы, в которой оно работает.

Результат тестирования отображен в Таблице 8.

Таблица 8 – Результат тестирования совместимости

| **Критерий тестирования** | **Результат** |
| --- | --- |
| Правильное отображение приложения в разных браузерах (IE, Firefox, Chrome, Safari, Opera) | Была найдена и исправлена проблема с отображением приложения в некоторых браузерах. В итоге приложение отображается во всех браузерах корректно |
| Корректное отображение шрифтов в разных браузерах | Шрифты отображаются правильно |
| JavaScript код выполняется работает в разных браузерах без ошибок | Ошибок выполнения кода не обнаружено |

## 10.4 Тестирование производительности

Тестирование производительности проводится с целью определения, как быстро работает вычислительная система или её часть под определённой нагрузкой.

Результат тестирования отображен в Таблице 9.

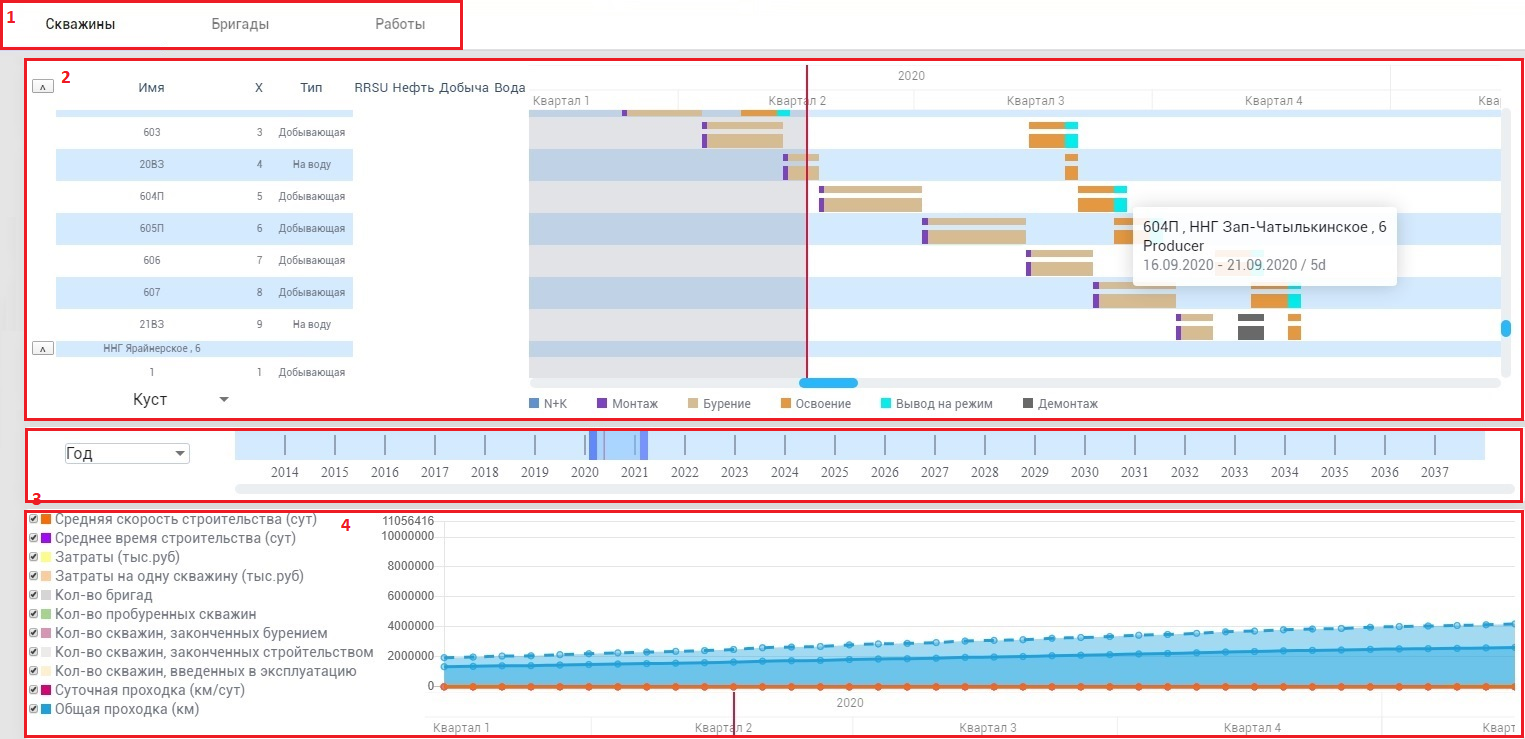
Таблица 9 – Результат тестирования производительности

| **Критерий тестирования** | **Результат** |
| --- | --- |
| Время загрузки страницы | Была найдена и исправлена проблема с загрузкой стран, в результате скорость загрузки не превышает 4 секунд |
| Отклик на действия пользователя | Отклик не вызывает неудобства в использовании приложения пользователем |

11 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

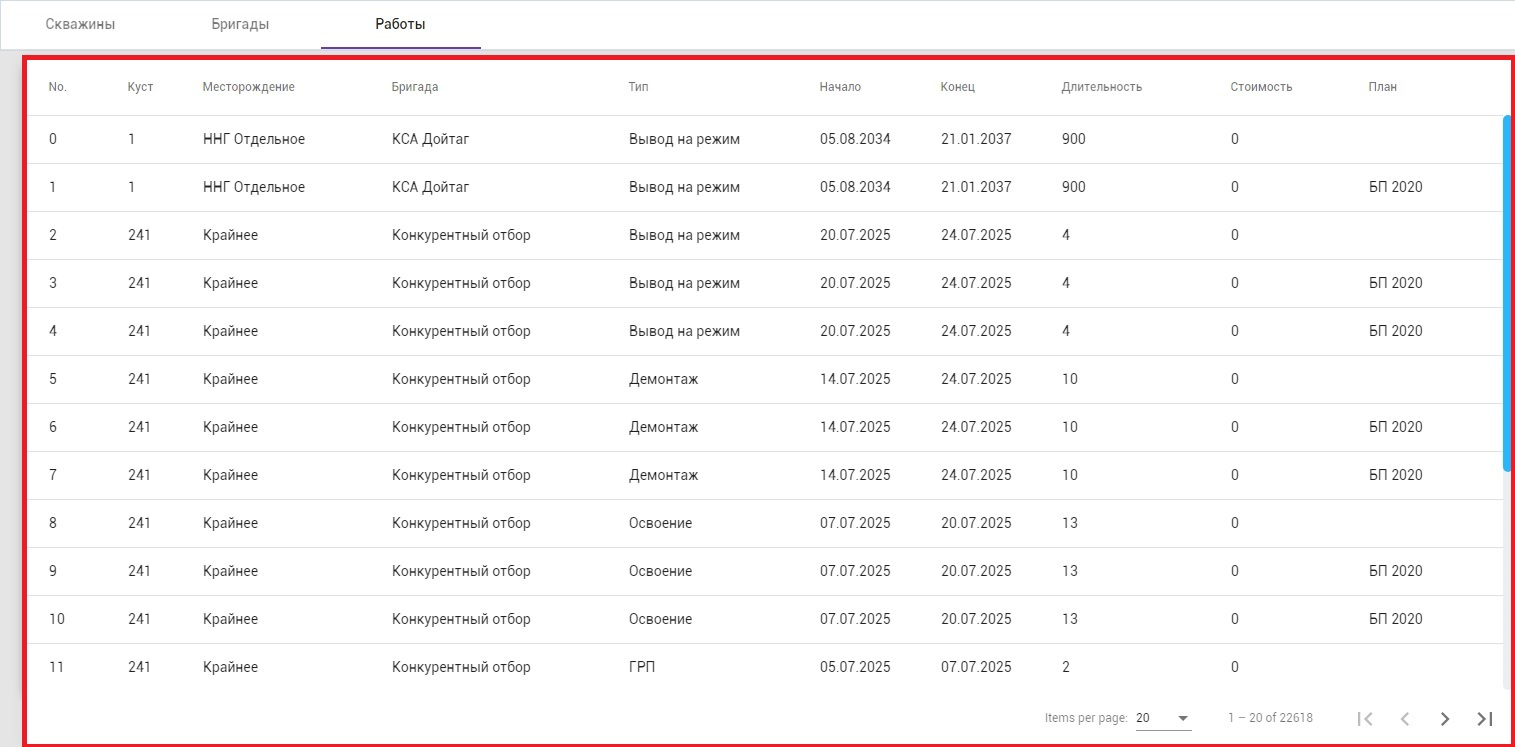
Для того чтобы запустить веб-сервис, необходимо открыть браузер и перейти по адресу http://91.201.72.215:52026/

После загрузки веб-сервиса открывается окно, со следующим содержимым (рисунок 18).

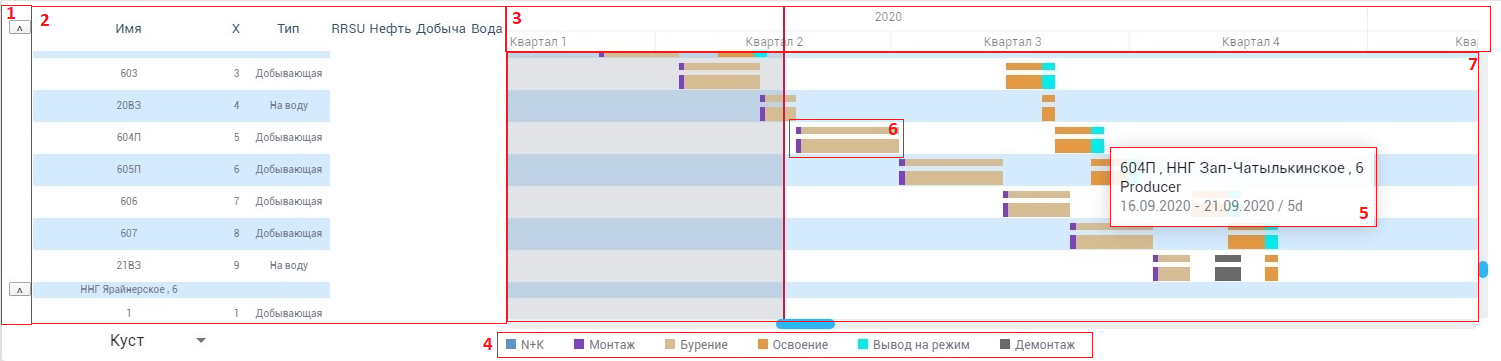
Рисунок 18 *–* Окно календарного графика

Все рабочее пространство разбито на 4 основных блока:

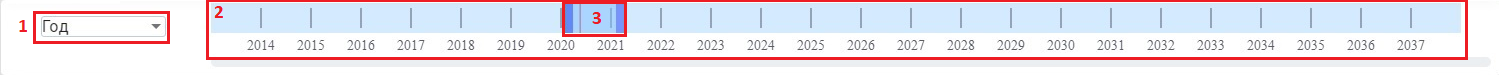
Блок 1 Рисунок 18 пункт 1 отвечает за переключение представлений, нажав на кнопку «Скважины», пользователь увидит работы, объединенные по скважинам, при нажатии на «Бригады» они будут объединены по бригадам. Если нажать «Работы», то все работы будут выведены в виде таблицы со всеми доступными параметрами (рисунок 19). Так же можно выполнять сортировку по любому из параметров.

Рисунок 19 *–* Отображение работ в виде таблицы

Блок 2 Рисунок 18 пункт 2 в нем отображается календарный график, здесь можно выбирать работы Рисунок 20 пункт 6 нажав на необходимую, для выбора нескольких работ необходимо зажимать CTRL, при наведении на работу появляется тултип с информацией о ней Рисунок 20 пункт 5. В таблице слева отображаются объекты группировки с определенными параметрами Рисунок 20 пункт 2. Кнопки Рисунок 20 пункт 1 позволяют сворачивать планы по объектам группировки. Внизу Рисунок 20 пункт 4 отображаются легенды.

Рисунок 20 *–* Окно отображения календарного плана

Блок 3 Рисунок 18 пункт 3 отвечает за масштабирование и перемещение по временной шкале. Тут пользователь может установить нужный масштаб уменьшая или увеличивая полосу масштаба Рисунок 21 пункт 3, так же для отображения нужного временного промежутка необходимо передвинуть ее на определенное место Рисунок 21 пункт 2. Изменить масштаб можно и с помощью меню Рисунок 21 пункт 1 выбрав необходимы (год, квартал, месяц).

Рисунок 21 *–* Панель навигации

Блок 4 Рисунок 18 пункт 4 отвечает за отображение графиков тут можно включить и отключить необходимые графики для отображения.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы поставленная задача была решена полностью. Полученный продукт, представляет предполагаемую изначальную ценность, которая заключалась в возможности одновременного мониторинга исполнения нескольких вариантов плана по ключевым целевым показателям.

Для решения поставленной задачи было выполнено следующее:

* Разработана универсальная компонента для отображения диаграммы Ганта;
* Разработано и реализовано программный продукт в виде загрузчика данных на сервер и веб-сервиса мониторинга этих данных.

Система будет иметь практическое применение, так как она является частью большой системы, разрабатываемой компанией ООО «ННТЦ».

Для дальнейшего развития веб-сервиса необходимо:

* Реализовать фильтрацию работ;
* Добавить возможность автоматической оптимизации плана;
* Редактирование плана в веб-сервисе.

Были проделаны следующие работы:

* Изучена новая предметная область бурения и освоения нефтяных скважин.
* Найдены и рассмотрены существующие библиотеки визуализации планов Gantt, Google Charts.
* Создана документированная спецификация функциональных и нефункциональных требований.
* Изучены новые технологии и программные средства:

1. Фреймворк Angular;
2. Язык разметки векторной графики SVG;
3. Стиль графического дизайна Material Design.

* Разработаны и реализованы: загрузчик, компонент для визуализации планов в виде расширенной диаграммы Ганта, одностраничное приложение для мониторинга.
* Выполнено тестирование и отладка веб-сервиса.
* Создано руководство пользователя.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Angular [Электронный ресурс] /фреймворк. –Web-сайт. – Google, 2020. – URL: https://angular.io/, свободный. – Яз. англ. – (Дата обращения: 06.02.2020).
2. Material Design [Электронный ресурс] / Material Design components for Angular –Web-сайт. – Google, 2019. – URL: https://material.angular.io/, свободный. – Яз. англ. – (Дата обращения: 06.02.2020).
3. SVG [Электронный ресурс] / Scalable Vector Graphics –Web-сайт. – W3C, 2010. – URL: https://www.w3.org/Graphics/SVG/, свободный. – Яз. англ. – (Дата обращения: 06.02.2020).
4. TypeScript [Электронный ресурс] / JavaScript that scales –Web-сайт. – Microsoft, 2020. – URL: https://www.typescriptlang.org/, свободный. – Яз. англ. – (Дата обращения: 06.02.2020).
5. JetBrains WebStorm [Электронный ресурс] / The smartest JavaScript IDE –Web-сайт. – JetBrains s.r.o., 2020. – URL: https://www.jetbrains.com/webstorm/, свободный. – Яз. англ. – (Дата обращения: 06.02.2020).
6. Node.js [Электронный ресурс] / JavaScript runtime built –Web-сайт. – Joyent, 2020. – URL: https://nodejs.org/en/, свободный. – Яз. англ. – (Дата обращения: 06.02.2020).
7. jsGantt [Электронный ресурс] / Developed by Shlomy Gantz and Brian Twidt. –Web-сайт. – Developed by Shlomy Gantz and Brian Twidt, 2019. – URL: http://jsgantt.com/, свободный. – Яз. англ. – (Дата обращения: 04.02.2020).
8. Frappe Gantt [Электронный ресурс] / Open Source Javascript Gantt. –Web-сайт. – Frappe Technologies Pvt. Ltd., 2020. – URL: https://frappe.io/gantt, свободный. – Яз. англ. – (Дата обращения: 06.02.2020).
9. Google developers [Электронный ресурс] / Google developers. –Web-сайт. – Google developers, 2020. – URL: https://developers.google.com/chart, свободный. – Яз. англ. – (Дата обращения: 04.02.2020).
10. FusionCharts [Электронный ресурс] / Build beautiful web & mobile dashboards. –Web-сайт. – InfoSoft, 2020. – URL: https://fusioncharts.com/, свободный. – Яз. англ. – (Дата обращения: 06.02.2020).
11. AnyChart [Электронный ресурс] / универсальные JavaScript диаграммы для HTML5. –Web-сайт. – AnyChart.Com, 2020. – URL: https://www.anychart.com/ru/, свободный. – Яз. рус. – (Дата обращения: 06.02.2020).
12. jQueryGantt [Электронный ресурс] / lightweight jQuery gantt plugin. –Web-сайт. – jQueryGantt, 2020. – URL: https://oguzhanoya.github.io/jquery-gantt/, свободный. – Яз. англ. – (Дата обращения: 06.02.2020).
13. GraphQL [Электронный ресурс] / A query language for your API –Web-сайт. – The GraphQL Foundation, 2020. – URL: https://graphql.org/, свободный. – Яз. англ. – (Дата обращения: 20.04.2020).

ПРИЛОЖЕНИЕ А

Ниже приведен фрагмент кода функции, которая парсит JSON-файл:

JsonParse(jsonData) {  
 const data = jsonData.Data.Brigade;  
 ***Object***.keys(data).forEach(bush => {  
 const pads = [];  
 const dataBrigade = data[bush];  
 ***Object***.keys(dataBrigade.Slots).forEach(slot => {  
 const pad = dataBrigade.Slots[slot];  
 const criticalWell = [];  
 ***Object***.keys(pad.CriticalWells).forEach(key => {  
 const well = pad.CriticalWells[key];  
 const startTime = new ***Date***(well.Start);  
 let endTime = new ***Date***(this.dataService.startData);  
 const startWell = toDay((+startTime - +endTime) );  
 endTime = new ***Date***(well.End);  
 criticalWell.push(new Pad(well.*Id*, well.Name, startWell, well.Duration, well.Type  
 , startTime.toLocaleDateString(), endTime.toLocaleDateString(), well.Description));  
 });  
 const eventStartTime = new ***Date***(pad.Start);  
 let eventEndTime = new ***Date***(this.dataService.startData);  
 const start = toDay((+eventStartTime - +eventEndTime) );  
 eventEndTime = new ***Date***(pad.End);  
 let padActialId;  
 this.actualPads.forEach(namePad => {  
 if (namePad.name === pad.Name) {  
 padActialId = namePad.id;  
 }  
 });  
 if (padActialId === undefined) {  
 this.actualPads.push(new PadIdName(this.actualPads.length , pad.Name));  
 padActialId = this.actualPads.length - 1;  
 }  
 pads.push(new Pad(padActialId, pad.Name, start, pad.Duration, pad.Type  
 , eventStartTime.toLocaleDateString(), eventEndTime.toLocaleDateString(), pad.Description, criticalWell));  
 });  
 const brigades = this.dataService.getParams();  
 let plan = new Plan(0 , pads);  
 if (brigades.length !== 0) {  
 brigades.forEach(value => {  
 if (dataBrigade.Name === value.name) {  
 value.AddPlan(plan);  
 plan = null;  
 }  
 });  
 if (plan !== null) {  
 this.dataService.setParams(new Brigade(dataBrigade.*Id*, dataBrigade.Name, plan));  
 }  
 } else {  
 this.dataService.setParams(new Brigade(dataBrigade.*Id*, dataBrigade.Name, plan));  
 }  
 });  
 let index = 0;  
 this.brigades = this.dataService.getParams();  
 this.brigades.forEach(brigades => {  
 brigades.plans.forEach( plan => {  
 plan.id = index++;  
 });  
 });  
 this.dataService.sumPlans = index;  
}

ПРИЛОЖЕНИЕ Б

Ниже приведен фрагмент кода функции, которая выполняет выделение объектов на календарном графике:

selectPad ($event) {  
 const idElement = $event.toElement.id;  
 if ($event.ctrlKey && this.oneClick === false) {  
 let isElement = false;  
 const copyPads = [];  
 this.selectPads.forEach( (value) => {  
 if (value === idElement) {  
 ***document***.getElementById(value).style.strokeWidth = '0px';  
 isElement = true;  
 } else {  
 copyPads.push(value);  
 }  
 });  
 this.selectPads = copyPads;  
 if (!isElement) {  
 ***document***.getElementById(idElement).style.strokeWidth = '1px';  
 this.selectPads.push(idElement);  
 }  
 this.oneClick = true;  
 } else if ($event.shiftKey && this.oneClick === false) {  
 const myRegex = /(\d+)-(\d+)-(\d+)-(\d+)/;  
 let matches = idElement.match(myRegex);  
 const idPad = matches[4];  
 const copyPads = [];  
 const selectId = [];  
 let index = 0;  
 this.selectPads.forEach( value => {  
 matches = value.match(myRegex);  
 const idPads = matches[4];  
 if (idPads === idPad) {  
 this.brigades.forEach(brigade => {  
 brigade.plans.forEach(plan => {  
 let i = 0;  
 plan.pads.forEach(pad => {  
 const id = brigade.id + '-' + plan.id + '-' + i + '-' + idPad;  
 if (value === id) {  
 selectId.push(id);  
 }  
 i++;  
 });  
 });  
 });  
 index++;  
 } else {  
 copyPads.push(value);  
 }  
 });  
 this.selectPads = copyPads;  
 if (selectId.length === index && index !== 0) {  
 selectId.forEach( id => {  
 ***document***.getElementById(id).style.strokeWidth = '0px';  
 });  
 } else {  
 this.brigades.forEach( (brigade) => {  
 brigade.plans.forEach(plan => {  
 let i = 0;  
 plan.pads.forEach(pad => {  
 if (pad.id == idPad) {  
 const id = brigade.id + '-' + plan.id + '-' + i + '-' + idPad;  
 ***document***.getElementById(id).style.strokeWidth = '1px';  
 this.selectPads.push(id);  
 }  
 i++;  
 });  
 });  
 });  
 }  
 this.oneClick = true;  
 } else if (this.selectPads.length === 0 && this.oneClick === false) {  
 ***document***.getElementById(idElement).style.strokeWidth = '1px';  
 this.selectPads.push(idElement);  
 this.oneClick = true;  
 } else if (this.oneClick === false) {  
 let isElement = false;  
 this.selectPads.forEach( (value) => {  
 if (value === idElement) {  
 isElement = true;  
 }  
 ***document***.getElementById(value).style.strokeWidth = '0px';  
 });  
 this.selectPads = [];  
 if (!isElement) {  
 ***document***.getElementById(idElement).style.strokeWidth = '1px';  
 this.selectPads.push(idElement);  
 }  
 this.oneClick = true;  
 }  
}