МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

Направление подготовки: 09.04.01 «Информатика и вычислительная техника»

Магистерская программа: «Вычислительные машины, комплексы, системы и сети»

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ УЧЕТА И ОБРАБОТКИ ДАННЫХ

Студент 31.05.2022 Иванов Никита Алексеевич
(подпись, дата) (ФИО полностью)

Руководитель 31.05.2022 Р.Н. Федюнин
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр 31.05.2022 А.С. Бычков
(подпись, дата) (фамилия, инициалы)

Рецензент [illegible] 06.06.2022 А.И. Мартышенец
(должность, место работы) (подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 02.06.2022 г. № 13)

Заведующий кафедрой М.А. Митрохин
(подпись) (фамилия, инициалы)

Работа защищена с отметкой отлично (протокол заседания ГЭК от 14.06.2022 № 7)

Секретарь ГЭК Е.И. Гурин
(подпись) (фамилия, инициалы)

Пенза 2022

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
«21» сентября 2020 г.

ЗАДАНИЕ
НА МАГИСТЕРСКУЮ ДИССЕРТАЦИЮ

1. Студент Иванов Никита Алексеевич гр. 20ВВм1 факультета вычислительной техники.

2.Тема работы «Программное обеспечение для учета и обработки данных»

Тема утверждена приказом ПГУ № 656/о от "21" сентября 2020 г.

3. Руководитель магистерской диссертации доцент, к.т.н. Федюнин Р.Н.

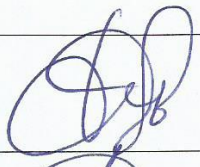
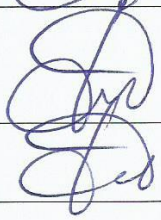
4. Задание на работу (назначение разработки, исходные данные и т.п.).

Назначение разработки – веб-приложение, позволяющее формировать перспективный и годовой план эксплуатации и выхода в ремонт техники. Необходимо предусмотреть возможность переноса приложения на другой компьютер с использованием средств контейнеризации. Приложение должно иметь удобный пользовательский интерфейс и позволять пользователю задавать годовые нормы расхода ресурсов и периодичность проведения планируемых мероприятий, трудоемкость проведения технического обслуживания, сроки проведения технического обслуживания по месяцам, очередность проведения технического обслуживания среди подразделений.

5. Перечень подлежащих разработке вопросов

Произвести анализ предметной области, определить стек технологий, разработать веб-приложение для формирования перспективного и годового плана эксплуатации и выхода в ремонт техники, выбрать программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, произвести отладку и тестирование приложения, оформить магистерскую диссертацию.

6. Календарный график выполнения работы

№ п/п	Планируемая форма работы во время практики	Количество часов	Календарные сроки проведения работы	Подпись руководителя
1	Выбор темы и разработка индивидуального плана проведения работ	26	21.09.2020 25.09.2020	
2	Подбор и изучение материала по теме работы	72	05.03.2021 14.03.2021	
3	Постановка задачи, выполняемой в рамках магистерской диссертации, и выбор стека технологий для ее решения	66	15.09.2021 24.09.2021	
4	Получение и анализ результатов	100	17.04.2022 13.04.2022	
5	Оформление магистерской диссертации	60	23.05.2022 31.05.2022	
	Общий объём часов	324		

Дата выдачи задания "21" сентября 2020 г.

Руководитель магистерской диссертации 21.09.2020 Федюнин Р.Н.
(подпись, дата)

Задание к исполнению принял студент 21.09.2020 Иванов Н.А.
(подпись, дата)

Работу к защите допустить
Декан факультета Фионова Л.Р.
(подпись, дата)
09.06.2022

РЕФЕРАТ

Магистерская диссертация содержит 107 листов, 31 рисунок, 1 таблицу, 8 формул, 8 источников, 2 приложения.

ВЕБ-ПРИЛОЖЕНИЕ, ВЕБ-ИНТЕФЕЙС, ВЕБ-СЕРВЕР, БАЗА ДАННЫХ, DJANGO, КОНТЕЙНЕРИЗАЦИЯ, DOCKER.

Цель работы – разработать веб-приложение, позволяющие формировать перспективный и годовой план эксплуатации и выхода в ремонт техники на основе имеющейся в базе данных информации.

Объектом разработки является приложение с удобным веб-интерфейсом, позволяющее пользователю внести в базу данных информацию о периодичности, трудозатратах, очередности и сроках проведения ремонтных работ, и на основе имеющихся данных о наличии техники в подразделениях и информации, введенной пользователем, осуществлять планирование ремонтных работ. Также необходимо предусмотреть возможность переноса приложения с использованием технологии контейнеризации.

В результате проведенной работы реализовано приложение, удовлетворяющее поставленной задаче, а именно, разработано веб-приложение, осуществляющее планирование эксплуатации и выхода в ремонт техники на основе имеющейся информации. Возможность переноса приложения реализовано посредством использования контейнеризатора приложений Docker.

Данное веб-приложение рекомендуется использовать для формирования перспективного и годового планов на основе имеющейся в базе данных информации. ПО является легко переносимым за счет использования средств контейнеризации.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ 5
ВВЕДЕНИЕ 6
1 Анализ предметной области 8
1.1 Планирование 8
1.2 Веб-приложение 9
1.2.1 Взаимодействие клиента и сервера 10
1.2.2 Достоинства и недостатки веб-приложений 11
1.3 Фреймворки веб-приложений 12
1.3.1 Фронтенд-фреймворки 12
1.3.2 Бэкенд-фреймворки 13
1.4 Современные СУБД 14
1.5 Docker 16
1.6 Выбор языка программирования и платформы для разработки 18
1.7 Выводы 19
2 Проектирование и разработка программного обеспечения 20
2.1 Постановка задачи 20
2.2 Задача № 1 «Формирование перспективного плана эксплуатации и выхода в ремонт техники» 20
2.2.1 Входная информация 21
2.2.2 Формирование выходного документа 22
2.3 Задача № 2 «Формирование годового плана эксплуатации и выхода в ремонт техники» 26
2.3.1 Входная информация 28
2.3.2 Формирование выходного документа 28
2.4 Описание функций 32
2.4.1 Описание функций веб-сервера 33
2.4.2 Описание функций веб-интерфейса 35
2.5 Описание работы программы 36

2.6 Отладка и тестирование 40
2.7 Выводы 42
3 Инструкция пользователя 43
3.1 Установленные годовые нормы расхода ресурсов и периодичность проведения планируемых мероприятий 44
3.2 Трудоемкость проведения ТО 46
3.3 Дни проведения ТО 46
3.4 Очередность проведения ТО среди подразделений 48
3.5 Перспективный план эксплуатации и выхода в ремонт 48
3.6 Годовой план эксплуатации и выхода в ремонт 53
ЗАКЛЮЧЕНИЕ 56
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 57
ПРИЛОЖЕНИЕ А Листинг программы 58
ПРИЛОЖЕНИЕ Б Презентация 100

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

БД – база данных

КР – капитальный ремонт

НСИ – нормативно-справочная информация

ПО – программное обеспечение

СР – средний ремонт

СУБД – система управления базами данных

ТО – техническое обслуживание

AJAX (Asynchronous Javascript and XML) – подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в фоновом обмене данными браузера с веб-сервером

API (Application Programming Interface) – программный интерфейс приложения

CSS (Cascading Style Sheets) – каскадные таблицы стилей

DOM (Document Object Model) - независящий от платформы и языка программный интерфейс, позволяющий программам и скриптам получить доступ к содержимому HTML-, XHTML- и XML-документов

HTML (HyperText Markup Language) – язык гипертекстовой разметки

HTTP (HyperText Transfer Protocol) – протокол передачи гипертекста

JSON (JavaScript Object Notation) – текстовый формат обмена данными, основанный на JavaScript.

URL (Uniform Resource Locator) – система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса

VM (Virtual Machine) – программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы

XML (eXtensible Markup Language) – расширяемый язык разметки

ВВЕДЕНИЕ

Работа с данными является неотъемлемой частью современной жизни. Развитие современных баз данных позволяет структурированно хранить большие объемы информации. Но порой у пользователей возникает необходимость осуществить некоторое планирование на основе имеющихся данных, что подталкивает к разработке приложений, осуществляющих данные задачи.

Влияние глобальной компьютерной сети Internet на современный мир располагает к использованию веб-интерфейсов. Применение веб-технологии не требует установки программного обеспечения на каждое устройство конечного пользователя, а позволяет развернуть его на сервере, имеющим доступ к базе данных. Доступ пользователя к сервису осуществляется посредством использования веб-интерфейсов.

Не всегда для разворачивания веб-приложения выделяется отдельный сервер. Для работы с ограниченным кругом лиц, оно может быть запушено на одном из локальных компьютеров пользователей. Однако данные компьютеры могут быть не оснащены необходимым набором приложений и пакетов, необходимых для запуска сервера.

Решением данной проблемы выступает применение программного обеспечения для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации Docker.

Docker позволяет разворачивать контейнеры внутри которых находится весь набор необходимых программ. Контейнер создается на основе образов, которые могут быть созданы программистом на основе написанной им программы или же скачаны существующие из репозитория.

Целью магистерской диссертации является написание программного обеспечения с использованием веб-технологии. Приложение позволяет осуществить перспективное и годовое планирование эксплуатации и выхода в ремонт техники, результатом работы является выходной документ формата xlsx.

Готовое приложение необходимо упаковать в контейнер с использованием технологии Docker. Контейнер должен включать в себя образ разработанного приложения и необходимые для его работы библиотеки, базу данных и графический интерфейс для взаимодействия с БД.

1 Анализ предметной области

Данный раздел содержит обзор на важность применения средств планирования. Также представлена возможность применение веб-технологий, принцип работы, достоинства и недостатки. В этом разделе рассматриваются современные фреймворки для разработки веб-приложений и свободные реляционные системы управления базами данных. Помимо этого, представлено описание программного обеспечения для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации.

1.1 Планирование

При работе с информацией, связанной с расходом ресурсов техники, пользователю может потребоваться заранее составить ряд задач исходя из имеющихся данных. Планирование позволяет построить графики проведения работ по обслуживанию и ремонту на различные промежутки времени, такие как месяц, год, десятилетие.

Цели создания систем планирования:

– минимизация затрат на поддержание работоспособности техники,

– сокращение количества отказа оборудования,

– переход от экстренных ремонтов к плановым,

– предупреждение остановки производственных процессов.

Помимо этого, применение планирования позволяет обеспечить:

– содержание техники в исправном состоянии,

– своевременное проведение технического обслуживания,

– технически правильное, эффективное и экономичное использование техники,

– управление расходом ресурса,

– содержание в работе техники с ресурсом до очередного среднего или капитального ремонта в пределах установленных норм.

Исходя из вышеперечисленных достоинств разработка системы планирования эксплуатации и выхода в ремонт техники является актуальной и востребованной.

1.2 Веб-приложение

Веб-приложения представляют собой программы, основанные на архитектуре клиент-сервер. Их особенностью является то, что приложение выполняется не на пользовательском компьютере, а на сервере. Работа ПО заключается в получении запросов от клиента, их обработке и выдачи результата. Обмен запросами и результатами между клиентом и сервером осуществляется посредством сети Internet. Общая структура веб-приложения представлена на рисунке 1.

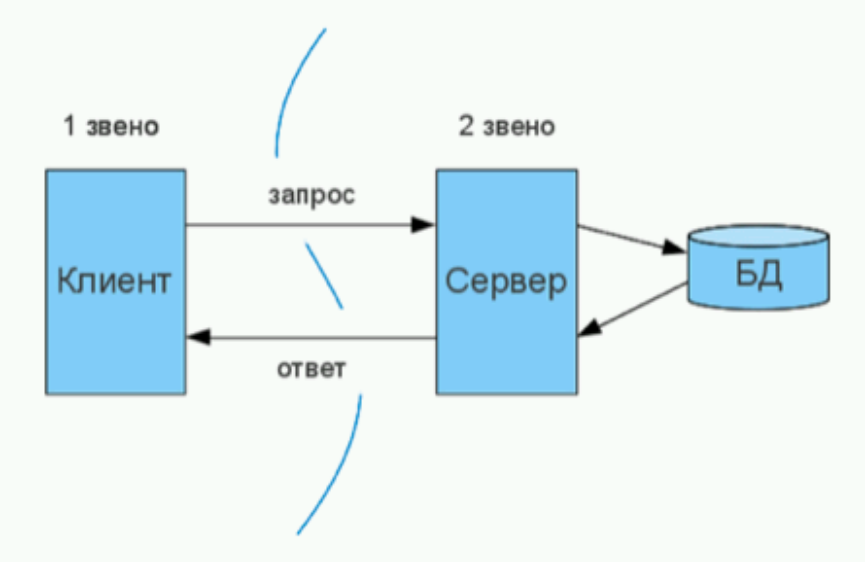


Рисунок 1 - Структура веб-приложения

В качестве интерфейса приложений выступает браузер, в котором отображается веб-интерфейс, описанный с помощью языка гипертекстовой разметки html. Основными задачами браузера являются отображение данных и предоставление пользователю возможности взаимодействия с приложением.

На стороне сервера веб-приложение занимается обработкой поступающих запросов, выбором необходимого порядка действий на поступивший запрос, формированием ответа на основе выполненных действий и отправкой результата работы обратно пользователю. Помимо этого, если

существует необходимость сохранения данных, сервер выполняет обмен данными с базой данных.

1.2.1 Взаимодействие клиента и сервера

Взаимодействие веб-браузера с сервером осуществляется посредством протокола передачи гипертекста HTTP. Браузер отправляет HTTP-запросы, которые содержат:

– путь (URL) определяет целевой ресурс и сервер,

– метод определяет необходимое действие. Наиболее используемыми являются GET, POST, PUT и DELETE позволяющие получить ресурс, создать новый, обновить существующий и удалить соответственно,

– дополнительную информацию.

Общая схема взаимодействия с веб-сервером представлена на рисунке 2.

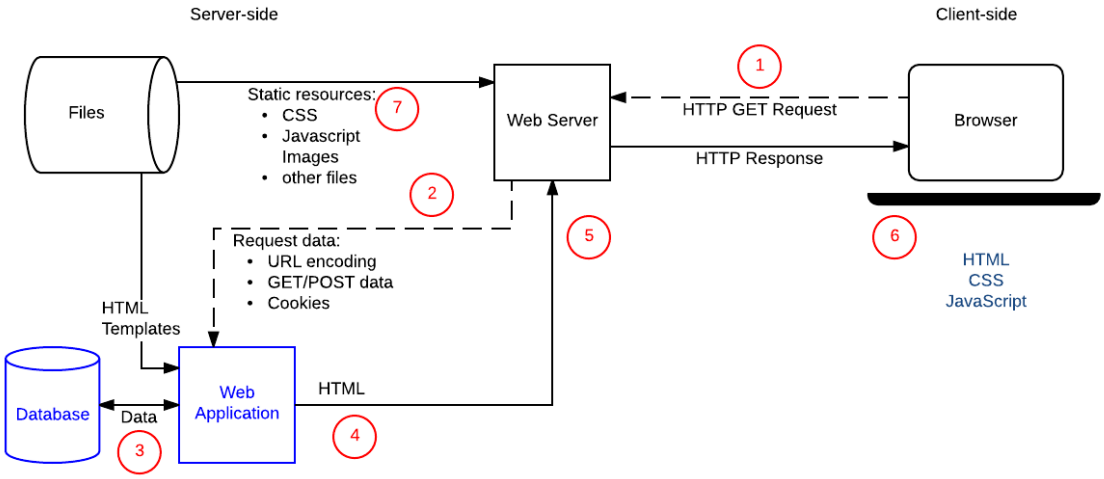


Рисунок 2 - Схема взаимодействия веб-приложения

Данная схема включает ряд этапов:

– веб-браузер отправляет HTTP-запрос на сервер,

– веб-сервер определяет, как обработать запрос на основе правил сопоставления шаблонов,

– веб-приложение выполняет задачи, требуемые запросом, в процессе которых производится взаимодействие с базой данных,

– веб-приложение формирует ответ на запрос,

– веб-приложение возвращает сгенерированный ответ,

– браузер производит обработку полученного результата,

– при необходимости производится загрузка статический файлов с веб-сервера.

1.2.2 Достоинства и недостатки веб-приложений

Веб-приложения становятся все более востребованными среди пользователей. Расмотрим их преимущество перед десктопным ПО.

Во-первых, использование веб-технологии избавляет конечного пользователя от необходимости установки приложения на свой компьютер, что в некоторых случаях позволяет сэкономить большое количество занимаемого места.

Во-вторых, приложения, лежащие на сервере, не требуют от пользователя настроек перед началом работы, этим занимаются разработчики непосредственно при установке ПО.

Третье преимущество – отсутствие требований к аппаратной платформе, все что требуется для работы это возможность запустить браузер и иметь стабильный доступ к сети интернет.

Следующим достоинством можно выделить мобильность. Так как вся необходимая информация располагается на сервере, то пользователь может работать как дома за компьютером, так и в пути с использованием мобильных средств.

Использование клиент-серверной архитектуры избавляет разработчиков от необходимости рассылки обновлений для ПО своим клиентам. Они способны самостоятельно произвести обновление на стороне сервера, тем самым все пользователь получат новую версию приложения.

Однако для работы веб-приложений необходимо мощное серверное оборудование, что делает размещение приложения более дорогим по сравнению с десктопным, где учитывает вычислительная мощность компьютера пользователя.

Также для работы необходимо стабильное и высокоскоростное подключение к сети интернет. Нельзя забывать и о обеспечении безопасности хранимой на сервере информации, так как в случае сбоя может произойти утечка данных.

Помимо этого, использование веб-приложений имеет некоторые задержки. Это связано с тем, что при выполнении запроса производится установка HTTP-соединения, обработка запроса сервером и отправка результата браузеру, который осуществляет загрузку полученного результата.

1.3 Фреймворки веб-приложений

Веб-фреймворк – платформа для создания веб-приложений и сайтов, вызволяющая облегчить разработку и объединение компонентов большого проекта. Преимуществом использования фреймворков является то, что фреймворк определяет унифицированную структуру для построенных на его базе приложений. Из-за этого приложения на фреймворках легче сопровождать и дорабатывать.

Веб-фреймворки можно разделить на два типа:

– фронтенд-фреймворки,

– бэкенд-фреймворки.

1.3.1 Фронтенд-фреймворки

Фронтенд-фреймворки предназначены для разработки внешнего вида веб-приложений. Они работают в браузеры и не связаны с логикой работы приложения. С их помощью появляется возможность улучшать и внедрять новые интерфейсы, создавать различные анимации и одностраничные приложения. Рассмотрим наиболее популярные из них.

React хотя и является библиотекой, но из-за своей значимости зачастую сравнивают с фреймворком. Преимуществами React является высокая производительность с использованием виртуального DOM, возможность использовать компоненты повторно. Помимо этого, фреймворк имеет стабильный и безопасный кода, легок в расширении и обучении.

Angular – фреймворк, основанный на языке TypeScript, разработан компанией Google и запущен в 2016 году. Главной особенностью является двусторонняя привязка данных, что позволяет осуществить синхронизацию между представлением и моделью в реальном времени. Помимо этого, фреймворк предоставляет кроссплатформенность, высокую производительность и прост в сопровождении.

Vue.js – фреймворк JavaScript для разработки пользовательских интерфейсов и одностраничных приложений. Из преимуществ можно выделить простоту освоения, хорошую документацию, возможность использовать с обычными редакторами и возможность быть интегрированным в другие фреймворки.

JQuery был запущен в 2006 году и является одним из самых старых фреймворков, несмотря на это остается актуальным и широко используемым. Он позволяет производить манипулирование DOM и CSS и позволяет в фоновом режиме производить обмен между клиентом и сервером посредством AJAX [1].

Исходя из достоинств, рассмотренных фронтенд фреймворков, был выбран jQuery для разрабатываемого приложения. Основными предпосылками к выбору послужили асинхронный фоновый обмен данными в архитектуре клиент-сервер, а также возможность уменьшить нагрузку на сервер за счет генерирования не всей страницы, а лишь ее части. Уменьшение генерации страницы также уменьшает размер трафика между клиентом и сервером.

1.3.2 Бэкенд-фреймворки

Бэкенд-фреймворки предназначены для работы на стороне сервера. Их основными задачами является маршрутизация URL-адресов и взаимодействие с базой данных. Помимо это они способны формировать выходные данные и отвечать за безопасность. Наиболее популярные из них на сегодняшний день представлены ниже.

Django фреймворк на основе языка программирования Python с открытым исходным кодом [2]. Основная задача – упростить разработку сложных веб-

сайтов и приложений, управляемы базами данных. Django обеспечивает высокую безопасность, предоставляет интерфейс администрирования и имеет обширные руководства и документацию, что упрощает его изучение.

Laravel это веб-фреймворк PHP с открытым исходным кодом, основанный на Symfony. Предназначен для разработки веб-приложений по архитектурному шаблону MVC (модель-представление-контроллер). Некоторые функции предоставляют различные способы доступа к базам данных и помогают в обслуживании и развертывании приложений.

Express JS представляет собой основу для бэкенда приложений на Node.js. Из преимуществ можно выделить простоту настройки и конфигурации, что ускоряет разработку. Фреймворк позволяет определить маршруты приложения на основе методов URL и HTTP и осуществить простое подключение к базам данных.

Ruby on Rails это серверный фреймворк написанный на языке Ruby облегчающий использование стандартов JSON и XML для передачи данных и HTML, CSS и JavaScript для взаимодействия с пользователем. К достоинствам можно отнести высокую производительность, высокую гибкость за счет использования приложением возможностей бэкенда и фронтенда Rails.

Так как серверная часть приложения планируется быть написана на языке программирования python, наиболее предпочтительным будет являться фреймворк Django.

1.4 Современные СУБД

Для доступа к данным, хранящимся в базе данных, а также взаимодействия с ними используется система управления базами данных. СУБД – это комплекс языковых и программных средств, предназначенный для создания, ведения и совместного использования БД многими пользователями. На сегодняшний день пользователю предоставляется широкий список СУБД, наиболее распространенные представлены ниже.

MySQL является наиболее популярной СУБД для веб-приложений. MySql поддерживает большое количество типов таблиц, таким образом пользователь

может выбрать таблицы поддерживающие полнотекстовый поиск типа MyISAM или таблицы InnoDB, позволяющие осуществлять транзакции на уровне отдельных записей. Система считается надежной и хорошо документированной, а также имеет простой и удобный для использования интерфейс, и пакетные команды, которые предоставляют возможность удобно обрабатывать большие объемы данных.

MongoDB является примером NoSQL-систем и использует JSON-подобные документы и схему базы данных. Данная СУБД предназначена для приложений, которые используют структурированные и неструктурированные данные. Изначально система была разработана не для работы с моделями реляционных данных, хотя и может это выполнить, то при работе с такими моделями может возникнуть проблема производительности. Однако MongoDB хорошо справляется с данными, которые нельзя отнести к реляционным.

MariaDB является ответвлением СУБД MySQL толчком к ее созданию стала необходимость обеспечения свободного статуса СУБД, в противовес политике лицензирования MySQL. Ядро предоставляет на выбор несколько систем хранения и это позволяет использовать ресурсы более оптимизировано, что повышает производительность обработки запросов. В состав СУБД включена подсистема хранения данных XtraDB позволяющее заменить подсистему хранения InnoDB, помимо этого, включены подсистемы Aria, PBXT и FederateX.

PostgreSQL является бесплатной системой управления базой данных, которая часто применяется для ведения баз данных веб-приложений. Является хорошо развитой и предоставляет возможность управлять структурированными и неструктурированными данными. Также позволяет импортировать информацию из других типов баз данных посредством собственных инструментов. СУБД является масштабируемой и позволяет обрабатывать терабайты данных.

В качестве системы управления базами данных для разрабатываемого приложения была выбрана СУБД PostgreSQL. Даная система позволяет

создавать вложенные подзапросы и сложные команды SQL. Также предоставляет поддержку внешних ключей, представлений и триггеров, что позволяет скрыть сложность структуры базы данных и отказаться от сложных команд SQL [3].

1.5 Docker

Результатом разработки приложения является ПО, код которого необходимо предоставить со всеми его составляющими, такими как библиотеки, фреймворки, сервер, база данных и т.д. Приложение может без проблем работать на компьютере разработчика, но при переносе на другое устройство могут возникнуть проблемы с запуском. Для решения таких проблем применяется программное обеспечение для автоматизации развертывания и управления приложениями в средах с поддержкой контейнеризации Docker.

Первоначально для решения данной проблемы применялись средства виртуализации, но они имеют ряд недостатков:

– не все виртуальные машины поддерживают совместное использование,

– использование виртуальных машин требует установки дополнительно операционной системы что увеличивает объем занимаемой памяти, а на сервере может быть более одной VM,

– медленная загрузка,

– обеспечение работы нескольких виртуальных машин требует большое количество аппаратных мощностей.

Docker в свою очередь разделяет ядро операционной системы на контейнеры, которые работают как отдельные процессы [4]. Он также предоставляет ряд других преимуществ:

– позволяет ускорить процесс разработки за счет отсутствия необходимости установки вспомогательных инструментов, так как их можно запускать в контейнерах,

– простое масштабирование,

– удобная инкапсуляция приложений,

– предоставляет возможность мониторинга контейнеров.

Docker поддерживает операционные системы Linux, Windows и macOS. Единственное отличие заключается в том, что на последних двух ОС платформа инкапсулируется в виртуальную машину.

Рассмотрим основные термины программного обеспечения Docker.

Образ – является основополагающим элементом любого контейнера. Образ может быть создан пользователем на основе разработанного приложения или же скачан из репозитория.

Контейнер – исполняемый экземпляр, который инкапсулирует программное обеспечение. Он состоит из образов. Особенностью является то, что его удаление и создание нового не занимает много времени.

Порт – это порт TCP/UDP. Позволяет установить связь порта контейнера с портом локального компьютера, тем самым определяя доступ к контейнеру.

Том представляет собой общую папку, предназначенную для сохранения данных контейнера независимо от его жизненного цикла. Являет необходимой частью, так как выключение контейнера приводит к уничтожению всех внесенных изменений.

На сегодняшний день компания Docker Inc предоставляет доступ к публичному репозиторию, которых хранит множество образов. Ресурс предоставляет официальные образы, разработанные компанией Docker, а также созданных в сотрудничестве с разработчиками ПО.

Разрабатываемое приложение требует использование сервера, базы данных и графического интерфейса для удобной работы с базой. Для этого необходимо создать три соответствующих контейнера. Но Docker позволяет управлять отдельными контейнерами, а не несколькими одновременно.

Решением проблемы является использование инструментального средства Docker Compose, входящее в состав Docker. Оно позволяет единовременно управлять несколькими контейнерами, из которых состоит приложение, что продемонстрировано на рисунке 3. Этот инструмент

предлагает те же возможности, что и Docker, но позволяет работать с приложениями более сложной архитектуры.

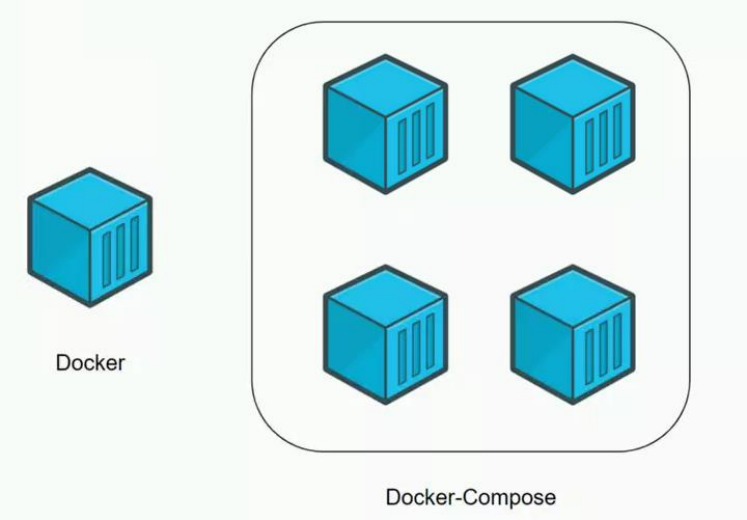


Рисунок 3 - Отличие Docker и Docker Compose

1.6 Выбор языка программирования и платформы для разработки

Так как веб-приложение состоит из веб-интерфейса и серверной части, необходимо определить язык программирования для каждой из них.

Для фронтенд разработки были выбраны HTML, CSS и JavaScript. С их помощью можно создавать веб-интерфейсы, способные обрабатывать события, производимые пользователем при работе. Также было принято решение использовать фреймворк jQuery, позволяющий легко взаимодействовать с элементами DOM и предоставляющий API для работы с AJAX [5].

Для серверной части приложения используется фреймворк Django, следовательно, и объектно-ориентированный, интерпретируемый и переносимый язык программирования Python [6]. Применения языка Python облегчает разработку программного обеспечения, так как позволяет использовать обширное количество готовых библиотек. Django в свою очередь берет на себя задачи по адресации, доступом к базе данных и обеспечении безопасности.

Для разработки программного обеспечения была выбрана интегрированная среда разработки для языка программирования Python – PyCharm. IDE имеет набор средств для эффективной разработки на языке

Python. PyCharm выполняет инспекцию кода, автодополнение, обеспечивает множество рефакторингов [7]. Помимо этого, поддерживает свежие версии фреймворка Django и содержит интегрированные консоли для Python, Django, SSH, отладчика и баз данных.

1.7 Выводы

В результате анализа предметной области была определена важность использования средств планирования. Также были рассмотрены возможность применения веб-технологии и выбраны современные вреймфорки для разработки такого приложения. Помимо этого, была выбрана СУБД, язык программирования и среда разработки для разрабатываемого ПО и выделены достоинства использования средств контейнеризации.

2 Проектирование и разработка программного обеспечения

В данной главе описана реализация разрабатываемого приложения. Особое внимание уделяется выделению и описанию различных сущностей, и их взаимодействию между собой. При этом освещены основные особенности реализации, связанные с разработкой приложения клиент-серверной архитектуры. В данном разделе объясняется принцип работы и внутреннее устройство системы в целом.

2.1 Постановка задачи

Целью разработки является создание веб-приложения, осуществляющее выполнение двух задач:

– формирование перспективного плана эксплуатации и выхода в ремонт,

– формирование годового плана эксплуатации и выхода в ремонт.

А после при помощи технологии контейнеризации реализовать возможность переноса разработанного приложения.

Для решения данной задачи необходимо определить набор входной и выходной информации для формирования планов. Также определить структуру серверной части приложения и пользовательского интерфейса. Помимо этого, определить способ применения программного обеспечения Docker для переноса приложения.

2.2 Задача № 1 «Формирование перспективного плана эксплуатации и выхода в ремонт техники»

Перспективный план эксплуатации и выхода в ремонт разрабатывается на 10 лет в целях:

– обеспечения равномерного выхода основных образцов техники в ремонт,

в течении межремонтного периода эксплуатации,

– своевременного проведения ТО,

– своевременной замены составных частей, выслуживших установленные сроки эксплуатации.

В плане указывается год отправки в средний и капитальный ремонт. Одновременный выход в ремонт одноименных образцов техники не должен быть более 25%, а при наличии четырех и менее – не более одного.

Перспективный план состоит из трех разделов:

1) планируемые мероприятия ТО и ремонта техники,

2) пояснительная записка для обобщения планируемых мероприятий,

3) исходные данные для планирования ТО и ремонта техники, периодичности их проведения.

В перспективном плане эксплуатации и выхода в ремонт техники планирование осуществляется по подразделениям.

После заполнения первого раздела перспективного плана эксплуатации и выхода в ремонт техники, заполняется второй раздел плана (пояснительная записка) путем суммирования количества образцов по планируемым мероприятиям.

2.2.1 Входная информация

Источниками входной информации для настоящей задачи являются:

– информация о наименованиях подразделений, численности техники,

– данные о наличии и качественном состоянии техники,

– данные об наработке,

– сведения о проведенных технических обслуживаниях и ремонтах,

– установленные годовые нормы расхода ресурсов и периодичность проведения мероприятий.

Первые четыре источника изначально находятся в базе данных, последний пользователь задает самостоятельно с использованием веб-интерфейса в соответствии с таблицей 1.

Таблица 1 - Установленные годовые нормы расхода ресурсов и периодичность проведения планируемых мероприятий

Образец и его составные части	Ед. изм рес	Годовая норма расхода ресурсов		Проводимые мероприятия							
		Обычная группа эксплуатации	С повышенным расходом	Ремонт		Техническое обслуживание					
				СР	КР	ТО-1	ТО-2	ТО-1х	ТО-2х	ТО-2х с ПКП	РТО
1	2	3	4	5	6	7	8	9	10	11	12
Камаз	год			3	5	1	2	2	3	3	1
шасси	км	120	150								
двигатель	м/ч	2000	2500								

Продолжение таблицы 1

Образец и его составные части	Ед. изм рес	Проводимые мероприятия				НР
		УБ	АБ	ВБ (ППО, ОУ)	Гус/ Шина	
1	2	13	14	15	16	17
Камаз	год	3	5	2	5	
шасси	км					600
двигатель	м/ч					7000

2.2.2 Формирование выходного документа

Перед формированием выходного документа «Перспективный план эксплуатации и выхода в ремонт техники» пользователю необходимо заполнить таблицу с установленными годовыми нормами расхода ресурсов и периодичность проведения планируемых мероприятий. В результате выполнения данной задачи формируется файл xlsx формата.

2.2.2.1 Формирование первого раздела перспективного плана эксплуатации и выхода в ремонт

Первый раздел формы подразделяется на два подраздела:

1) техника обычной группы эксплуатации,

2) техника с повышенным расходом ресурса.

Выходной документ содержит следующие графы:

– графа 1 «подразделение»,

– графа 2 «Образец техники» - указываются марка и составные части в объеме, указанном в таблице 1 графе 1,

– графа 3 «Заводской номер»,

– графа 4 «Год выпуска (КР)» - указывается год выпуска, а в скобках год проведения последнего капитального ремонта,

– графа 5 «Единица измерения ресурса» - таблица 1 графа 2 соответственно наименованию изделия и составной части,

– графа 6 «Наработка с начала эксплуатации (после КР)» - для изделия это разность между годом, указанного в шапке, графы 10 и графой 4, в скобках указывается наработка после проведения последнего капитального ремонта,

– графа 7 «до СР» - разность между значениями таблицы 1 графы 5 и графой 6. Если был проведен капитальный ремонт, тогда в расчет брать наработку после капитального ремонта,

– графа 8 «до КР» - разность между значениями таблицы 1 графы 6 и графой 6. Если был проведен капитальный ремонт, тогда в расчет брать наработку после капитального ремонта,

– графа 9 «Ресурс, отпускаемый на год» - соответствует графе 3 таблицы 1 для обычной группы эксплуатации, графе 4 таблицы 1 для техники с повышенным расходом,

– графы 10 - 19 - в автоматическом режиме производится распределение мероприятий, подлежащих планированию по годам. Мероприятия указываются в дробном виде. В числителе указывается наименование мероприятия, а в знаменателе количество мероприятий в году.

Определение года проведения ТО-1, ТО-2 и выхода техники в средний и капитальный ремонт определяются следующими алгоритмами.

Определение года проведения ТО-1 определяется формулой (1):

$$N_{\text{доТО}-1} \leq (G_{\text{пл}} - G_{\text{послТО}-1}), \quad (1)$$

где $N_{\text{доТО-1}}$ – периодичность проведения ТО-1;

$G_{\text{пл}}$ – первый год периода в перспективном плане эксплуатации и выхода в ремонт;

$G_{послТО\text{-}1}$ – год последнего проведения ТО-1;

$G_{первТО\text{-}1}$ – год проведения первого ТО-1 за период указанный перспективном плане.

Если формула (1) истинна, тогда год проведения первого ТО-1 за указанный в плане период $G_{первТО\text{-}1} = G_{\text{пл}}$, иначе $G_{первТО\text{-}1} = G_{послТО\text{-}1} + N_{\text{доТО-1}}$. Последующие года проведения ТО-1 проводятся с периодичностью $N_{\text{доТО-1}}$.

Проведение года проведения ТО-2 соответствует формуле (2):

$$N_{\text{доТО}-2} \leq (G_{\text{пл}} - G_{\text{послТО}-2}), \quad (2)$$

где $N_{\text{доТО-2}}$ – периодичность проведения ТО-2;

$G_{\text{пл}}$ – первый год периода в перспективном плане эксплуатации и выхода в ремонт;

$G_{послТО\text{-}2}$ – год последнего проведения ТО-2;

$G_{первТО\text{-}2}$ – год проведения первого ТО-2 за период указанный перспективном плане. Последующие года проведения ТО-2 проводятся с периодичностью $N_{\text{доТО-2}}$.

Определение года проведения по остальным видам технического обслуживания и плановым мероприятиям производится по приведенному выше алгоритму.

Для определения количества проведения технический обслуживаний в году придерживаются следующих правил. Если значение в годах в таблице 1 графах 10-20 больше и равено 1, тогда знаменатель соответствующего мероприятия равен 1, иначе знаменатель соответствующего мероприятия равен 1 разделенная на значение соответсвующей графы.

При проведении ТО-2 (ТО-2х) в том числе проводятся и работы ТО-1 (ТО-1х), поэтому если в один год положено провести ТО-2 (ТО-2х) и ТО-1 (ТО-

1х) необходимо:

– если знаменатель ТО-1 (ТО-1х) равняется знаменателю ТО-2 (ТО-2х), тогда в данной графе ТО-1 (ТО-1х) не указывается, указывается ТО-2 (ТО-2х),

– если знаменатель ТО-1 (ТО-1х) больше знаменателя ТО-2 (ТО-2х), тогда в данной графе указываются ТО-1 (ТО-1х) и ТО-2 (ТО-2х), а в знаменателе ТО-1 (ТО-1х) указывается число меньшее на величину знаменателя ТО-2 (ТО-2х).

Достижение неснижаемого запаса ресурса определяется формулой (3):

$$G_{\text{НР}i} = G_{\text{пл}} + G_{\text{э}i}\ , \qquad (3)$$

где $G_{\text{НР}i}$ – год достижения неснижаемого запаса ресурса *i-го* образца;

$G_{\text{пл}}$ – первый год периода в перспективном плане эксплуатации и выхода в ремонт;

$G_{\text{э}i}$ – планируемый срок эксплуатации *i-го* образца, решается в соответствии с формулой (4).

$$G_{\text{э}i=}(N_{\text{доКР}i} - (N_{\text{нэ}i} + Z_{\text{НР}i}))/N_{\text{год}i}\ , \qquad (4)$$

где $N_{\text{доКР}i}$ – норма наработки до *i-го* капитального ремонта;

$N_{\text{нэ}i}$ – наработка *i-го* образца с начала эксплуатации или после последнего капитального ремонта;

$Z_{\text{НР}i}$ – установленный неснижаемый запас ресурса *i-го* образца;

$N_{год\text{э}i}$ – годовая норма расхода ресурса *i-го*.

Год выхода техники в средний и капитальный ремонт планируется по сроку службы. Если в один год у группы техники выход в ремонт по сроку эксплуатации будет более 25%, тогда 25% техники с наибольшим расходом ресурса планировать на данный год, а остальные переносятся на следующий, в котором происходит отбор по такому же алгоритму. Группу техники считать от 4 единиц и более. Для техники менее 4 единиц выход в ремонт не более 1 единицы в год.

Если на один год для одного образца выпадает СР и КР одновременно по сроку службы, тогда СР и КР планируются для этого образца по наработке.

Если образец выработал межремонтный ресурс и для СР и для КР, тогда остается КР.

2.2.2.2 Формирование второго раздела перспективного плана эксплуатации и выхода в ремонт

«Пояснительная записка» состоит из следующих графов:

– графа 1 «Планируемые работы с аббревиатурой мероприятий» - в первой строке проставляется показатель «В эксплуатации 15 лет и более», далее в строках указываются все запланированные в разделах плана мероприятия,

– графа 2 «Наименование техники» - в шапке формы указываются вся номенклатура, приведенная в первом и втором разделах мероприятия. В строке «В эксплуатации 15 лет и более» проставляется количество техники одноименной номенклатуры, у которых разница между годом, указанным в шапке данного раздела и годом выпуска, указанным в графе 4 разделов 1 и 2 плана больше или равно 15. В остальных строках суммируются мероприятия по одноименной номенклатуре.

2.2.2.3 Формирование третьего раздела перспективного плана эксплуатации и выхода в ремонт

Раздел «Исходные данные для планирования» формируется в автоматическом режиме и содержит:

– графа 1 «Аббревиатура мероприятий»,

– графа 2 «Проводимые мероприятия»,

– графа 3 в шапке формы указываются вся номенклатура, а в строках периодичность проведения мероприятий (1 раз в … лет), которая определена в таблице 1.

2.3 Задача № 2 «Формирование годового плана эксплуатации и выхода в ремонт техники»

На основании данных перспективного плана разрабатывается годовой план.

Годовой план эксплуатации и выхода в ремонт является основным документом, определяющим использование техники в течение года. Основные цели разработки:

– поддержание техники в рабочем состоянии,

– обеспечение исправной техникой,

– контроль за расходом годового ресурса,

– своевременное проведение ТО и ремонта,

– своевременное обеспечение запасными частями и принадлежностями для проведения ТО и ремонта,

– равномерная загрузка ремонтно-восстановительных органов.

К основным задачам планирования относятся:

– определение времени использования техники,

– распределение годового лимита расхода ресурсов, в соответствии с установленными нормами по месяцам года,

– установление сроков выхода техники в ТО по месяцам года,

– планирование выполнения работ ремонтными органами.

Разработка плана предусматривает распределение годового лимита расхода ресурсов, выделяемых, в соответствии с установленными нормами по месяцам для образцов всех групп эксплуатации.

Годовой план эксплуатации и выхода в ремонт техники состоит из двух разделов:

– первый раздел содержит мероприятия ТО и ремонта основной номенклатуры,

– второй раздел является пояснительной запиской для обобщения планируемых мероприятий.

В выходном документе расход ресурсов указывается по каждому образцу в отдельности с распределением его по месяцам и обобщением за каждое полугодие и год.

Выход образцов обычной группы эксплуатации в ТО в годовом плане указывается по месяцам года на основе данных перспективного плана эксплуатации и выхода в ремонт техники.

При составлении годового плана эксплуатации и выхода в ремонт необходимо руководствоваться следующими принципами:

– недопущение перерасхода ресурсов сверх установленных норм,

– планирование эксплуатации образцов в пределах минимального межремонтного ресурса,

– равномерный выход образцов в ремонт в течение года.

Из общего годового лимита расхода ресурсов на образцы техники необходимо выделять резерв на проведение итоговых и контрольных проверок.

2.3.1 Входная информация

Источниками входной информации для настоящей задачи являются:

– информация о наименованиях подразделений, численности техники,

– данные о наличии и качественном состояния техники,

– данные об общей наработке,

– сведения о проведенных технических обслуживаниях и ремонтах,

– установленные годовые нормы расхода ресурсов и периодичность проведения мероприятий,

– перспективный план эксплуатации и выхода в ремонт техники.

Помимо этого, пользователь должен иметь возможность задавать трудоемкость и количество требуемых специалистов для проведения технического обслуживания, определять основные и резервные дни проведения ТО по месяцам, а также очередность проведения по подразделениям.

2.3.2 Формирование выходного документа

Годовой план эксплуатации и выхода в ремонт формируется после разработки перспективного плана эксплуатации и выхода в ремонт.

2.3.2.1 Формирование первого раздела годового плана эксплуатации и выхода в ремонт

Первый раздел формы подразделяется на два подраздела:

1) техника обычной группы эксплуатации,
2) техника с повышенным расходом ресурса.

И содержит следующие графы:

– графа 1 «Подразделение»,

– графа 2 «Марка машины»,

– графа 3 «Заводской номер» - соответствует графе 3 перспективного плана,

– графа 4 «Условный номер»;

– графа 5 «Год выпуска (КР)» - соответствует графе 4 перспективного плана,

– графа 6 «Группа эксплуатации»,

– графа 7 «Единица измерения наработки» - соответствует графе 5 перспективного плана,

– графа 8 «Наработка с начала эксплуатации (после КР)» - соответствует графе 6 перспективного плана,

– графа 9 «до очередного ремонта» - если в планируемом году спланировано проведение СР, тогда соответствует графе 7 перспективного плана, иначе соответствует графе 8 перспективного плана,

– графа 10 «до очередного ТО» - вычисляется по формуле (5):

$$Z_{\text{рес(ТО}i)} = N_{\text{доТО}i} - N_{\text{послеТО}}\ , \qquad (5)$$

где $Z_{\text{рес(ТО}i)}$ – запас ресурса до очередного *i-го* технического обслуживания;

$N_{\text{доТО}i}$ – норма наработки до *i-го* технического обслуживания;

$N_{\text{послеТО}}$ – наработка после последнего технического обслуживания.

– графа 11 «Дата последнего ТО»,

– графа 12 «Дата последнего РТО»,

– графа 13 «Дата постановки на хранение»,

– графа 14 «Годовая норма расхода ресурсов» - соответствует графам 3 и 4 таблицы 1,

– графа 15 «Резерв годового ресурса» - составляет 5% от годового ресурса,

– графа 16 «Разрешенный расход ресурса на год» - равен 95% годового расхода,

– графа 17-22 «декабрь … май» - алгоритм формирования приведен ниже,

– графа 23 «Итого за первое полугодие» - сумма граф 17-22,

– графа 24-29 «июнь … ноябрь» - алгоритм аналогичен графам 17-22,

– графа 30 «Итого за второе полугодие» - сумма граф 24-29;

– графа 31 «Итого за год» - сумма графы 23 и 30.

Планирование технического обслуживания по месяцам производится в автоматическом режиме в следующей очередности:

– техника с повышенным расходом ресурса,

– техника требующее проведения ТО-2 обычной группы эксплуатации,

– остальная техника обычной группы эксплуатации.

Очередность среди подразделений внутри групп эксплуатации и период проведения обслуживания в каждом месяце устанавливается пользователь при помощи веб-интерфейса. Также для планирования проведения технического обслуживания пользователь должен установить:

– трудоемкость для каждого образца техники,

– количество специалистов для каждого образца техники.

Распределение плановых технических обслуживаний по месяцам и дням года производится в автоматическом режиме на основании показателей D_{TOi} и K_{TOi} по описанной выше очередности. Период проведения ТО должен планироваться в рамках одного месяца. Если дней в месяце для проведения, ТО

недостаточно, тогда период учитывает резервные дни. Если недостаточно и с учетом резервных дней, тогда этот образец переносится на следующий месяц.

Для определения периода проведения ТО придерживаются следующих правил. Если T_{TOi} / $S_{рвоi} \geq 8$, тогда определяем период проведения D_{TOi} по формуле (6):

$$D_{\text{ТО}i} = T_{\text{ТО}i} / (8 * S_{\text{рво}i}) , \quad (6)$$

где D_{TOi} – количество дней необходимых для проведения технического обслуживание *i-го* образца;

T_{TOi} – трудоемкость *i-го* технического обслуживания *i-ой* номенклатуры;

$S_{рвоi}$ – количество специалистов, привлекаемых для обслуживания *i-ой* номенклатуры.

Если же T_{TOi} / $S_{рвоi} < 8$, тогда определяем количество проведенных ТО одной номенклатуры за один день по формуле (7):

$$K_{\text{ТО}i} = (8 * S_{\text{рво}i})/T_{\text{ТО}i} , \quad (7)$$

где K_{TOi} – количество проведенных технических обслуживаний *i-ой* номенклатуры за одон день.

Период проведения обслуживания определяется по формуле (8):

$$D_{\text{ТО}i} = S_{\text{ВВТ}i}/K_{\text{ТО}i} , \quad (8)$$

где D_{TOi} – количество дней необходимых для проведения технического обслуживание *i-го* образца;

K_{TOi} – количество проведенных технических обслуживаний *i-ой* номенклатуры за один день;

T_{TOi} – трудоемкость *i-го* технического обслуживания *i-ой* номенклатуры;

$S_{ВВТi}$ – количество *i-ой* номенклатуры в *i-ом* подразделении;

$S_{рвоi}$ – количество специалистов, привлекаемых для обслуживания *i-ой* номенклатуры.

В графе ТО записывается в виде дроби. В числителе проставляется вид ТО, а в знаменателе период проведения обслуживания. ТО не должно быть назначено ранее 5 месяцев от даты, указанной в графе 11.

Для планирования расхода ресурса пользователь должен установить процентное соотношение расхода ресурса за полугодия на основе данных графы 16. Затем установить процентное соотношение в полугодии по месяцам. Распределение осуществляется с учетом запланированных мероприятий на год.

2.3.2.2 Формирование второго раздела годового плана эксплуатации и выхода в ремонт

Пояснительная записка годового плана эксплуатации и выхода в ремонт включает в себя следующие данные:

– графа 1 «Марка машины» - марки машин, указанные в графе 2 годового плана,

– графа 2 «Количество машин» - количество машин одинаковой номенклатуры указанные в графе 2 плана,

– графа 3 «Ресурс на год» - сумма графы 14 плана одинаковой номенклатуры с единицей измерения км (м/ч),

– графа 4-15 - сумма одноименных граф 16-25 и граф 26-37;

– графа 16 «Всего ТО» - сумма граф 17-25;

– графы 17-25 - сумма соответствующего вида технического воздействия указанных в графах 17-22 и 24-29 плана;

– графа 26 «Всего ТО» - сумма граф 27-37;

– графы 27-37 - заполняются в ручном режиме. Значение в графах 17-25 уменьшаются на величину значения соответствующих граф.

2.4 Описание функций

Разработанное приложение построено на клиент-серверной архитектуре, поэтому содержит ряд функций, которые описывают взаимодействия пользователя с интерфейсом, так и решают поставленную задачу на стороне сервера.

2.4.1 Описание функций веб-сервера

Для формирования перспективного и годового планов, записи и получении данных из базы данных, на сервере предусмотрены ряд функций.

load_nsi_grid() предназначена для загрузки данных из базы данных для таблицы в пункте меню «Установленные годовые нормы расхода ресурсов и периодичность проведения планируемых мероприятий». Функция загружает из базы данных информацию о имеющихся в наличии образцов техники и установленных для них норм расхода ресурсов. После чего отправляет полученную информацию пользователю на веб-интерфейс, где обрабатывается для построения элемента интерфейса.

update_nsi_grid() вызывается в случаи внесения изменений в таблице «Установленные годовые нормы расхода ресурсов и периодичность проведения планируемых мероприятий». Осуществляет внесение изменений в базу данных на основе полученных данных, таких как идентификатор строки, имя поля и новое значение ячейки.

Функция load_trudoemkost_to() загружает информацию о трудоемкости выполняемых технических обслуживаний и количестве требуемых специалистов для каждого экземпляра машины.

update_trudoemkost_to() осуществляет запись изменений л трудоемкости и количестве специалистов в базу данных.

Функции load_date_to и update_date_to предназначены для работы с загрузкой и установкой новых сроков проведения технического обслуживания по месяцам. А load_queue_to() и update_queue_to() позволяют предопределить очередность проведения ТО среди подразделений.

build_podrazdel_tree() загружает информацию о подчиненных подразделениях на основе которой строится структурное дерево, представленное в веб-интерфейсе в разделе «Перспективный план эксплуатации и выхода в ремонт».

build_plan_db() – функция для формирования перспективного плана эксплуатации и выхода в ремонт. Данная функция включает в себя ряд других функций:

– start_planing_one() – определяет год проводимых мероприятий помимо достижения неснижаемого ресурса и выходе техники в средний и капитальный ремонт,

– planing_nr() – устанавливает срок достижений неснижаемого ресурса,

– planinig_sr_kr – осуществляют расчет выхода экземпляров техники в средний и капитальный ремонт в соответствии с требованиями о процентном соотношении нахождении в ремонте,

– print_plan_one() – формирует первый раздел выходного документа,

– print_pz() – формирует пояснительную записку итогового документа,

– print_data() – завершает формирование выходного документа добавлением в него третьего раздела.

build_year_plan() – формирует годовой план эксплуатации и выхода в ремонт. Функция также как и build_plan_db() основывается на выполнении ряда других: create_year_db() производит расчеты и запись в базу, а print_year_plan() формирует документ.

При создании документа применяется рад вспомогательных функций, предназначенных для печати:

– get_project_root() – определяет путь к шаблону документа,

– create_cell() – формирует ячейку в таблице,

– print_empty_row() – добавляет пустую строку в случае отсутствия данных,

– print_single_string() – производит печать строки с данными.

Документы, формируемые функциями build_plan_db() и build_year_plan() не хранятся на сервере, а сохраняются в байтах в буфере в памяти, а после

отправляются клиенту, где уже происходит преобразование в документ с разрешением xlsx.

2.4.2 Описание функций веб-интерфейса

При взаимодействии пользователя с интерфейсом происходит ряд событий, которые обрабатываются языком программирования JavaScript.

Нажатие пользователем на пункт меню обрабатывается событием *on click* в результате чего вызывается функция, открывающая соответствующий пункт меню. Если до этого какой-то пункт меню уже был открыт, то производится его закрытие.

При внесении изменений в таблицы «Установленные годовые нормы расхода ресурсов и периодичность проведения планируемых мероприятий» и «Трудоемкость проведения ТО» происходит проверка, что значение является числовым и отличается от предыдущего. В случае успеха значение изменяется и отправляется запрос на серверную часть приложения для внесения изменений в базу данных, где выполняются функции update_nsi_grid() и update_trudoemkost_to() соответственно, иначе остается прежним.

При использовании кнопки поиск в одноименных пунктах меню производится проверка что строка поиска содержит данные после чего выделяется первая срока соответствующая поиску. Последующие нажатия осуществляют поиск дальнейших совпадений. Если соответствий не находится, срока поиска выделяется красным.

Изменение периода проведения ТО в пункте меню «Дни проведения ТО» и места в очереди подразделений в пункте «Очередность проведения ТО среди подразделений» приводит к вызову функций update_date_to() и update_queue_to() на сервере.

Строка поиска в пункте «Перспективный план эксплуатации и выхода в ремонт» осуществляется выборку среди элементов структурного дерева и его перерисовку в соответствии с результатами поиска. А нажатие кнопки «Произвести расчет перспективного плана» отправляет запрос на формирование итогового документа.

Изменение значений ползунков в меню «Годовой план эксплуатации и выхода в ремонт» приводит к уменьшению максимального значения других, а нажатие кнопки «Произвести расчет годового плана» к формированию плана веб-сервером.

Помимо событий, которые обрабатывают все взаимодействия с интерфейсом применяются функции для первоначальной загрузки данных на странице:

– update_grid_data() – производит загрузку данных и построение таблицы «Установленные годовые нормы расхода ресурсов и периодичность проведения планируемых мероприятий»,

– build_trudoemkost_grid() – строит таблицу «Трудоемкость проведения ТО» на основе полученных данных,

– build_date_grid() – получение информации о периодах проведения ТО,

– build_queue_grid() – построение списка подразделений в соответствии с полученной из базы данных очередность.

2.5 Описание работы программы

При взаимодействии пользователя с интерфейсом выполняется ряд действий, который показан на рисунке 4. Пользователь вносит изменения в нормативно справочной информации или выполняет расчет задачи, на основе чего формирует GET или POST запрос, включающий в себя URL и данные, используемые в вычислениях на сервере. После чего запрос отправляется на сервер, где на основе получено URL определяется порядок выполняемых действий. В случае изменения данных НСИ производиться запись в базу данных новых значений, а при выполнении расчета задачи планирования получение данных. Следующим шагом является формирование ответа на полученный запрос и его отправка обратно веб-браузеру. На основе данного описания была разработана UML-диаграмма последовательности, которая показана на рисунке 4.

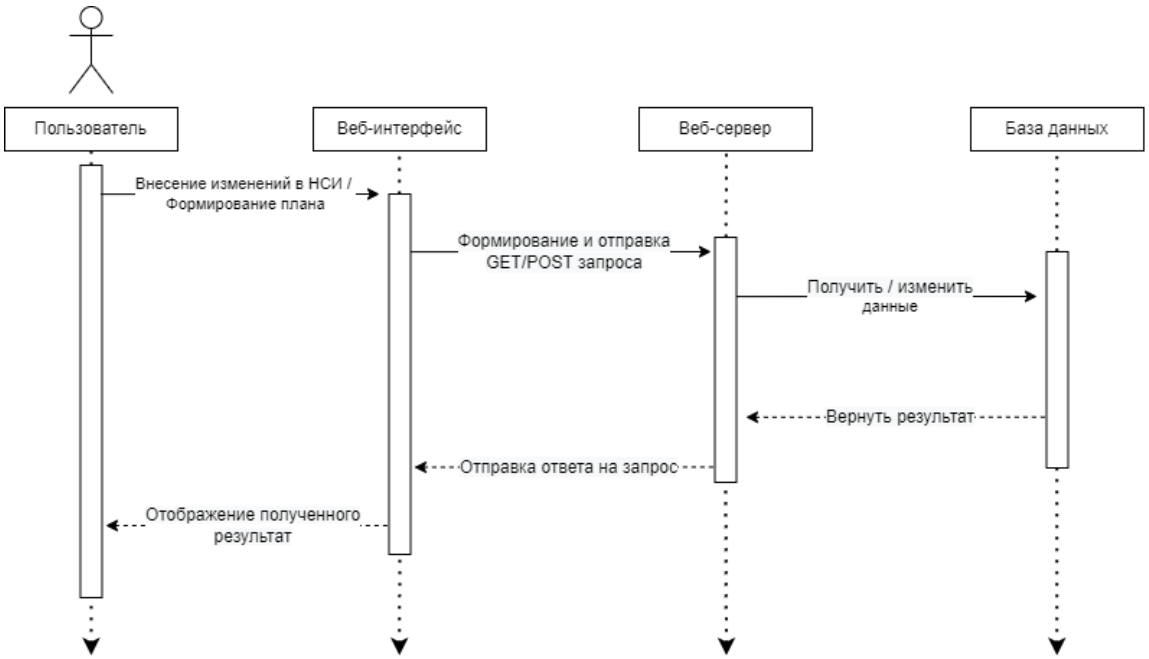


Рисунок 4 - UML-диаграмма последовательности

На рисунке 5 в виде UML-диаграммы показаны возможности, предоставляемые пользователю при взаимодействии с приложением.

Приложение позволяет пользователю выполнить следующие действия:

– установленные годовые нормы расхода ресурсов и периодичность проведения мероприятий – открытие данного пункта меню предоставляет пользователю осуществлять внесение изменений в одноимённую таблицу, а также использовать поиск для удобного перемещения по таблице,

– трудоемкость проведения ТО – позволяет изменить показатели трудоемкости и количества специалистов для проведения технического обслуживания,

– дни проведения ТО – пользователь может заполнить основной и резервный период проведения ТО,

– очерёдность проведения ТО среди подразделений – позволяет определить порядок подразделений для проведения ТО,

– перспективный план эксплуатации и выхода в ремонт – позволяет пользователю сформировать перспективный план. Для этого необходимо выбрать интересующие подразделения и нажать кнопку «Произвести расчет перспективного плана»,

– годовой план эксплуатации и выхода в ремонт – позволяет пользователю сформировать годовой план. Для этого необходимо распределить годовой ресурс по месяцам и нажать кнопку «Произвести расчет годового плана».

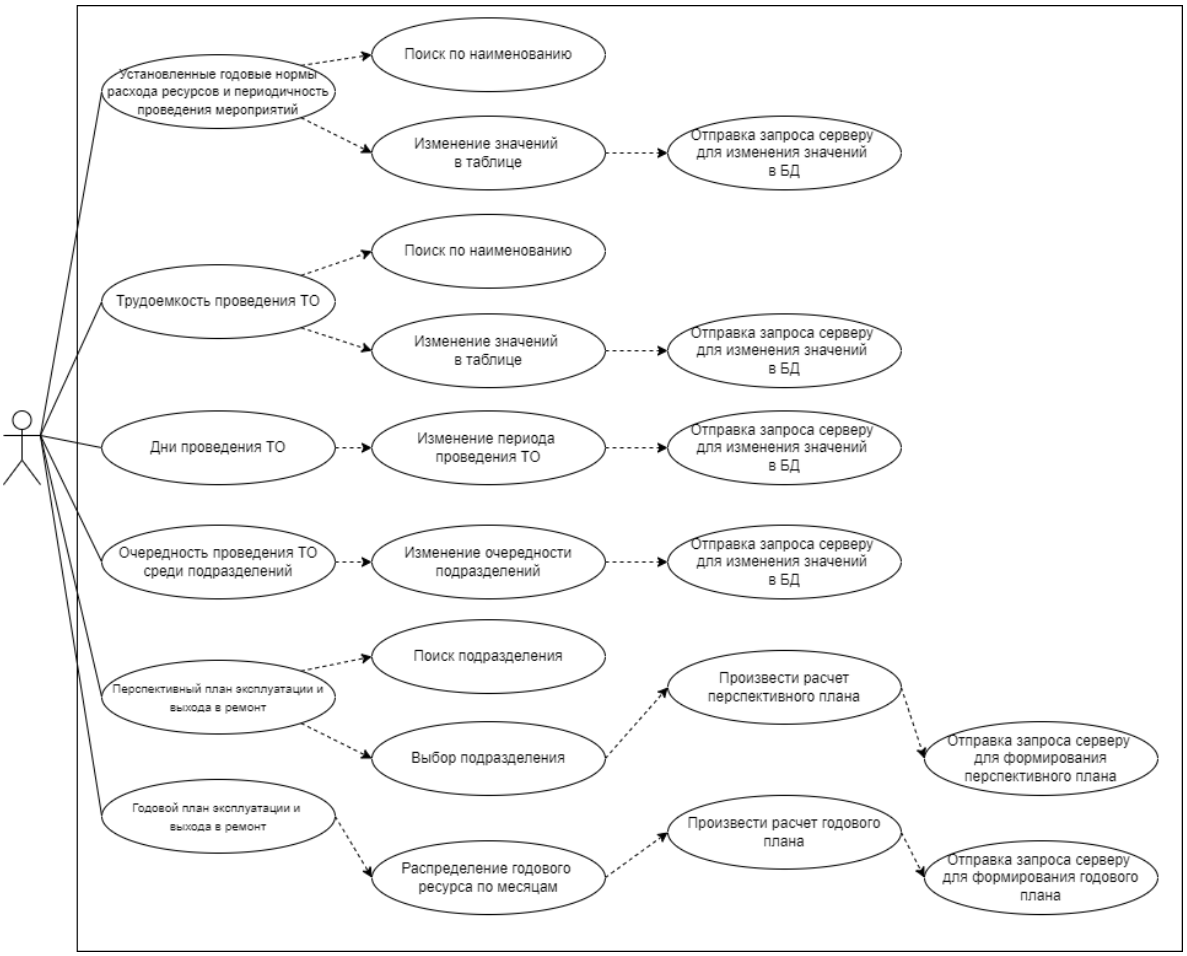


Рисунок 5 - UML-диаграмма вариантов использования

Этапы, через которые проходит интерфейс пользователя при формировании планов, отображает диаграмма деятельности, показанная на рисунке 6.

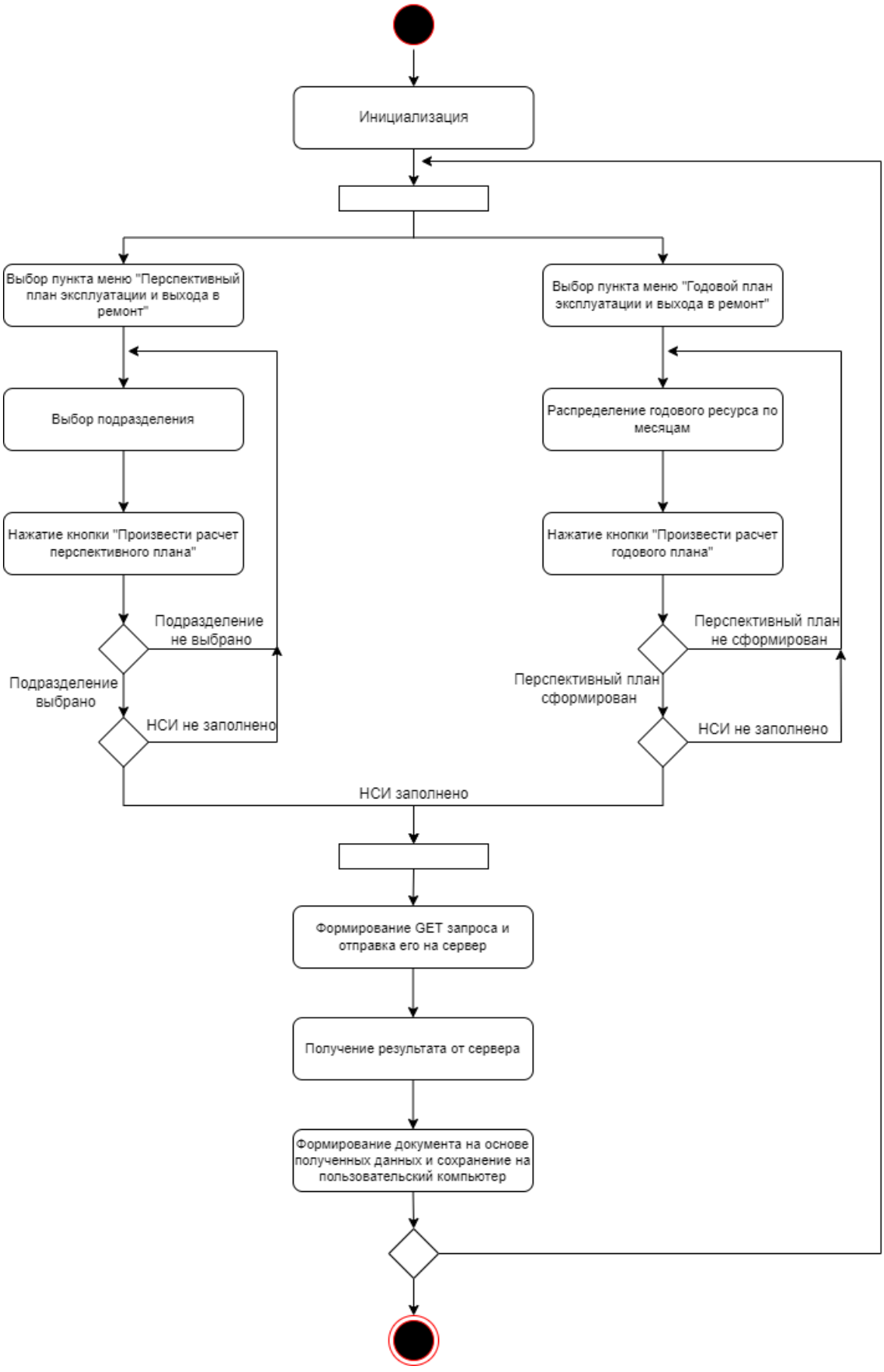


Рисунок 6 - UML-диаграмма деятельности

Диаграмма деятельности демонстрирует описание процесса формирования перспективного и годового плана со стороны пользователя.

Если формируется перспективный план, то в первую очередь определяются необходимые подразделения и нажимается кнопка «Произвести расчет перспективного плана». После чего проверяется было ли выбрано хотя бы одно подразделение, если нет, то выводится сообщение на экран об ошибке. В противном случае проверяется правильность заполнения нормативно-справочной информации, в случае успеха переходит к следующему этапу.

Если формируется годовой план, то сначала распределяется годовой ресурс и после нажимается кнопка «Произвести расчет годового плана». Далее проверяется был ли сформирован перспективный план, если нет, то выводится сообщение на экран, иначе проверяется правильность заполнения НСИ.

Последующие пункты идентичны для обоих планов. Веб браузер формирует GET запрос к серверу для формирования плана и отправляет его. После ожидает получение ответа и на основе полученных данных формирует xlsx документ, который сохраняется пользователю на компьютер.

2.6 Отладка и тестирование

Отладка приложения проводилась на компьютере без предустановленного необходимого программного обеспечения. Для развертывания разработанного приложения на таком компьютере было использовано ПО Docker, позволяющее разворачивать контейнеры внутри которых находится весь набор необходимых программ [8].

Для работы разработанного приложения необходимо развернуть три контейнера содержащих следующих образы:

– образ разработанного приложения,

– база данных postgresql,

– графический интерфейс для работы с базой данных pgadmin, является необязательным.

Для запуска контейнеров использовался Docker Compose, параметры которого описаны в файле docker-compose.yml. Для создания контейнера с разработанным приложением в файле был описан сервис web (рисунок 7).

```
web:
  container_name: web
  build: .
  command: python "Web server"/manage.py runserver 0.0.0.0:8000
  volumes:
    - .:/usr/src/app
  ports:
    - "8000:8000"
  depends_on:
    - db
  networks:
    - postgres
```

Рисунок 7 - Контейнер разработанного приложения

Данный контейнер создается на основе образа приложение описанного в файле Dockerfile (рисунок 8).

```
Dockerfile > ...
1   FROM python:3.8
2   WORKDIR /usr/src/app
3   COPY requirements.txt ./
4   RUN pip install -r requirements.txt
5
```

Рисунок 8 - Dockerfile разработанного приложения

Из рисунка видно, что образ создается на основе python 3.8, после чего происходит выбор рабочей директории, копирование файла requirements.txt с локальной машины в рабочую директорию контейнера и выполняется установка всех описанных в данном файле библиотек.

Контейнеры для базы данных и графического интерфейса для доступа к ней строятся на основе образов, скачанных из интернета.

В результате использования контейнеров, удалось запустить приложение на неподготовленном для этого компьютере.

В ходе тестирования приложения были выявлены проблемы, которые необходимо было решить. Одной из проблем было то, что при запуске контейнеров не были установлены все необходимые для работы библиотеки,

из-за чего запуск приложения не удавалось осуществить. Решением послужило дополнение списка библиотек в файле requirements.txt.

Другой проблемой стала очередность запуска контейнеров. Так как для работы приложения и графического интерфейса базы данных необходима запущенная БД в файле docker-compose.yml для сервисов был добавлен параметр depends_on, позволяющий определить зависимости между сервисами и очередность их запуска.

Особенностью использования контейнеров является то, что после их остановки, все внесенные изменения не сохраняются. Для решения данной проблемы использовались volumes (томы), позволяющие восстановить все внесенные изменения после перезапуска контейнера.

Также вовремя тестирования в приложения были добавлены сообщения об ошибках, что позволяет пользователю понимать, что он сделал не так. Помимо этого, исправлены мелкие недочеты в выходном документе.

После предпринятых мер разработанное приложение работало надежно и формировало перспективный и годовой план без ошибок.

2.7 Выводы

В данном разделе рассмотрены задачи, выполняемые разработанным приложением. Описаны основные функции клиентской и серверной части веб-приложения. При помощи UML-диаграмм представлены особенности взаимодействия пользователя с ПО, а также рассмотрены проблемы, возникшие на этапе отладки и тестирования.

3 Инструкция пользователя

При запуске программы перед пользователем появляется окно загрузки, информирующее о загрузке информации из базы данных (рисунок 9).

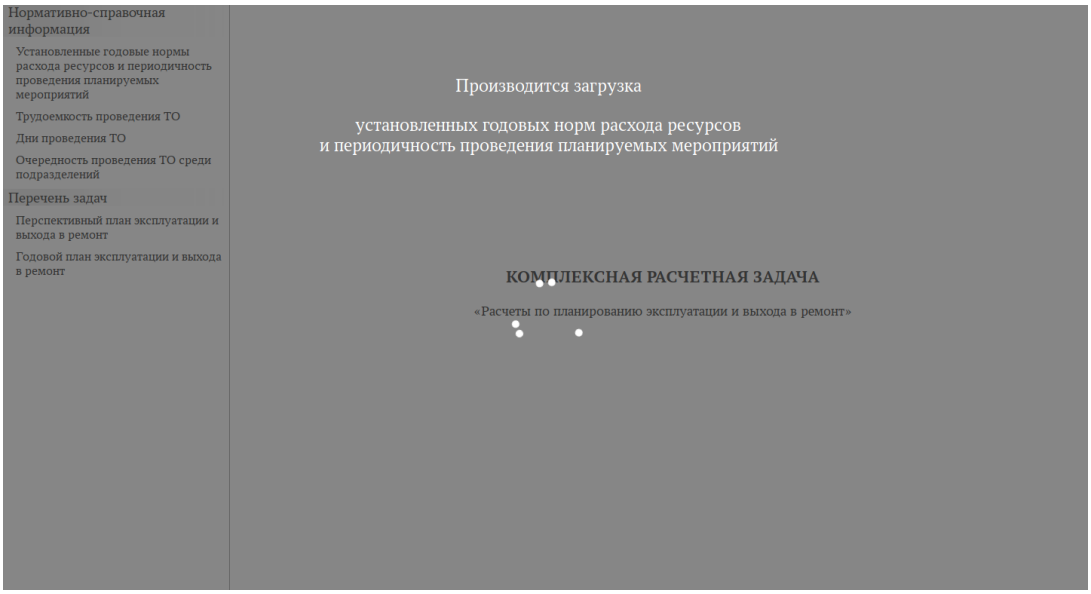


Рисунок 9 - Окно загрузки данных из базы данных

В течении данного процесса производится загрузка следующей информации:

– установленные годовые норма расхода ресурсов и периодичность проведения мероприятий,

– трудоемкость проведения ТО,

– дни проведения ТО,

– очередность проведения ТО среди подразделений,

– дерево подразделений.

После загрузки перед пользователем открывается основное окно программы, представленное на рисунке 10. В левой части находится меню, а в правой основное рабочее пространство.

Рисунок 10 - Главное окно программы

Меню содержит следующие пункты:

– установленные годовые нормы расхода ресурсов и периодичность проведения планируемых мероприятий,

– трудоемкость проведения ТО,

– дни проведения ТО,

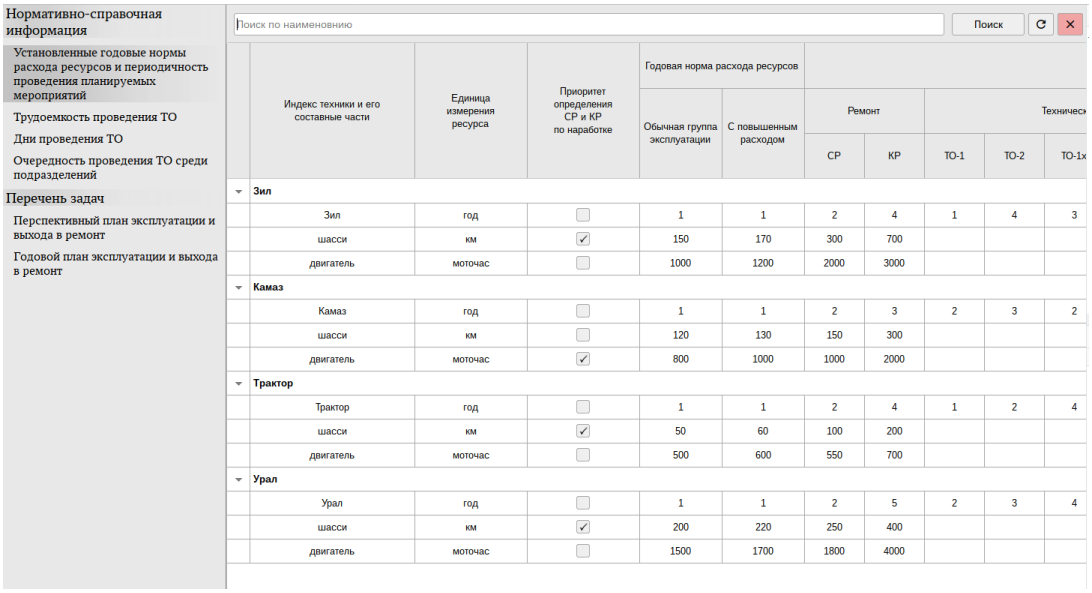
– очередность проведения ТО среди подразделений,

– перспективный план эксплуатации и выхода в ремонт,

– годовой план эксплуатации и выхода в ремонт.

3.1 Установленные годовые нормы расхода ресурсов и периодичность проведения планируемых мероприятий

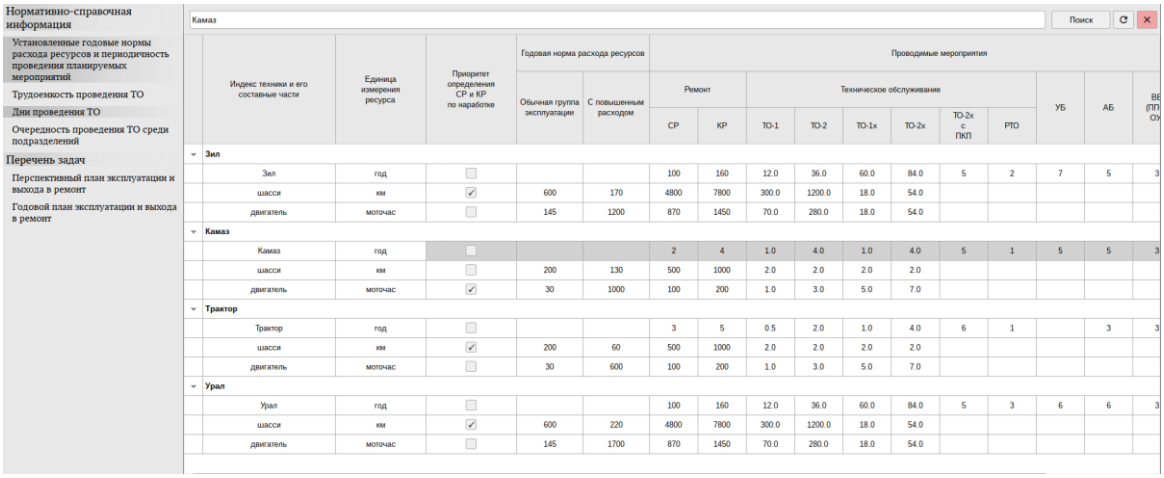
Данный пункт меню открывает таблицу с образцами техники, находящихся в данном или подчиненных подразделениях, их нормами расхода ресурсов и периодичностью мероприятий (рисунок 11). Обязательными к заполнению являются поля «годовая норма расхода ресурсов», «СР» и «КР». Также в поле «приоритет определения СР и КР по наработке» пользователь должен определить на основе какой составной части будет производится сравнение по наработке между одинаковыми видами техники.



Индекс техники и его составные части	Единица измерения ресурса	Приоритет определения СР и КР по наработке	Годовая норма расхода ресурсов						
			Обычная группа эксплуатации	С повышенным расходом	Ремонт		Техническ		
					СР	КР	ТО-1	ТО-2	ТО-1х
Зил									
Зил	год	☐	1	1	2	4	1	4	3
шасси	км	☑	150	170	300	700			
двигатель	моточас	☐	1000	1200	2000	3000			
Камаз									
Камаз	год	☐	1	1	2	3	2	3	2
шасси	км	☐	120	130	150	300			
двигатель	моточас	☑	800	1000	1000	2000			
Трактор									
Трактор	год	☐	1	1	2	4	1	2	4
шасси	км	☑	50	60	100	200			
двигатель	моточас	☐	500	600	550	700			
Урал									
Урал	год	☐	1	1	2	5	2	3	4
шасси	км	☑	200	220	250	400			
двигатель	моточас	☐	1500	1700	1800	4000			

Рисунок 11 - Пункт меню «Установленные годовые нормы расхода ресурсов и периодичность проведения планируемых мероприятий»

Поиск по наименованию, расположенный вверху страницы позволяет быстро перемещаться по таблице. В случае нахождение элемента соответствующего параметрам поиска он выделяется в таблице как показано на рисунке 12.



Индекс техники и его составные части	Единица измерения ресурса	Приоритет определения СР и КР по наработке	Годовая норма расхода ресурсов		Проводимые мероприятия									
			Обычная группа эксплуатации	С повышенным расходом	Ремонт		Техническое обслуживание						УБ	АБ
					СР	КР	ТО-1	ТО-2	ТО-1х	ТО-2х	ТО-2х с ПКП	РТО		
Зил														
Зил	год	☐			100	160	12.0	36.0	60.0	84.0	5	2	7	5
шасси	км	☑	600	170	4800	7800	300.0	1200.0	18.0	54.0				
двигатель	моточас	☐	145	1200	870	1450	70.0	280.0	18.0	54.0				
Камаз														
Камаз	год	☐			2	4	1.0	4.0	1.0	4.0	5	1	5	5
шасси	км	☐	200	130	500	1000	2.0	2.0	2.0	2.0				
двигатель	моточас	☑	30	1000	100	200	1.0	3.0	5.0	7.0				
Трактор														
Трактор	год	☐			3	5	0.5	2.0	1.0	4.0	6	1		3
шасси	км	☑	200	60	500	1000	2.0	2.0	2.0	2.0				
двигатель	моточас	☐	30	600	100	200	1.0	3.0	5.0	7.0				
Урал														
Урал	год	☐			100	160	12.0	36.0	60.0	84.0	5	3	6	6
шасси	км	☑	600	220	4800	7800	300.0	1200.0	18.0	54.0				
двигатель	моточас	☐	145	1700	870	1450	70.0	280.0	18.0	54.0				

Рисунок 12 - Поиск техники в таблице

Если список найденных элементов достиг конца или элементы не были найдены, то строка поиска выделяется красным цветом (рисунок 13).

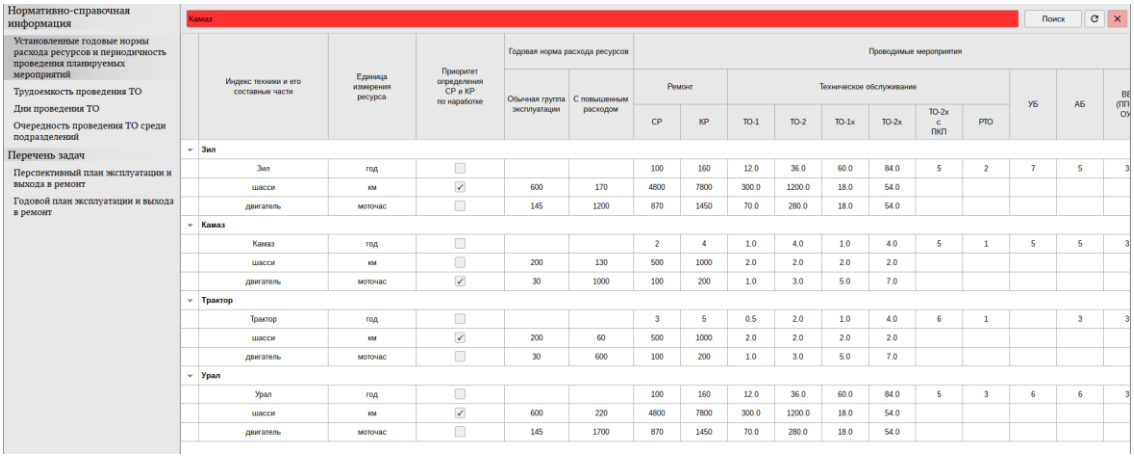


Рисунок 13 - Элемент не найден

3.2 Трудоемкость проведения ТО

Это окно (рисунок 14) позволяет задать пользователю трудозатраты проведения технического обслуживания первого и второго типов, а именно трудоемкость в часах и количество необходимых специалистов для проведения обслуживания. А поиск работает аналогичном образом как в предыдущем пункте меню.

Рисунок 14 - Пункт меню «Трудоемкость проведения ТО»

3.3 Дни проведения ТО

Предоставляет возможность пользователю задать интервалы проведения технического обслуживания, а также установить резервные дни (рисунок 15).

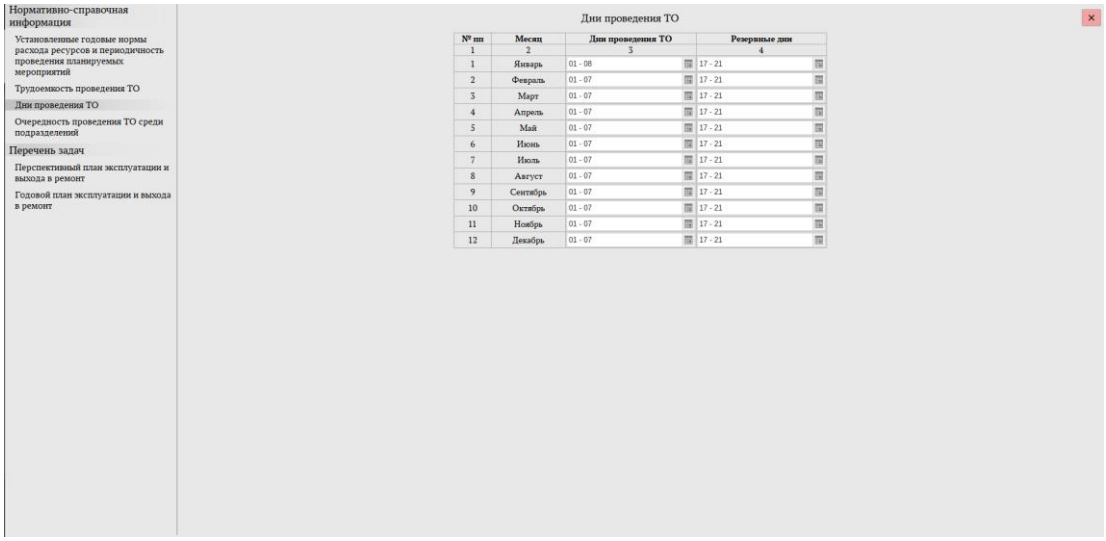


Рисунок 15 - Пункт меню «Дни проведения ТО»

При выборе иконки календаря справа от поля ввода, появляется календарь на выбранный месяц планируемого года и предлагает выбрать интервал проведения мероприятий (рисунок 16).

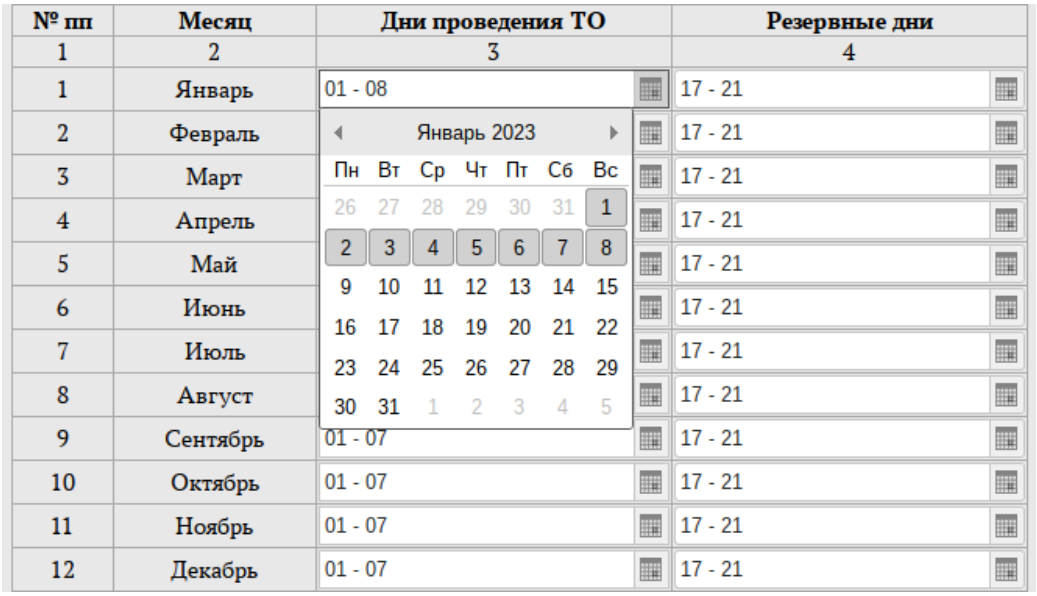


Рисунок 16 - Выбор основных дней проведения ТО

Резервные дни могут быть выбраны в интервале после завершения основных дней до окончания месяца (рисунок 17).

Рисунок 17 - Выбор резервных дней проведения ТО

3.4 Очередность проведения ТО среди подразделений

Приложение позволяет определить в каком порядке подразделения будут отправлены в техническое обслуживание в планируемом году. Для изменения позиции пользователю необходимо перетащить выбранное подразделение на нужное место (рисунок 18).

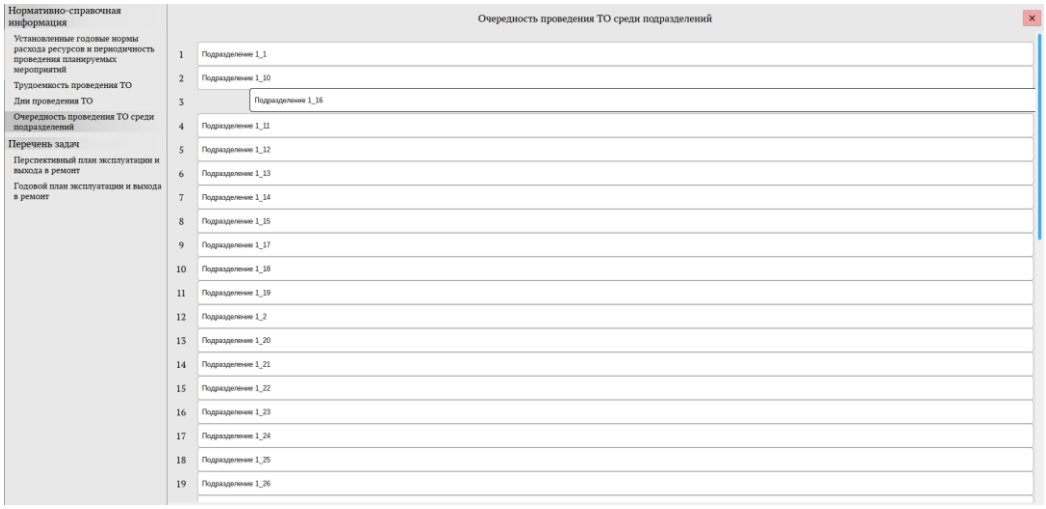


Рисунок 18 - Изменение очередности проведения ТО среди подразделений

3.5 Перспективный план эксплуатации и выхода в ремонт

Данный пункт позволяет выполнить расчетную задачу и сформировать перспективный план эксплуатации и выхода в ремонт. Все действия, которые необходимо выполнить пользователю, это выбрать подчинённые

подразделения, которые необходимо учитывать при формировании плана и нажать соответствующую кнопку запуска расчета задачи (рисунок 19).

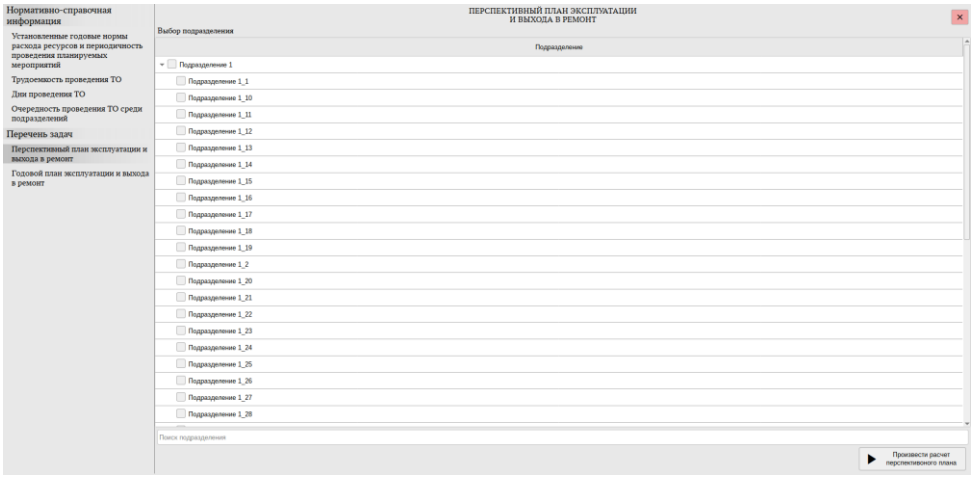


Рисунок 19 - Пункт меню «Перспективный план эксплуатации и выхода в ремонт»

Если пользователь не выбрал ни одного подразделения, то он получает соответствующе сообщение об ошибке (рисунок 20).

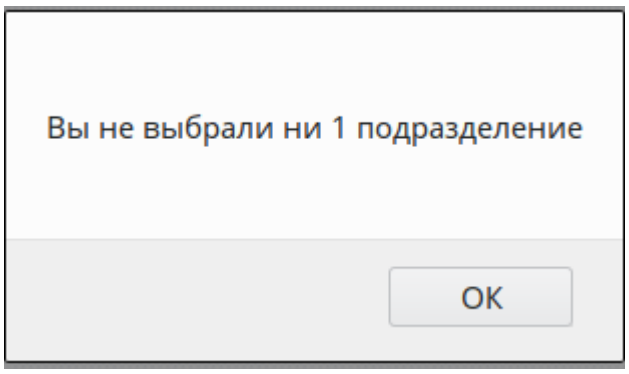


Рисунок 20 - Ошибка выбора подразделения

Для удобства поиска подразделений предусмотрен поиск, осуществляющий отсеивание неподходящих вариантов. Его использование продемонстрированно на рисунке 21.

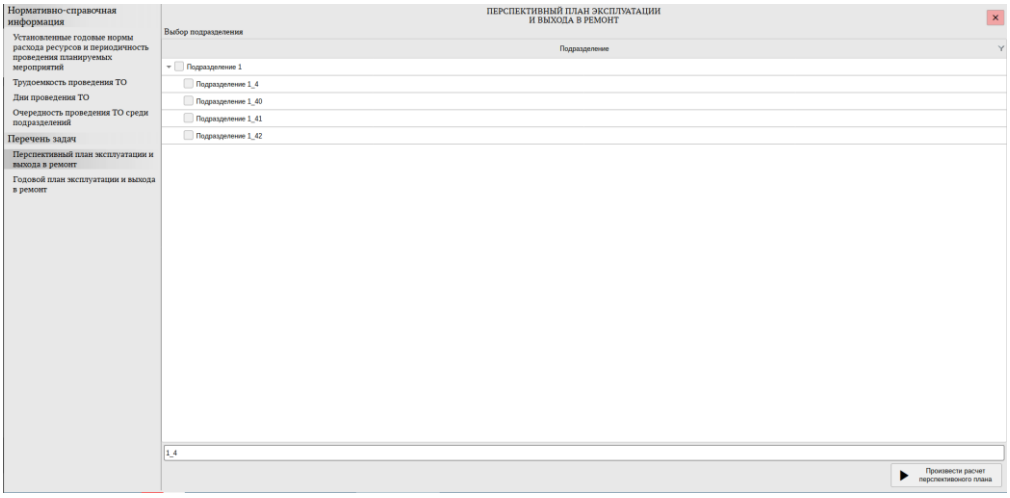


Рисунок 21 - Поиск по подразделениям

После запуска расчета плана перед пользователем появляется экран ожидания формирования плана как на рисунке 22.

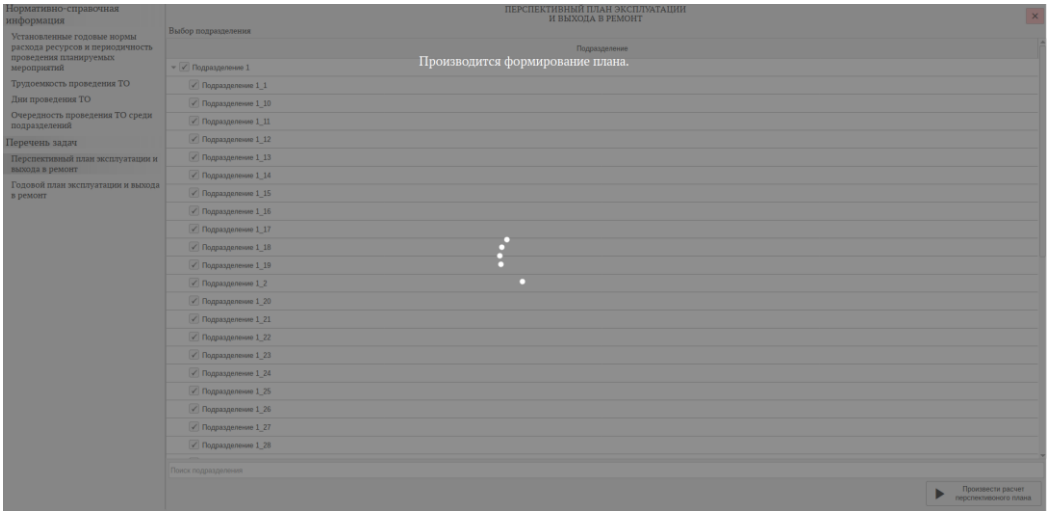


Рисунок 22 - Экран ожидания формирования плана

Результатом выполнения является получение пользователем xlsx документа (рисунок 23), структура которого представлена на рисунках 24 – 26.

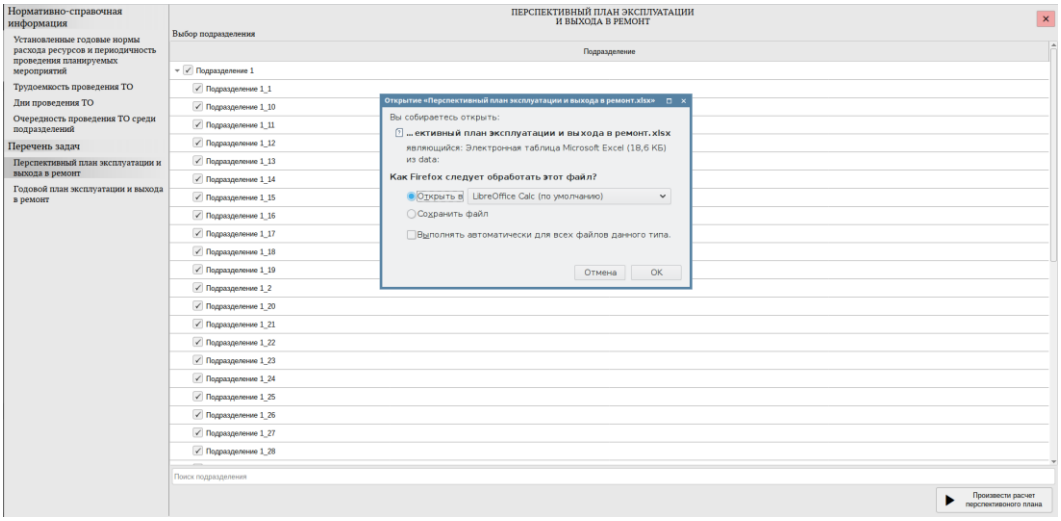


Рисунок 23 - Результат выполнения задачи

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	ПЕРСПЕКТИВНЫЙ ПЛАН																			
2	ЭКСПЛУАТАЦИИ И ВЫХОДА В РЕМОНТ																			
3	Подразделение 1 на период с 2023 по 2032 года																			
4																				
5	подразделение	Индекс	Заводской номер	Год выпуска (КР)	Единица измерения ресурса	Наработка с начала эксплуатации (после КР)	Запас ресурса		Ресурс отпускаемый на год	Планируемые работы по годам										примечание
6							до СР	до КР		2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	
7	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
8	ТЕХНИКА ОБЫЧНОЙ ГРУППЫ ЭКСПЛУАТАЦИИ																			
9	Подразделение 1_42	Урал	т72-2	2020	год	3	97	157		ТО-2/1			РТО		ТО-2х с ПКП	РТО			РТО	
10		шасси	т72-201		км		4800	7800	600	ТО-2х/1			ППО		Гус/Шпив	УБ			ППО	
11		двигатель	т72-202		мото/час		870	1450	145	ТО-2х с ПКП						АБ				
12										РТО						ПДО				
13										УБ										
14										АБ										
15										ПДО										
16										Гус/Шпив										
17																				
18	Подразделение 1_30	Трактор	70422-1	2000	год	23	-20	-18		ТО-1/1	ТО-1/1	ТО-1/1	ТО-1/1	ТО-1/1	ТО-1/1	ТО-1/1	ТО-1/1	ТО-1/1	ТО-1/1	
19		шасси	7041		км	13	487	987	200	ТО-1/1 ТО-2/1	ТО-1/2	ТО-1/1 ТО-2/1	ТО-1/2	ТО-1/1 ТО-2/1	ТО-1/2	ТО-1/1 ТО-2/1	ТО-1/2	ТО-1/1 ТО-2/1	ТО-1/2	
20		двигатель	7042		мото/час	500	-400	-300	30	ТО-2х/1	ТО-1х/1	ТО-1х/1	ТО-1х/1	ТО-2х/1	ТО-1х/1	ТО-1х/1	ТО-1х/1	ТО-2х/1	ТО-1х/1	
21										ТО-2х с ПКП	РТО	РТО	РТО	РТО	РТО	ТО-2х с ПКП	РТО	РТО	РТО	
22										РТО			АБ	КР		РТО	Гус/Шпив		АБ	
23										АБ			ППО			АБ	СР		ППО	
24										ПДО			СР			ПДО			КР	
25										Гус/Шпив										
26										СР										
27																				
28	Подразделение 1_30	Трактор	7042240	2010 (2022)	год	1	2	4		ТО-1/1	ТО-1/1	ТО-1/1	ТО-1/1	ТО-1/1	ТО-1/1	ТО-1/1	ТО-1/1	ТО-1/1	ТО-1/1	
29		шасси	7042240-1		км		500	1000	200	ТО-1/1 ТО-2/1	ТО-1/2	ТО-1/1 ТО-2/1	ТО-1/2	ТО-1/1 ТО-2/1	ТО-1/2	ТО-1/1 ТО-2/1	ТО-1/2	ТО-1/1 ТО-2/1	ТО-1/2	
30		двигатель	7042240-2		мото/час		100	200	30	ТО-2х/1	ТО-1х/1	ТО-1х/1	ТО-1х/1	ТО-2х/1	ТО-1х/1	ТО-1х/1	ТО-1х/1	ТО-2х/1	ТО-1х/1	
31										ТО-2х с ПКП	РТО	РТО	РТО	РТО	РТО	ТО-2х с ПКП	РТО	РТО	РТО	
32										РТО		СР	АБ		КР	РТО	Гус/Шпив	СР	АБ	
33										АБ			ППО			АБ			ППО	
34										ПДО						ПДО				
35										Гус/Шпив										
36																				
37	ие 1_31	Камаз	yyy333	2011	год	12	-10	-8		ТО-2/1	ТО-1/1	ТО-1/1	ТО-1/1	ТО-2/1	ТО-1/1	ТО-1/1	ТО-1/1	ТО-2/1	ТО-1/1	
38		шасси	yy		км		500	1000	200	ТО-2х/1	ТО-1х/1	ТО-1х/1	ТО-1х/1	ТО-2х/1	ТО-1х/1	ТО-1х/1	ТО-1х/1	ТО-2х/1	ТО-1х/1	
39		двигатель	yyy422		мото/час		100	200	30	ТО-2х с ПКП	РТО	РТО	РТО	РТО	ТО-2х с ПКП	РТО	РТО	РТО	РТО	
40										РТО		СР	ППО	Гус/Шпив	РТО	ПДО		Гус/Шпив	ППО	

Перспективный план | Пояснительная записка | Исходные данные

Рисунок 24 - Первый раздел перспективного плана

	A	B	C	D	E
1	**III. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**				
2	**к перспективному плану эксплуатации и выхода в ремонт на 2023 год**				
3					
4	Планируемые работы с аббревиатурой мероприятий	Индекс			
5		Зил	Трактор	Камаз	Урал
6	1	2	3	4	5
7	В эксплуатации 15 лет и более (15)	0	0	0	2
8	Капитальный ремонт (КР)	1	0	0	0
9	Средний ремонт (ремонт по техническому состоянию) СР (РТС)	1	0	0	2
10	Техническое обслуживание № 1 (ТО-1)	0	0	0	3
11	Техническое обслуживание № 2 (ТО-2)	2	1	1	3
12	Техническое обслуживание № 1х (ТО-1х)	0	0	0	0
13	Техническое обслуживание № 2х (ТО-2х)	2	1	1	3
14	Техническое обслуживание № 2х с ПКП (ТО-2х с ПКП)	2	1	1	3
15	Регламентированное техническое обслуживание (РТО)	2	1	1	3
16	Замена АБ (АБ)	2	1	1	3
17	Освидетельствование воздушных баллонов ВБ	2	1	1	3
18	Замена гусеничной ленты / шин (Гус / Шина)	2	1	1	3
19	Замена укрывочных брезентов (УБ)	2	1	1	0
20	Достижение неснижаемого ресурса (НР)	0	0	0	0
21	**III. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**				
22	**к перспективному плану эксплуатации и выхода в ремонт на 2024 год**				
23					
24	Планируемые работы с аббревиатурой мероприятий	Индекс			
25		Зил	Трактор	Камаз	Урал
26	1	2	3	4	5
27	В эксплуатации 15 лет и более (15)	0	0	0	2
28	Капитальный ремонт (КР)	0	0	0	0
29	Средний ремонт (ремонт по техническому состоянию) СР (РТС)	0	0	0	0
30	Техническое обслуживание № 1 (ТО-1)	2	0	0	3
31	Техническое обслуживание № 2 (ТО-2)	0	0	0	0
32	Техническое обслуживание № 1х (ТО-1х)	2	0	0	3

Перспективный план | Пояснительная записка | Исходные данные

Рисунок 25 - Второй раздел перспективного плана

	A	B	C	D	E	F
1	**IV. ИСХОДНЫЕ ДАННЫЕ ДЛЯ ПЛАНИРОВАНИЯ**					
2						
3	Аббревиатура мероприятий	Проводимые мероприятия	Периодичность проведения			
4			Наименование ВВТ			
5			**Зил**	**Трактор**	**Камаз**	**Урал**
6	1	2	3	4	5	6
7	КР 1	1-й Капитальный ремонт	1 раз в 160 лет	1 раз в 5 лет	1 раз в 4 года	1 раз в 160 лет
8	СР 1	1-й Средний ремонт	1 раз в 100 лет	1 раз в 3 года	1 раз в 2 года	1 раз в 100 лет
9	ТО-1	Проведение ТО-1	1 раз в 12 лет	1 раз в 0.5 лет	1 раз в 1 год	1 раз в 12 лет
10	ТО-2	Проведение ТО-2	1 раз в 36 лет	1 раз в 2 года	1 раз в 4 года	1 раз в 36 лет
11	ТО-1х	Проведение ТО-1х	1 раз в 60 лет	1 раз в 1 год	1 раз в 1 год	1 раз в 60 лет
12	ТО-2х	Проведение ТО-2х	1 раз в 84 лет	1 раз в 4 года	1 раз в 4 года	1 раз в 84 лет
13	ТО-2х с ПКП	Проведение ТО-2х с ПКП	1 раз в 5 лет	1 раз в 6 лет	1 раз в 5 лет	1 раз в 5 лет
14	РТО	Проведение РТО	1 раз в 2 года	1 раз в 1 год	1 раз в 1 год	1 раз в 3 года
15	УБ	Замена укрывочных брезентов	1 раз в 7 лет		1 раз в 5 лет	1 раз в 6 лет
16	АБ	Замена АБ	1 раз в 5 лет	1 раз в 3 года	1 раз в 5 лет	1 раз в 6 лет
17	ВБ (ППО, ОУ)	Освидетельствование воздушных баллонов (баллонов ППО, огнетушителей)	1 раз в 3 года	1 раз в 3 года	1 раз в 3 года	1 раз в 3 года
18	НР	Достижение неснижаемого ресурса	после 800 км		после 6000 моточас	
19	Гус Шина	Замена гусеничной ленты / шин	1 раз в 4 года	1 раз в 7 лет	1 раз в 4 года	1 раз в 5 лет

Рисунок 26 - Третий раздел перспективного плана

3.6 Годовой план эксплуатации и выхода в ремонт

Данный пункт позволяет задать расход ресурса по месяцам и выполнить формирование годового плана (рисунок 27).

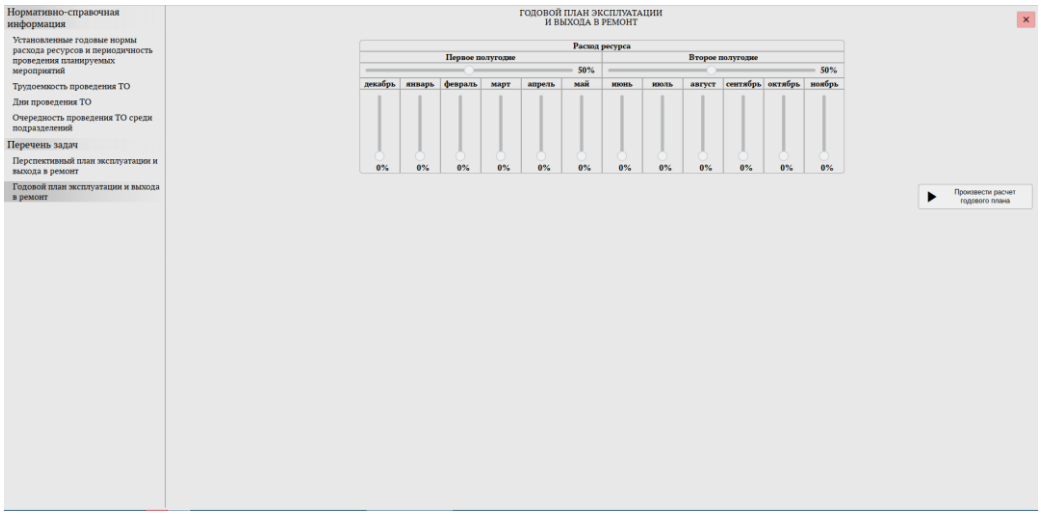


Рисунок 27 - Пункт меню «Годовой план эксплуатации и выхода в ремонт»

Пользователю необходимо учесть при распределении ресурса, что сначала выполняется разделение по полугодиям, а после по месяцам. Он не сможет выбрать расход ресурса более 100 процентов, так как это ограничено приложением. Если не задать расход ресурса, то в итоговом документе соответствующие поля будут иметь пустые значения. Также пользователь не сможет сформировать годовой план, если ранее не был сформирован перспективный план, об этом его предупредит информационное сообщение, показанное на рисунке 28.

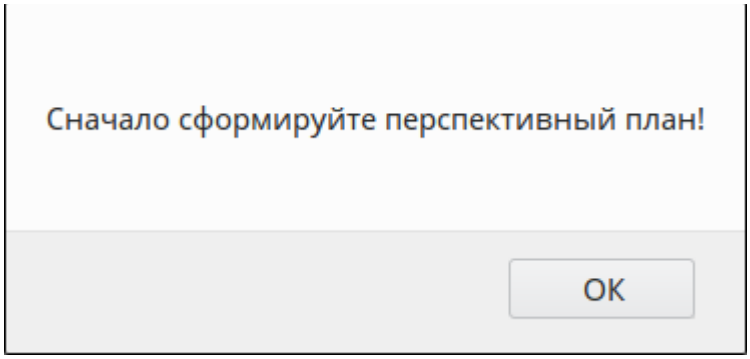


Рисунок 28 - Ошибка формирования годового плана

При запуске задачи перед пользователем появиться аналогичное окно ожидания, как и при формировании перспективного плана (рисунок 22).

Результатом работы будет файл формата xlsx (рисунок 29), структура которого показана на рисунках 30 и 31.

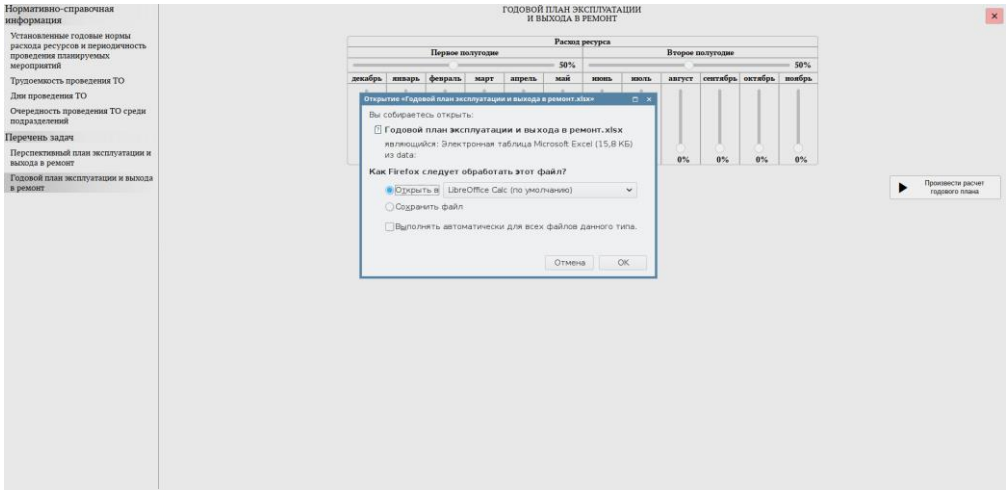


Рисунок 29 - Результат формирования плана

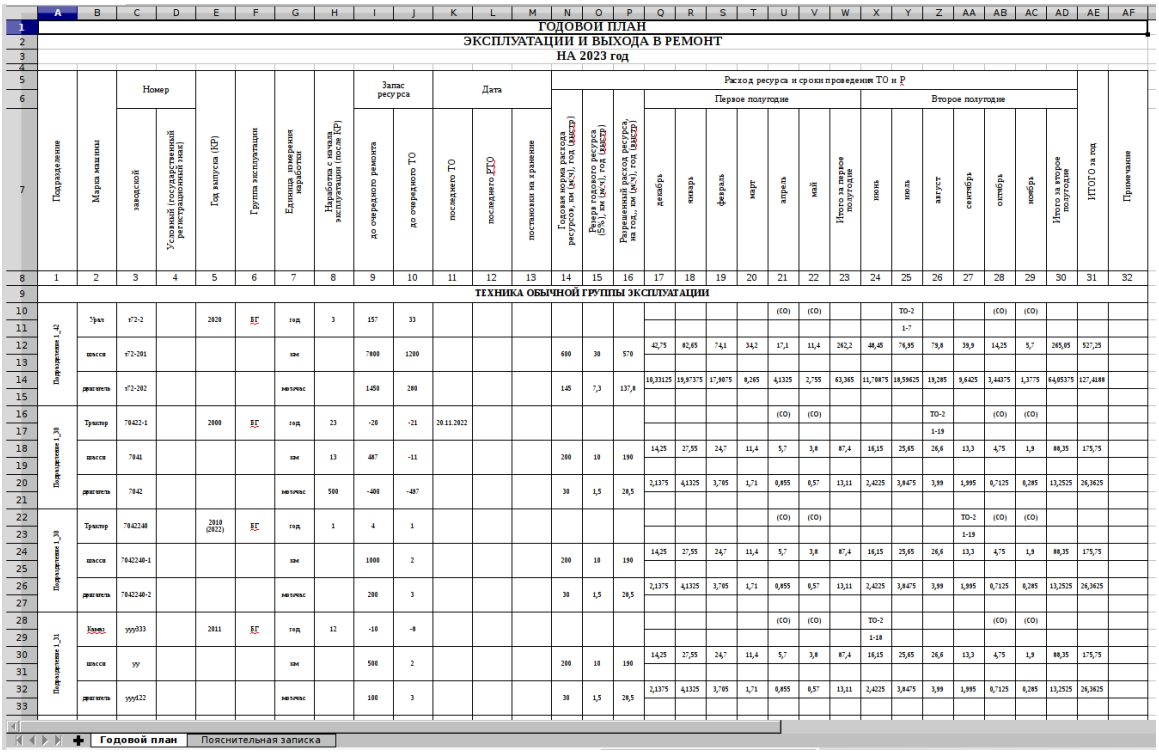


Рисунок 30 - Первый раздел годового плана

III. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к годовому плану эксплуатации и выхода в ремонт на 2023 год

Марка машины	Количество машин	Ресурс на год, км (м/ч)	Запланировано технических обслуживаний и ремонтов на 2023 год, ед.													Планируемые работы, в том числе, ед.															Примечание
																войсковыми ремонтно-восстановительными органами									специалистами промышленности								
			Всего ТО	ТО-1	ТО-2	СО	ТО-1х	ТО-2х	ТО-2х с ДКП	РДО	РДС	ТР	СР	КР	Всего ТО	ТО-1	ТО-2	СО	ТО-1х	ТО-2х	ТО-2х с ДКП	РДО	ТР	СР	Всего ТО	ТО-1	ТО-2	СО	ТО-1х	ТО-2х	ТО-2х с ДКП	РДО	РДС	ТР	СР	КР	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
ТЕХНИКА																																					
Камаз	2	330	10	0	2	1	0	2	2	2	0	0	1	0	10	0	2	1	0	2	2	2	0	1	0												
Зил	1	600	4	0	1	0	0	1	1	1	0	0	0	0	4	0	1	0	0	1	1	1	0	0	0												
Урал	1	600	5	0	1	1	0	1	1	1	0	0	0	0	5	0	1	1	0	1	1	1	0	0	0												
Трактор	3	460	17	1	3	2	0	3	3	3	0	0	2	0	17	1	3	2	0	3	3	3	0	2	0												

Рисунок 31 - Второй раздел годового плана

ЗАКЛЮЧЕНИЕ

В ходе выполнения магистерской диссертации были получены навыки формирования документов планирования на основе имеющихся данных. Также были изучены особенности построения веб-приложений с использованием современных фронтенд и бэкенд веб-фреймворков. Помимо этого, были получены навыки в применении средств контейнеризации для переноса приложений.

Результатом выполненной работы является веб-приложение, позволяющее формировать перспективный и годовой планы эксплуатации и выхода в ремонт техники. На основе образа разработанного программного обеспечения был создан контейнер, который позволяет переносить приложение на другие устройства.

Была проведена экспериментальная проверка работоспособности. Для этого приложение было запущено в контейнере с помощью ПО Docker на компьютере без установленного языка программирования python и базы данных PostgreSQL. Тестирование ПО позволило выявить проблемы и предпринять меры по их устранению.

В дальнейшем приложение может быть усовершенствовано путем добавления большего количества мероприятий по техническому обслуживанию и ремонту.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дакетт Джон Javascript и jQuery. Интерактивная веб-разработка: Издательство Э, 2020

2. Форсье Джефф, Биссекс Пол Django. Разработка веб-приложений на Python: Символ-Плюс, 2009

3. Joshua D. Drake, John C. Worsley Practical PostgreSQL: O'Reilly Media, Inc., 2002

4. Adrian Mouat Using Docker: O'Reilly Media, Inc., 2015

5. Научная библиотека избранных естественно-научных изданий – URL: https://scask.ru/.

6. Мэтиз Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения: Питер, 2021

7. Форум программистов и сисадминов Киберфорум – URL: https://www.cyberforum.ru/.

8. Система вопросов и ответов о программировании Stack Overflow – URL: https://stackoverflow.com/.

ПРИЛОЖЕНИЕ А

Листинг программы

Файл «views.py»

```
from django.shortcuts import render
from django.http import HttpResponse
from django.db import connection
from datetime import datetime
from plan_er.py_files.plan1_db import start_planing_one, planinig_sr_kr, planing_nr
from plan_er.py_files.plan2_db import create_year_db
from plan_er.py_files.print_plan1 import print_plan_one
from plan_er.py_files.print_plan2 import print_year_plan
import json

cursor = connection.cursor()

# Загрузка страницы
def index(request):
    return render(request, 'plan_er/plan_er.html')

# Загрузка НСИ
def load_nsi_grid(request):
    id_vch = request.GET.get('id_vch', None)
    al_d = []
    cur_d = []

    query1 = '''SELECT distinct on (kart.id_prsn) kart.id, kart.id_prsn FROM main.dok_kart as kart
    left join main.kls_discript as dis on (kart.id_prsn = dis.id_prsn)
    WHERE dis.id_discript =31 and kart.kod_dok=\'043\'
    and kart.id_vch_ch in (select v_id_vch from main.getlistvchpodch('' + str(
        id_vch) + '',0) where v_id_vch = '' + str(id_vch) + '' or l_id_parent = '' + str(
        id_vch) + '')
    and lower(fild_str) not like \'%боеприпас%\' and lower(fild_str) not like \'%ракет%\' and
lower(fild_str) not like \'%имущ%\' and kart.id_servis = 1 and kart.id_depend=2
    order by kart.id_prsn,id desc'''

    cursor.execute(query1)
    osnovn = cursor.fetchall()

    arr = []
    for os in osnovn:
        query1_1 = '''select kart.id,kart.id_prsn from main.dok_kart as kart, main.dok_kart_tex as tex
where kart.id=tex.id_kart and
        kart.id_prsn='' + str(os[1]) + '' and tex.id_chprsn=0  and tex.id_edizm=13 and kart.id_vch_ch
in(
        select v_id_vch from main.getlistvchpodch('' + str(
            id_vch) + '',0) where v_id_vch = '' + str(id_vch) + '' or l_id_parent = '' + str(
            id_vch) + ''
        ) order by kart.id limit 1'''
        cursor.execute(query1_1)
        os_res = cursor.fetchall()
```

```
        if len(os_res) != 0:
            arr.append(os_res[0])

    osnovn = arr

    query2 = "select id_rav,id_rav_grid,ed_izm from ravbook.nsi_ravbook where id_vch = " +
str(id_vch)

    cursor.execute(query2)
    curent_data_of_obrazec = cursor.fetchall()

    for cur in curent_data_of_obrazec:
        cur_d.append([cur[0], cur[1], int(cur[2])])

    for i in osnovn:
        id_kart = i[0]
        id_name = i[1]

        query3 = "select distinct id_chprsn from main.dok_kart2 where id_kart =" + str(id_kart)

        cursor.execute(query3)
        sostav = cursor.fetchall()

        fullpart_nomen = []
        if sostav != None:
            if len(sostav) != 0:
                fullpart_nomen.append(id_name)
        for j in sostav:
            fullpart_nomen.append(j[0])

        query4 = "select short_name from main.kls_prsn where id =" + str(id_name)

        cursor.execute(query4)
        id_name_text = cursor.fetchone()
        id_name_text = id_name_text[0]

        rav_part = -1
        for k in fullpart_nomen:
            id_rav_grid = str(id_name) + str(k)

            rav_part += 1
            # query = "select ed.abbrev as
edizm,nsi.god_resurs,nsi.rem1_cp_max,nsi.rem1_kp_max,nsi.rem2_cp_max,nsi.rem2_kp_max," \
            #         "nsi.rem3_max,nsi.rem_to1_max,nsi.rem_to2_max,nsi.rem_to1x_max,nsi.rem_to2x_
max " \
            #         "from main.nsi_resurs as nsi left join main.sl_edizm as ed on (ed.id in (8, 9, 10, 11,
15, 16, 17)) where nsi.id_prsnvoor =" + str(
            #     k)

            query5 = "select
id_edizm,nsi.god_resurs,nsi.rem1_cp_max,nsi.rem1_kp_max,nsi.rem2_cp_max,nsi.rem2_kp_max,
nsi.rem3_max," \
```

```
            "nsi.rem_to1_max,nsi.rem_to2_max,nsi.rem_to1x_max,nsi.rem_to2x_max from
main.nsi_resurs as nsi where nsi.id_prsnvoor =" + str(
            k)

        if rav_part == 0:
            query5 = query5 + " and nsi.id_edizm = 13"
        else:
            query5 += " and id_edizm in (8, 9, 10, 11, 15, 16, 17)"

        cursor.execute(query5)
        data = cursor.fetchall()
        if len(data) != 0:
            cur_ed_izm = data[0][0]

        for dl in data:
            al_d.append([int(id_name), int(id_rav_grid), dl[0]])
            query6 = "select id_rav from ravbook.nsi_ravbook where id_rav =" + str(
                id_name) + " and id_rav_grid = " + str(id_rav_grid) + " and ed_izm = '" + str(
                dl[0]) + "' and id_vch=" + str(id_vch)

            cursor.execute(query6)
            is_in = cursor.fetchall()

            new_dl = []
            for v in dl:
                if v == None:
                    v = "NULL"
                new_dl.append(v)

            query7 = "select short_name from main.kls_prsn where id =" + str(k)
            cursor.execute(query7)
            short_name = cursor.fetchone()
            short_name = short_name[0]

            if rav_part != 0:
                if new_dl[0] == 15:
                    short_name = 'орудие'
                elif new_dl[0] == 8:
                    short_name = 'шасси'
                elif new_dl[0] == 9:
                    short_name = 'двигатель'
                else:
                    short_name = ''

            if rav_part == 0:
                new_dl[1] = 'NULL'

            if rav_part == 1:
                prior = "true"
            else:
                prior = "false"
```

```
            if len(is_in) == 0:
                query8 = "insert into
ravbook.nsi_ravbook(id_rav,id_rav_grid,id_part_rav,group_index_name,rav_index_name,ed_izm,c
ur_ed_izm," \
                         "year_resource_norm_bg, year_resource_norm_pr, sr_remont_1, kap_remont_1,
sr_remont_2, kap_remont_2, spisanie, " \
                         "to_1, to_2, to_1x,to_2x, id_vch, selected_prior) (select " + str(
                    id_name) + "," + str(id_rav_grid) + "," + str(rav_part) \
                         + ",'" + str(id_name_text) + "','" + str(short_name) + "','" + str(
                    new_dl[0]) + "','" + str(
                    cur_ed_izm) + "'," + str(
                    new_dl[1]) + "," + str(new_dl[1]) + "," + str(new_dl[2]) + "," + str(
                    new_dl[3]) + "," \
                         + str(new_dl[4]) + "," + str(new_dl[5]) \
                         + "," + str(new_dl[6]) + "," + str(new_dl[7]) + "," + str(new_dl[8]) + "," + str(
                    new_dl[9]) + "," \
                         + str(new_dl[10]) + "," + str(
                    id_vch) + "," + prior + " from main.kls_prsn where id =" + str(k) + ")"

                cursor.execute(query8)
            else:
                query8 = "update ravbook.nsi_ravbook set rav_index_name='" + str(
                    short_name) + "', year_resource_norm_bg=" + str(
                    new_dl[1]) + ", sr_remont_1=" + str(new_dl[
                                                        2]) + ", " \
                                                              "kap_remont_1=" + str(
                    new_dl[3]) + ", sr_remont_2=" + str(new_dl[4]) + ", kap_remont_2=" + str(new_dl[
                                                                                                  5]) + ", spisanie=" + str(
                    new_dl[
                        6]) + ", to_1=" + str(new_dl[7]) + ", to_2=" + str(new_dl[8]) + ", to_1x=" + str(
                    new_dl[9]) + ",to_2x=" + str(new_dl[
                                                     10]) + " where id_rav_grid =" + str(
                    id_rav_grid) + " and ed_izm ='" + str(new_dl[0]) + "' and id_vch= " + str(id_vch)

                cursor.execute(query8)

                query9 = "update ravbook.nsi_ravbook as rav set cur_ed_izm = " \
                         "(select rav3.ed_izm from ravbook.nsi_ravbook as rav3 where rav3.id_rav_grid =
rav.id_rav_grid limit 1) " \
                         "where id_rav_grid in (select distinct rav.id_rav_grid from ravbook.nsi_ravbook
as rav where rav.id_vch = " + id_vch + " " \
                                                                                                                                    "and
rav.cur_ed_izm not in (select rav2.ed_izm from ravbook.nsi_ravbook as rav2 where " \
                                                                                                                                    "rav2.i
d_rav_grid = rav.id_rav_grid))"

                cursor.execute(query9)

    not_in = []
    if len(cur_d) != 0:
        for b in cur_d:
            if b not in al_d:
```

```
            not_in.append(b)

    for b in not_in:
        query10 = "delete from ravbook.nsi_ravbook where id_rav =" + str(b[0]) + " and id_rav_grid =
" + str(
            b[1]) + " and id_vch = " + str(
            id_vch) + "and ed_izm ='" + str(b[2]) + "'"

        cursor.execute(query10)

    query11 = "select * from ravbook.grid_view where id_vch=" + str(id_vch)

    cursor.execute(query11)
    result = cursor.fetchall()

    jsonstr = ""
    for i in result:
        if i[25] == False:
            state = "false"
        else:
            state = "true"
        jsonstr += ',{"id":"' + str(i[0]) + '",' + \
                   '"id_part_rav": "' + str(i[1]) + '",' + \
                   '"index_rav": "' + str(i[2]) + '",' + \
                   '"ed_izm": "' + str(i[3]) + '",' + \
                   '"year_resource_norm_bg": "' + str(i[4]) + '",' + \
                   '"year_resource_norm_pr": "' + str(i[5]) + '",' + \
                   '"sr_remont_1": "' + str(i[6]) + '",' + \
                   '"kap_remont_1": "' + str(i[7]) + '",' + \
                   '"sr_remont_2": "' + str(i[8]) + '",' + \
                   '"kap_remont_2": "' + str(i[9]) + '",' + \
                   '"spisanie": "' + str(i[10]) + '",' + \
                   '"to_1": "' + str(i[11]) + '",' + \
                   '"to_2": "' + str(i[12]) + '",' + \
                   '"to_1x": "' + str(i[13]) + '",' + \
                   '"to_2x": "' + str(i[14]) + '",' + \
                   '"to_2x_pkp": "' + str(i[15]) + '",' + \
                   '"rto": "' + str(i[16]) + '",' + \
                   '"gsm": "' + str(i[17]) + '",' + \
                   '"ykr_brez": "' + str(i[18]) + '",' + \
                   '"ab": "' + str(i[19]) + '",' + \
                   '"tank_shlem": "' + str(i[20]) + '",' + \
                   '"balon": "' + str(i[21]) + '",' + \
                   '"gys_shin": "' + str(i[22]) + '",' + \
                   '"nr": "' + str(i[23]) + '",' + \
                   '"selected_prior": "' + str(state) + '"}'

    jsonstr = '[' + jsonstr[1:len(jsonstr)] + ']'

    query12 = "select id_rav_grid,ed_izm from ravbook.nsi_ravbook where id_vch =" + str(id_vch)
    query12 = "select nsi.id_rav_grid,ed.abbrev from ravbook.nsi_ravbook as nsi " \
```

```
            "left join main.sl_edizm as ed on (ed.id=nsi.ed_izm::numeric) where id_vch =" +
str(id_vch)

    cursor.execute(query12)
    result = cursor.fetchall()

    asd = """
    for i in result:
        asd += '",{"id":"' + str(i[0]) + '","' + \
               'ed_izm": "' + str(i[1]) + '"}'
    asd = '[' + asd[1:len(asd)] + ']'

    jsonstr += "," + asd

    return HttpResponse({jsonstr})

# Обновление НСИ
def update_nsi_grid(request):
    id_vch = request.GET.get('id_vch', None)
    id = request.GET.get('id', None)
    ed = request.GET.get('ed_izm', None)
    query14 = "select id from main.sl_edizm where abbrev='" + str(ed) + "'"
    cursor.execute(query14)
    result = cursor.fetchone()
    ed = result[0]

    datafield = request.GET.get('datafield', None)
    value = request.GET.get('value', None)
    if value == '':
        value = 'Null'

    if datafield == 'cur_ed_izm':
        query15 = "select id from main.sl_edizm where abbrev='" + str(value) + "'"
        cursor.execute(query15)
        result = cursor.fetchone()
        value = result[0]

        query16 = "update ravbook.nsi_ravbook set " + str(datafield) + " = '" + str(
            value) + "' where id_rav_grid = " + str(
            id) + " and id_vch = " + str(id_vch)

        cursor.execute(query16)

        query17 = "select * from ravbook.grid_view "

        cursor.execute(query17)
        result = cursor.fetchall()

        jsonstr = """
        for i in result:
            if i[25] == False:
                state = "false"
```

```
            else:
                state = "true"
            jsonstr += ',{"id":"' + str(i[0]) + '",' + \
                       '"id_part_rav": "' + str(i[1]) + '",' + \
                       '"index_rav": "' + str(i[2]) + '",' + \
                       '"ed_izm": "' + str(i[3]) + '",' + \
                       '"year_resource_norm_bg": "' + str(i[4]) + '",' + \
                       '"year_resource_norm_pr": "' + str(i[5]) + '",' + \
                       '"sr_remont_1": "' + str(i[6]) + '",' + \
                       '"kap_remont_1": "' + str(i[7]) + '",' + \
                       '"sr_remont_2": "' + str(i[8]) + '",' + \
                       '"kap_remont_2": "' + str(i[9]) + '",' + \
                       '"spisanie": "' + str(i[10]) + '",' + \
                       '"to_1": "' + str(i[11]) + '",' + \
                       '"to_2": "' + str(i[12]) + '",' + \
                       '"to_1x": "' + str(i[13]) + '",' + \
                       '"to_2x": "' + str(i[14]) + '",' + \
                       '"to_2x_pkp": "' + str(i[15]) + '",' + \
                       '"rto": "' + str(i[16]) + '",' + \
                       '"gsm": "' + str(i[17]) + '",' + \
                       '"ykr_brez": "' + str(i[18]) + '",' + \
                       '"ab": "' + str(i[19]) + '",' + \
                       '"tank_shlem": "' + str(i[20]) + '",' + \
                       '"balon": "' + str(i[21]) + '",' + \
                       '"gys_shin": "' + str(i[22]) + '",' + \
                       '"nr": "' + str(i[23]) + '",' + \
                       '"selected_prior": "' + str(state) + '"}'

        jsonstr = '[' + jsonstr[1:len(jsonstr)] + ']'

        return HttpResponse({jsonstr})

    elif ed == None:
        query18 = "update ravbook.nsi_ravbook set " + str(datafield) + " = " + str(
            value) + " where id_rav_grid = " + str(
            id) + " and id_vch = " + str(id_vch)

        cursor.execute(query18)

    else:
        query19 = "update ravbook.nsi_ravbook set " + str(datafield) + " = " + str(
            value) + " where id_rav_grid = " + str(
            id) + " and ed_izm = '" + str(ed) + "' and id_vch = " + str(id_vch)

        cursor.execute(query19)

    return HttpResponse({1})

# Загрузка трудоемкости
def load_trudoemkost_to(request):
    id_vch = request.GET.get('id_vch', None)
    query13 = "select * from ravbook.trudoemkost_to where id_vch=" + str(id_vch)
```

```
    query13 = ''' SELECT id_rav_grid, group_index_name, rav_index_name, to1_tr , to1_sp , to2_tr ,
to2_sp , to1x_tr ,
    to1x_sp , to2x_tr , to2x_sp, to2x_pkp_tr , to2x_pkp_sp , rto_tr , rto_sp , id_vch
    FROM ravbook.nsi_ravbook where id_vch = ''' + str(id_vch) + ''' and id_part_rav=0'''

    cursor.execute(query13)
    result = cursor.fetchall()

    jsonstr = """
    for i in result:
        jsonstr += ''',{"id":""" + str(i[0]) + """,''' + \
                """group_index": """ + str(i[1]) + """,''' + \
                """index_rav": """ + str(i[2]) + """,''' + \
                """to1_tr": """ + str(i[3]) + """,''' + \
                """to1_sp": """ + str(i[4]) + """,''' + \
                """to2_tr": """ + str(i[5]) + """,''' + \
                """to2_sp": """ + str(i[6]) + """,''' + \
                """to1x_tr": """ + str(i[7]) + """,''' + \
                """to1x_sp": """ + str(i[8]) + """,''' + \
                """to2x_tr": """ + str(i[9]) + """,''' + \
                """to2x_sp": """ + str(i[10]) + """,''' + \
                """to2x_pkp_tr": """ + str(i[11]) + """,''' + \
                """to2x_pkp_sp": """ + str(i[12]) + """,''' + \
                """rto_tr": """ + str(i[13]) + """,''' + \
                """rto_sp": """ + str(i[14]) + """}'''
    jsonstr = '''[''' + jsonstr[1:len(jsonstr)] + ''']'''

    return HttpResponse({jsonstr})

def update_trudoemkost_to(request):
    id = request.POST.get('id', None)
    datafield = request.POST.get('datafield', None)
    value = request.POST.get('value', None)
    if value == '':
        value = 'Null'
    id_vch = request.POST.get('id_vch', None)
    query_update_trudoemkost = '''update ravbook.nsi_ravbook set ''' + str(datafield) + '''=''' + str(
        value) + ''' where id_vch = ''' + str(id_vch) + ''' and id_rav_grid=''' + str(id)
    cursor.execute(query_update_trudoemkost)
    return HttpResponse({0})

# Загрузка дат ТО
def load_date_to(request):
    id_vch = request.GET.get('id_vch', None)
    query = '''select id_vch from ravbook.date_to where id_vch=''' + str(id_vch)
    cursor.execute(query)
    res = cursor.fetchall()
    if not res:
        query = '''insert into ravbook.date_to
(jan_main,jan_rezerv,feb_main,feb_rezerv,mar_main,mar_rezerv,
        apr_main,apr_rezerv,may_main,may_rezerv,jun_main,jun_rezerv,jul_main,jul_rezerv,aug_mai
n,aug_rezerv,
```

```
        sep_main,sep_rezerv,oct_main,oct_rezerv,nov_main,nov_rezerv,dec_main,dec_rezerv,id_vch)
        values('{1,7}','{17,21}','{1,7}','{17,21}','{1,7}','{17,21}','{1,7}','{17,21}','{1,7}','{17,21}','{1,7
}','{17,21}',
        '{1,7}','{17,21}','{1,7}','{17,21}','{1,7}','{17,21}','{1,7}','{17,21}','{1,7}','{17,21}','{1,7}','{17,
21}','" + str(
            id_vch) + "')"
        cursor.execute(query)

    query = "select * from ravbook.date_to where id_vch='" + str(id_vch)
    cursor.execute(query)
    res = cursor.fetchall()
    res = res[0]
    a = res[1][0]

    text = "{"jan_main_from":'" + str(res[1][0]) + "',"jan_main_to":'" + str(
        res[1][1]) + "',"jan_rez_from":'" + str(res[2][0]) + "',"jan_rez_to":'" + str(res[2][1]) + "',
    "feb_main_from":'" + str(res[3][0]) + "',"feb_main_to":'" + str(res[3][1]) + "',"feb_rez_from":'" +
str(
        res[4][0]) + "',"feb_rez_to":'" + str(res[4][1]) + "',
    "mar_main_from":'" + str(res[5][0]) + "',"mar_main_to":'" + str(res[5][1]) + "',"mar_rez_from":'"
+ str(
        res[6][0]) + "',"mar_rez_to":'" + str(res[6][1]) + "',
    "apr_main_from":'" + str(res[7][0]) + "',"apr_main_to":'" + str(res[7][1]) + "',"apr_rez_from":'" +
str(
        res[8][0]) + "',"apr_rez_to":'" + str(res[8][1]) + "',
    "may_main_from":'" + str(res[9][0]) + "',"may_main_to":'" + str(res[9][1]) + "',"may_rez_from":'"
+ str(
        res[10][0]) + "',"may_rez_to":'" + str(res[10][1]) + "',
    "jun_main_from":'" + str(res[11][0]) + "',"jun_main_to":'" + str(res[11][1]) + "',"jun_rez_from":'"
+ str(
        res[12][0]) + "',"jun_rez_to":'" + str(res[12][1]) + "',
    "jul_main_from":'" + str(res[13][0]) + "',"jul_main_to":'" + str(res[13][1]) + "',"jul_rez_from":'" +
str(
        res[14][0]) + "',"jul_rez_to":'" + str(res[14][1]) + "',
    "aug_main_from":'" + str(res[15][0]) + "',"aug_main_to":'" + str(res[15][1]) +
"',"aug_rez_from":'" + str(
        res[16][0]) + "',"aug_rez_to":'" + str(res[16][1]) + "',
    "sep_main_from":'" + str(res[17][0]) + "',"sep_main_to":'" + str(res[17][1]) + "',"sep_rez_from":'"
+ str(
        res[18][0]) + "',"sep_rez_to":'" + str(res[18][1]) + "',
    "oct_main_from":'" + str(res[19][0]) + "',"oct_main_to":'" + str(res[19][1]) + "',"oct_rez_from":'"
+ str(
        res[20][0]) + "',"oct_rez_to":'" + str(res[20][1]) + "',
    "nov_main_from":'" + str(res[21][0]) + "',"nov_main_to":'" + str(res[21][1]) +
"',"nov_rez_from":'" + str(
        res[22][0]) + "',"nov_rez_to":'" + str(res[22][1]) + "',
    "dec_main_from":'" + str(res[23][0]) + "',"dec_main_to":'" + str(res[23][1]) + "',"dec_rez_from":'"
+ str(
        res[24][0]) + "',"dec_rez_to":'" + str(res[24][1]) + "'}"

    return HttpResponse({text})
```

```
# Обновлениие дат ТО
def update_date_to(request):
    id_vch = request.GET.get('id_vch', None)
    value = request.GET.get('value', None)
    datafield = request.GET.get('datafield', None)

    update_date = "update ravbook.date_to set " + str(datafield) + " = '{" + str(
        value) + "}' where id_vch = " + str(id_vch)

    cursor.execute(update_date)

    return HttpResponse({0})

def update_priority(id_vch):
    query_prior = "'select id_vch from ravbook.queue_to where main_vch = '" + str(id_vch) + "' order
by priority"'
    cursor.execute(query_prior)
    res = cursor.fetchall()
    if res != None:
        priority = 1
        for i in res:
            query_prior2 = "'update ravbook.queue_to set priority = '" + str(priority) + "' where
main_vch='" + str(
                id_vch) + "' and id_vch = '" + str(i[0])
            cursor.execute(query_prior2)
            priority += 1

def load_queue_to(request):
    id_vch = request.GET.get('id_vch', None)
    load_queue_1 = "'select id_vch from ravbook.queue_to where main_vch = '" + str(id_vch)
    cursor.execute(load_queue_1)
    in_data = cursor.fetchall()
    res = []
    for i in in_data:
        res.append(i[0])
    in_data = res

    load_queue_2 = "'select v_id_vch from main.getlistvchpodch('" + str(
        id_vch) + "',0) where v_id_vch = '" + str(id_vch) + "' or l_id_parent = '" + str(
        id_vch)
    cursor.execute(load_queue_2)
    result2 = cursor.fetchall()
    new_data = []
    for i in result2:
        new_data.append(i[0])

    if len(new_data) > 1:
        new_data.remove(int(id_vch))

    if len(in_data) == 0:
        prior = 1
        for i in new_data:
```

```
            load_queue_3 = "'insert into ravbook.queue_to(id_vch, priority, main_vch) values('" + str(
                i) + "','" + str(prior) + "','" + str(id_vch) + "')'"
            prior += 1
            cursor.execute(load_queue_3)

    else:
        not_in = []
        for i in in_data:
            if i not in new_data:
                not_in.append(i)

        for i in not_in:
            load_queue_4 = "'delete from ravbook.queue_to where id_vch = '" + str(i) + "' and
main_vch='" + str(
                id_vch)
            cursor.execute(load_queue_4)

        to_add = []
        for i in new_data:
            if i not in in_data:
                to_add.append(i)

        load_queue_5 = "'select max(priority) from ravbook.queue_to where main_vch='" +
str(id_vch)
        cursor.execute(load_queue_5)
        priority = (cursor.fetchall())[0][0]

        for i in to_add:
            priority += 1
            load_queue_6 = "'insert into ravbook.queue_to(id_vch, priority, main_vch) values('" + str(
                i) + "','" + str(priority) + "','" + str(id_vch) + "')'"
            cursor.execute(load_queue_6)

        update_priority(id_vch)

    load_queue_7 = "'select qto.id_vch, rsv.namevch from ravbook.queue_to as qto,
main.rsv_vchsvoi as rsv where qto.main_vch='" + str(
        id_vch) + "' and rsv.id_vch = qto.id_vch order by qto.priority'"
    cursor.execute(load_queue_7)
    fin_res = cursor.fetchall()
    fin_str = """
    for i in fin_res:
        fin_str += "',""" + str(i[0]) + "'s":""" + str(i[1]) + """"
    fin_str = "'{'" + fin_str[1:len(fin_str)] + "'}'"

    return HttpResponse({fin_str})

def update_queue_to(request):
    vch = request.POST.get('vch', None)
    prior = request.POST.get('prior', None)
    id_vch = request.POST.get('id_vch', None)
    vch = json.loads(vch)
```

```
    prior = json.loads(prior)
    el = 0
    for i in vch:
        vch[el] = i[0:len(i) - 1]
        el += 1

    for i in range(len(vch)):
        query_queue_update1 = '''update ravbook.queue_to set priority = ''' + str(
            prior[i]) + ''' where main_vch='''' + str(id_vch) + ''' and id_vch = ''' + str(vch[i])
        cursor.execute(query_queue_update1)
    print(123456)
    return HttpResponse({0})

# Загрузка дерева подразделений
def build_podrazdel_tree(request):
    id_vch = request.GET.get('id_vch', None)
    podrazdel = '''select v_id_vch,l_id_parent,nam from main.getlistvchpodch(''' + str(
        id_vch) + ''',0) where v_id_vch = ''' + str(id_vch) + ''' or l_id_parent = ''' + str(id_vch)
    cursor.execute(podrazdel)
    result = cursor.fetchall()

    jsonstr = """

    for i in result:
        jsonstr += ''',{"id":""" + str(i[0]) + """,''' + \
                   """parent_id": """ + str(i[1]) + """,''' + \
                   """name": """ + str(i[2]) + """}'''

    jsonstr = '''[''' + jsonstr[1:len(jsonstr)] + ''']'''

    return HttpResponse({jsonstr})

# Перспективный план
def build_plan_db(request):
    id_vch = request.GET.get('id_vch_podrazdel', None)
    id_vch_main = request.GET.get('id_vch', None)

    year = datetime.now().year + 1
    if id_vch_main == id_vch:
        sub_st = '''select v_id_vch from main.getlistvchpodch(''' + str(
            id_vch_main) + ''',0) where l_id_parent = ''' + str(id_vch_main)
        cursor.execute(sub_st)
        id_vch = ""
        ids = cursor.fetchall()
        if ids != None and len(ids) != 0:
            for i in ids:
                id_vch += str(i[0]) + ","
            id_vch = id_vch[0:len(id_vch) - 1]
        else:
            id_vch = id_vch_main

    query20 = "delete from ravbook.perspective_plan where id_vch in (" + id_vch + ")"
```

```
    cursor.execute(query20)
    # id_vch_podrazdelenie = id_vch

    for type in range(2):
        if type == 0:
            query21 = '"SELECT kart.id, kart.id_prsn,kart.num_part, kart.god_isg as god_isg_obraze,
kart2.id_chprsn,kart2.num_part,id_vch, kartpodp.id_grupp,kartpodp.num_str
            FROM main.dok_kart as kart
            left join main.kls_discript as dis on (kart.id_prsn = dis.id_prsn)
            left join main.dok_kart2 as kart2 on (kart.id = kart2.id_kart)
            left join main.dok_kartpodp as kartpodp on (kartpodp.id_kart= kart.id)
            WHERE dis.id_discript =31 and kart.kod_dok='043'
            and kart.id_vch_ch in ('" + id_vch + "')
            and lower(fild_str) not like '%боеприпас%' and lower(fild_str) not like '%ракет%' and
lower(fild_str) not like '%имущ%'
            and kart.id_prsn in (select distinct id_rav from ravbook.nsi_ravbook)  and kart2.num_formul
!= '' and kart.id_servis = 1 and kart.id_depend=2
            and kartpodp.num_str!='' and kartpodp.id_grupp = 1 order by id_vch, kart.id_prsn,kart.id
,kart2.id_chprsn"'

        if type == 1:
            query21 = '"SELECT kart.id, kart.id_prsn,kart.num_part, kart.god_isg as god_isg_obraze,
kart2.id_chprsn,kart2.num_part,id_vch, kartpodp.id_grupp,kartpodp.num_str
            FROM main.dok_kart as kart
            left join main.kls_discript as dis on (kart.id_prsn = dis.id_prsn)
            left join main.dok_kart2 as kart2 on (kart.id = kart2.id_kart)
            left join main.dok_kartpodp as kartpodp on (kartpodp.id_kart= kart.id)
            WHERE dis.id_discript =31 and kart.kod_dok='043'
            and kart.id_vch_ch in ('" + id_vch + "')
            and lower(fild_str) not like '%боеприпас%' and lower(fild_str) not like '%ракет%' and
lower(fild_str) not like '%имущ%'
            and kart.id_prsn in (select distinct id_rav from ravbook.nsi_ravbook)  and kart2.num_formul
!= '' and kart.id_servis = 1 and kart.id_depend=2
            and kartpodp.num_str!='' and kartpodp.id_grupp = 8 order by id_vch, kart.id_prsn,kart.id
,kart2.id_chprsn"'
        if type == 2:
            # query21 = '"SELECT kart.id, kart.id_prsn,kart.num_part, kart.god_isg as god_isg_obraze,
kart2.id_chprsn,kart2.num_part,id_vch FROM main.dok_kart as kart
            # left join main.kls_discript as dis on (kart.id_prsn = dis.id_prsn)
            # left join main.dok_kart2 as kart2 on (kart.id = kart2.id_kart)
            # WHERE dis.id_discript =31 and kart.kod_dok='043'
            # and kart.id_vch_ch in ('" + id_vch + "')
            # and lower(fild_str) not like \'%боеприпас%\' and lower(fild_str) not like \'%ракет%\' and
lower(fild_str) not like \'%имущ%\'
            # and kart.id_prsn in (select distinct id_rav from ravbook.nsi_ravbook)  and
kart2.num_formul != '' and kart.id_servis = 1
            # order by id_vch, kart.id_prsn,kart.id ,kart2.id_chprsn"'

            query21 = '"SELECT distinct on (kart.id_prsn) kart.id, kart.id_prsn FROM main.dok_kart as
kart
            left join main.kls_discript as dis on (kart.id_prsn = dis.id_prsn)
            WHERE dis.id_discript =31 and kart.kod_dok='043' and
```

```
            kart.id_vch_ch in (select v_id_vch from main.getlistvchpodch('" + str(
                id_vch) + "',0) where v_id_vch = '" + str(id_vch) + "' or l_id_parent = '" + str(
                id_vch) + "')
            and kart.id_servis=2 order by kart.id_prsn,id desc'"

        cursor.execute(query21)
        result = cursor.fetchall()

        old_kart_id = -1
        for res in result:
            id_vch_podrazdelenie = res[6]
            if res[0] != old_kart_id:
                old_kart_id = res[0]
                new_element = []
                new_element.append(res[0])
                new_element.append(id_vch_podrazdelenie)
                new_element.append(res[1])
                new_element.append(int(str(res[1]) + str(res[1])))
                new_element.append(res[2])
                new_element.append(res[3])

                query22 = "select date_rem from main.dok_kart_tex where id_kart = " + str(
                    res[0]) + " and id_prsn = " + str(
                    res[1]) + " and id_chprsn = 0 and (id_rem = 103 or id_rem = 105)"

                cursor.execute(query22)
                kr = cursor.fetchone()

                narabotka = 0

                if kr != None:
                    kr = kr[0].year
                    narabotka = year - kr
                    new_element.append(kr)
                else:
                    narabotka = year - res[3]
                    new_element.append(0)

                query23 = "select cur_ed_izm from ravbook.nsi_ravbook where id_rav_grid =" +
str(new_element[3])

                cursor.execute(query23)
                ed_izm = cursor.fetchone()
                new_element.append(ed_izm[0])

                new_element.append(narabotka)
                query24 = "select sr_remont_1,
kap_remont_1,year_resource_norm_bg,year_resource_norm_pr from ravbook.nsi_ravbook where
id_vch =" + str(
                    id_vch_main) + " and id_rav_grid = " + str(new_element[3])

                cursor.execute(query24)
```

```
            resurs = cursor.fetchall()
            if len(resurs) != 0:
                new_element.append(int(resurs[0][0] - narabotka))
                new_element.append(int(resurs[0][1] - narabotka))
            else:
                new_element.append('Null')
                new_element.append('Null')

            query25 = "insert into
ravbook.perspective_plan(id_kart,id_vch_podrasdel,rav_index_name,id_grid,zavod_num,god_izg,k
r," \
                      "ed_izm,narabotka,zapas_sr,zapas_kr,resurs_na_god,group_cat,id_vch) values(" +
str(
                new_element[0]) + "," + str(new_element[1]) + "," + str(new_element[2]) + "," + str(
                new_element[3]) + ",'" + str(new_element[4]) + "'," + str(new_element[5]) + "," + str(
                new_element[6]) + ",'" + str(new_element[7]) + "'," + str(new_element[8]) + "," + str(
                new_element[9]) + "," + str(new_element[10]) + ",NULL," + str(type) + "," +
str(res[6]) + ")"
            cursor.execute(query25)

            if type != 2:
                start_planing_one(int(str(res[1]) + str(res[1])), id_vch_main, res[0], cursor, year, type,
                                  str(res[6]))

        new_element = []

        new_element.append(res[0])
        new_element.append(id_vch_podrazdelenie)
        new_element.append(res[4])
        new_element.append(int(str(res[1]) + str(res[4])))
        new_element.append(res[5])
        new_element.append('Null')
        new_element.append('Null')

        query26 = "select cur_ed_izm from ravbook.nsi_ravbook where id_rav_grid =" +
str(new_element[3])

        cursor.execute(query26)
        ed_izm = cursor.fetchone()
        new_element.append(ed_izm[0])

        query27 = " select narab_ch, narab_rem_ch from main.dok_kart_tex where id_kart = " + str(
            res[0]) + " and id_chprsn = " + str(
            res[4]) + " and id_edizm =" + str(ed_izm[0])
        cursor.execute(query27)
        narabotka = cursor.fetchone()

        # if len(narabotka)!=0:
        if narabotka != None:
            if narabotka[1] == None:
                if narabotka[0] == None:
                    narabotka = 0
```

```
                    new_element.append(narabotka)
                else:
                    narabotka = narabotka[0]
                    new_element.append(narabotka)
            else:
                narabotka = narabotka[1]
                new_element.append(narabotka)
        else:
            narabotka = 0
            new_element.append(narabotka)

        query28 = "select sr_remont_1,
kap_remont_1,year_resource_norm_bg,year_resource_norm_pr from ravbook.nsi_ravbook where
id_vch =" + str(
            id_vch_main) + " and ed_izm = cur_ed_izm and id_rav_grid = " + str(new_element[3])

        cursor.execute(query28)
        resurs = cursor.fetchall()
        narabotka = float(narabotka)
        sr_rem = 0
        if resurs[0][0] != None:
            sr_rem = int(resurs[0][0])
        kr_rem = 0
        if resurs[0][1] != None:
            kr_rem = int(resurs[0][1])
        new_element.append(sr_rem - narabotka)
        new_element.append(kr_rem - narabotka)

        if type != 1:
            year_resurs = resurs[0][2]
        else:
            year_resurs = resurs[0][3]

        query29 = "insert into
ravbook.perspective_plan(id_kart,id_vch_podrasdel,rav_index_name,id_grid,zavod_num,god_izg,k
r," \
                  "ed_izm,narabotka,zapas_sr,zapas_kr,resurs_na_god,group_cat,id_vch) values(" + str(
            new_element[0]) + "," + str(new_element[1]) + "," + str(new_element[2]) + "," + str(
            new_element[3]) + "'" + str(new_element[4]) + "'," + str(new_element[5]) + "," + str(
            new_element[6]) + "'" + str(new_element[7]) + "'," + str(new_element[8]) + "," + str(
            new_element[9]) + "," + str(new_element[10]) + "," + str(year_resurs) + "," + str(type) +
"," + str(
            res[6]) + ")"
        cursor.execute(query29)

    planing_nr(cursor, id_vch, year, id_vch_main)
    planinig_sr_kr(cursor, id_vch, id_vch_main)
    result_doc = print_plan_one(cursor, id_vch, year, id_vch_main)

    import base64
    file_64 = base64.b64encode(result_doc)
```

```
    return HttpResponse(file_64)
    return HttpResponse({0})

def build_year_plan(request):
    a = create_year_db(request, cursor)
    if a == 0:
        result_doc = print_year_plan(request, cursor)
        import base64
        file_64 = base64.b64encode(result_doc)
        return HttpResponse(file_64)
    elif a == -1:
        return HttpResponse({-1})
    elif a == -2:
        return HttpResponse({-2})
```

Файл «plan1_db.py»

```
import math

def start_planing_one(grid_id, id_vch, kart, cursor, curent_year, type, cur_vch):
    query30 =
"select   sr_remont_1,kap_remont_1,to_1,to_2,to_1x,to_2x,to_2x_pkp,rto,gsm,ykr_brez,ab,tank_shl
em,balon,gys_shin,nr" \
             " from ravbook.nsi_ravbook where id_vch = " + str(
        id_vch) + " and id_rav_grid =" + str(grid_id)
    cursor.execute(query30)
    result = cursor.fetchall()

    sr = result[0][0]
    kr = result[0][1]
    to1 = result[0][2]
    to2 = result[0][3]
    to1x = result[0][4]
    to2x = result[0][5]
    to2x_pkp = result[0][6]
    rto = result[0][7]
    gsm = result[0][8]
    yb = result[0][9]
    ab = result[0][10]
    tsch = result[0][11]
    ppo = result[0][12]
    gys_shin = result[0][13]

    """----------------------------------------------------TO1----------------------------------------------------
"""
    query31 = "select date_rem from main.dok_kart_tex where id_kart = " + str(
        kart) + " and id_chprsn = 0 and id_rem = 502 order by date_rem desc limit 1"
    cursor.execute(query31)
    result = cursor.fetchone()
    if result != None:
        if math.ceil(to1) <= (curent_year - result[0].year):
            to1_value = curent_year
        else:
```

```
            to1_value = result[0].year + math.ceil(to1)
        else:
            to1_value = curent_year

        """----------------------------------------------------TO2----------------------------------------------------
"""
        query32 = "select date_rem from main.dok_kart_tex where id_kart = " + str(
            kart) + " and id_chprsn = 0 and id_rem = 503 order by date_rem desc limit 1"
        cursor.execute(query32)
        result = cursor.fetchone()
        if result != None:
            if math.ceil(to2) <= (curent_year - result[0].year):
                to2_value = curent_year
            else:
                to2_value = result[0].year + math.ceil(to2)
        else:
            to2_value = curent_year

        """----------------------------------------------------TO1x----------------------------------------------------
-"""
        query33 = "select date_rem from main.dok_kart_tex where id_kart = " + str(
            kart) + " and id_chprsn = 0 and id_rem = 508 order by date_rem desc limit 1"
        cursor.execute(query33)
        result = cursor.fetchone()
        if result != None:
            if math.ceil(to1x) <= (curent_year - result[0].year):
                to1x_value = curent_year
            else:
                to1x_value = result[0].year + math.ceil(to1x)
        else:
            to1x_value = curent_year

        """----------------------------------------------------TO2x----------------------------------------------------
-"""
        query34 = "select date_rem from main.dok_kart_tex where id_kart = " + str(
            kart) + " and id_chprsn = 0 and id_rem = 509 order by date_rem desc limit 1"
        cursor.execute(query34)
        result = cursor.fetchone()
        if result != None:
            if math.ceil(to2x) <= (curent_year - result[0].year):
                to2x_value = curent_year
            else:
                to2x_value = result[0].year + math.ceil(to2x)
        else:
            to2x_value = curent_year

        """------------------------------------------------TO2x с ПКП------------------------------------------------
---"""
        query35 = "select date_rem from main.dok_kart_tex where id_kart = " + str(
            kart) + " and id_chprsn = 0 and id_rem = 514 order by date_rem desc limit 1"
        cursor.execute(query35)
        result = cursor.fetchone()
```

```
    if result != None:
        if math.ceil(to2x_pkp) <= (curent_year - result[0].year):
            to2x_pkp_value = curent_year
        else:
            to2x_pkp_value = result[0].year + math.ceil(to2x_pkp)
    else:
        to2x_pkp_value = curent_year

    """----------------------------------------------------PTO----------------------------------------------------
"""
    query36 = "select date_rem from main.dok_kart_tex where id_kart = " + str(
        kart) + " and id_chprsn = 0 and id_rem = 505 order by date_rem desc limit 1"
    cursor.execute(query36)
    result = cursor.fetchone()
    if result != None:
        if math.ceil(rto) <= (curent_year - result[0].year):
            rto_value = curent_year
        else:
            rto_value = result[0].year + math.ceil(rto)
    else:
        rto_value = curent_year

    query37 = "select date_rem from main.dok_kart_tex where id_kart = " + str(
        kart) + " and id_chprsn = 0 and id_rem = 302 order by date_rem desc limit 1"
    cursor.execute(query37)
    result = cursor.fetchone()
    if result != None:
        if math.ceil(gsm) <= (curent_year - result[0].year):
            gsm_value = curent_year
        else:
            gsm_value = result[0].year + math.ceil(gsm)
    else:
        gsm_value = curent_year

    query38 = "select date_rem from main.dok_kart_tex where id_kart = " + str(
        kart) + " and id_chprsn = 0 and id_rem = 303 order by date_rem desc limit 1"
    cursor.execute(query38)
    result = cursor.fetchone()
    if result != None:
        if math.ceil(yb) <= (curent_year - result[0].year):
            yb_value = curent_year
        else:
            yb_value = result[0].year + math.ceil(yb)
    else:
        yb_value = curent_year

    query39 = "select date_rem from main.dok_kart_tex where id_kart = " + str(
        kart) + " and id_chprsn = 0 and id_rem = 304 order by date_rem desc limit 1"
    cursor.execute(query39)
    result = cursor.fetchone()
    if result != None:
        if math.ceil(ab) <= (curent_year - result[0].year):
```

```
        ab_value = curent_year
    else:
        ab_value = result[0].year + math.ceil(ab)
else:
    ab_value = curent_year

query40 = "select date_rem from main.dok_kart_tex where id_kart = " + str(
    kart) + " and id_chprsn = 0 and id_rem = 305 order by date_rem desc limit 1"
cursor.execute(query40)
result = cursor.fetchone()
if result != None:
    if math.ceil(tsch) <= (curent_year - result[0].year):
        tsch_value = curent_year
    else:
        tsch_value = result[0].year + math.ceil(tsch)
else:
    tsch_value = curent_year

query41 = "select date_rem from main.dok_kart_tex where id_kart = " + str(
    kart) + " and id_chprsn = 0 and id_rem = 305 order by date_rem desc limit 1"
cursor.execute(query41)
result = cursor.fetchone()
if result != None:
    if math.ceil(ppo) <= (curent_year - result[0].year):
        ppo_value = curent_year
    else:
        ppo_value = result[0].year + math.ceil(ppo)
else:
    ppo_value = curent_year


query42 = "select date_rem from main.dok_kart_tex where id_kart = " + str(
    kart) + " and id_chprsn = 0 and id_rem = 306 order by date_rem desc limit 1"
cursor.execute(query42)
result = cursor.fetchone()
if result != None:
    if math.ceil(gys_shin) <= (curent_year - result[0].year):
        gys_shin_value = curent_year
    else:
        gys_shin_value = result[0].year + math.ceil(gys_shin)
else:
    gys_shin_value = curent_year

nr_value = curent_year

final_year = curent_year + 10
mas_year_count = 1

while curent_year < final_year:
    resulting_massiv = []

    to = []
```

```
if to1 != None and to1 != 0:
    to1_num = 0
    while int(to1_value) == curent_year:
        to1_num += 1
        to.append("TO-1/" + str(to1_num))
        to1_value += to1

if to2 != None and to2 != 0:
    to2_num = 0
    while int(to2_value) == curent_year:
        to2_num += 1
        to.append("TO-2/" + str(to2_num))
        to2_value += to2

if to1 != None and to2 != None:
    if to1_num != 0 and to2_num != 0:
        to.clear()
        if to1 < 1:
            to1_znam = 1 / to1
        else:
            to1_znam = 1
        if to2 < 1:
            to2_znam = 1 / to2
        else:
            to2_znam = 1

        to1_num = to1_num - int(to2_num)

        if to1_znam == to2_znam:
            for i in range(to1_num):
                to.append("TO-1/" + str(i + 1))
            for i in range(to2_num):
                to.append("TO-2/" + str(i + 1))
        elif to1_znam > to2_znam:
            for i in range(to1_num):
                to.append("TO-1/" + str(i + 1))
            for i in range(to2_num):
                to.append("TO-1/" + str(to1_num) + " TO-2/" + str(i + 1))
resulting_massiv += to

tox = []
if to1x != None and to1x != 0:
    to1x_num = 0
    while int(to1x_value) == curent_year:
        to1x_num += 1
        tox.append("TO-1x/" + str(to1x_num))
        to1x_value += to1x

if to2x != None and to2x != 0:
    to2x_num = 0
    while int(to2x_value) == curent_year:
        to2x_num += 1
```

```
        tox.append("ТО-2х/" + str(to2x_num))
        to2x_value += to2x

if to1x != None and to2x != None:
    if to1x_num != 0 and to2x_num != 0:
        tox.clear()
        if to1x < 1:
            to1x_znam = 1 / to1x
        else:
            to1x_znam = 1
        if to2x < 1:
            to2x_znam = 1 / to2x
        else:
            to2x_znam = 1

        to1x_num = to1x_num - int(to2x_num)

        if to1x_znam == to2x_znam:
            for i in range(to1x_num):
                tox.append("ТО-1х/" + str(i + 1))
            for i in range(to2x_num):
                tox.append("ТО-2х/" + str(i + 1))
        elif to1x_znam > to2x_znam:
            for i in range(to1x_num):
                tox.append("ТО-1х/" + str(i + 1))
            for i in range(to2x_num):
                tox.append("ТО-1х/" + str(to1x_num) + " ТО-2х/" + str(i + 1))
resulting_massiv += tox

if to2x_pkp != None and to2x_pkp != 0:
    while int(to2x_pkp_value) == curent_year:
        resulting_massiv.append("ТО-2х с ПКП")
        to2x_pkp_value += to2x_pkp

if rto != None and rto != 0:
    while int(rto_value) == curent_year:
        resulting_massiv.append("РТО")
        rto_value += rto

if gsm != None and gsm != 0:
    while int(gsm_value) == curent_year:
        resulting_massiv.append("ГСМ")
        gsm_value += gsm

if yb != None and yb != 0:
    while int(yb_value) == curent_year:
        resulting_massiv.append("УБ")
        yb_value += yb

if ab != None and ab != 0:
    while int(ab_value) == curent_year:
        resulting_massiv.append("АБ")
```

```
            ab_value += ab

        if tsch != None and tsch != 0:
            while int(tsch_value) == curent_year:
                resulting_massiv.append("ТШ")
                tsch_value += tsch

        if ppo != None and ppo != 0:
            while int(ppo_value) == curent_year:
                resulting_massiv.append("ППО")
                ppo_value += ppo

        if gys_shin != None and gys_shin != 0:
            while int(gys_shin_value) == curent_year:
                resulting_massiv.append("Гус/ Шина")
                gys_shin_value += gys_shin


        appending_string = ""
        for i in resulting_massiv:
            appending_string += "\"" + str(i) + "\","
        appending_string = "update ravbook.perspective_plan set year_" + str(mas_year_count) + "
='{" + str(
            appending_string[0:len(appending_string) - 1]) + "}' where id_kart = " + str(
            kart) + " and id_grid = " + str(grid_id) + " and id_vch =" + str(cur_vch) + " and group_cat =
" + str(type)
        cursor.execute(appending_string)

        curent_year += 1
        mas_year_count += 1

    return 1

def planing_nr(cursor, id_vch, year, id_vch_main):
    for type in range(2):
        query43 = '''select pp.id_kart,nsi.kap_remont_1,pp.narabotka,nsi.nr,pp.resurs_na_god from
ravbook.perspective_plan as pp,
        ravbook.nsi_ravbook as nsi where pp.id_grid = nsi.id_rav_grid and nsi.selected_prior = true
        and pp.id_vch in ('" + str(id_vch) + "') and nsi.id_vch = '" + str(
            id_vch_main) + "' and group_cat='" + str(
            type) + "' order by id_kart'"
        cursor.execute(query43)
        result = cursor.fetchall()

        for res in result:
            start_year = year
            end_year = year + 10
            i = 1
            if (res[1] != None) and (res[2] != None) and (res[3] != None) and (res[4] != None):
                srok_exp = int((res[1] - (res[2] + res[3])) / res[4])
                nr_year = start_year + srok_exp
                while start_year < end_year:
```

```
            if start_year == nr_year:
                query44 = '''select id_vch_podrasdel,year_''' + str(i) + ''' from
ravbook.perspective_plan as pp , ravbook.nsi_ravbook as nsi
                                    where nsi.id_rav_grid = pp.id_grid and nsi.id_part_rav = 0 and
id_kart = ''' + str(
                    res[0]) + ''' and pp.id_vch in (''' + str(id_vch) + ''') and nsi.id_vch =''' + str(
                    id_vch_main) + ''' and pp.group_cat = ''' + str(type)
                cursor.execute(query44)
                year_datas = cursor.fetchone()
                year_data = year_datas[1]
                year_data.append("НР")

                appending_string = ""
                for data in year_data:
                    appending_string += "\"" + str(data) + "\","
                appending_string = '''update ravbook.perspective_plan set year_''' + str(
                    i) + ''' ='{''' + str(appending_string[0:len(
                    appending_string) - 1]) + '''}' where id = (select distinct pp.id from
ravbook.perspective_plan as pp ,
                                      ravbook.nsi_ravbook as nsi where nsi.id_rav_grid =
pp.id_grid and nsi.id_part_rav = 0 and id_kart = ''' + str(
                    res[0]) + ''' and group_cat=''' + str(type) + ''') and id_vch = ''' + str(year_datas[0])
                cursor.execute(appending_string)
                break
            start_year += 1
            i += 1

def planinig_sr_kr(cursor, id_vch, id_vch_main):
    # по категориям БГ ПР других Родов
    for i in range(2):
        query45 = '''select distinct pp.rav_index_name from ravbook.perspective_plan as pp,
ravbook.nsi_ravbook as nsi
    where group_cat = ''' + str(
            i) + ''' and pp.rav_index_name = nsi.id_rav and nsi.id_part_rav = 0  and nsi.id_vch = ''' +
str(id_vch_main)
        cursor.execute(query45)
        id_prsn = cursor.fetchall()
        # Образцы техники
        for example in id_prsn:
            query46 = '''select pp.id_vch_podrasdel, pp.id_kart,pp.narabotka,nsi.sr_remont_1,
nsi.kap_remont_1,pp.resurs_na_god from ravbook.perspective_plan as pp
                left join ravbook.nsi_ravbook as nsi on (pp.id_grid = nsi.id_rav_grid) where
                ((nsi.selected_prior=true and cast (pp.id_grid as text) like \'%''' + str(
                example[0]) + '''%\') or pp.rav_index_name = ''' + str(example[0]) + ''')
                and pp.id_vch in (''' + str(id_vch) + ''') and nsi.id_vch = ''' + str(
                id_vch_main) + ''' and pp.group_cat = ''' + str(i) + '''
                order by pp.id_kart, pp.id_grid = ''' + str(example[0]) + str(example[0]) + ''' desc'''
            cursor.execute(query46)
            result = cursor.fetchall()

            if result != None:
                query47 = '''delete from ravbook.sr_kr'''
```

```
cursor.execute(query47)
query47_1 = '"SELECT setval('ravbook.sr_kr_id_seq', 1)"'
cursor.execute(query47_1)
item = 0
sr_year = 0
sr_years = []
kr_year = 0
kr_years = []
# Кол образцов
while item < len(result):
    sr_years.clear()
    kr_years.clear()
    sr = result[item]
    narabotka = result[item + 1]
    if (sr[3] - sr[2]) <= 0:
        sr_year = 1
    else:
        sr_year = 1 + sr[3] - sr[2]
    while sr_year < 11:
        sr_years.append(sr_year)
        sr_year += sr[3]

    if (sr[4] - sr[2]) <= 0:
        kr_year = 1
    else:
        kr_year = 1 + sr[4] - sr[2]
    while kr_year < 11:
        kr_years.append(kr_year)
        kr_year += sr[4]

    # Добавление строки для СР для образца
    query48 = '"insert into ravbook.sr_kr(kart,narabotka,sr,kr, type_sr_kr)
        values('" + str(sr[1]) + "'," + str(sr[2]) + "'," + str(sr[3]) + "'," + str(
        sr[4]) + "', \'sr\')"'
    cursor.execute(query48)

    # Добавление строки для КР для образца
    query49 = '"insert into ravbook.sr_kr(kart,narabotka,sr,kr,year_resurs,type_sr_kr)
                            values('" + str(narabotka[1]) + "'," + str(
        narabotka[2]) + "'," + str(
        narabotka[3]) + "'," + str(narabotka[4]) + "'," + str(
        narabotka[5]) + "', \'kr\')"'
    cursor.execute(query49)

    # Установка предпологаемого выхода в СР по НСИ
    for s in sr_years:
        query50 = '"update ravbook.sr_kr set year_'" + str(
            s) + "' = true where type_sr_kr = \'sr\' and kart='" + str(sr[1])
        cursor.execute(query50)
    for k in kr_years:
        # Установка предпологаемого выхода в КР по НСИ
        query51 = '"update ravbook.sr_kr set year_'" + str(
```

```
                    k) + "' = true where type_sr_kr = \'kr\' and kart="' + str(sr[1])
                cursor.execute(query51)

            query52 = "'select * from ravbook.sr_kr where type_sr_kr =\'kr\' and kart="' + str(sr[1])
            cursor.execute(query52)
            res_with_cheks = cursor.fetchone()

            # Установка наработки для образца
            for num_year in range(1, 11):
                query53 = "'update ravbook.sr_kr set year_"' + str(
                    num_year) + "'_narabotka = "' + str(
                    res_with_cheks[2] + num_year * res_with_cheks[5]) + "' where kart="' + str(sr[1])
                cursor.execute(query53)

            item += 2

        # Начало обработки всех образцов по 1 виду номенклатуры для подразделения
        # Определяем кол образцов
        query54 = "select distinct kart from ravbook.sr_kr"
        cursor.execute(query54)
        limit = cursor.fetchall()
        limit = len(limit)

        if limit >= 4:
            limit = int(limit / 100 * 25)
        else:
            limit = 1

        for num_year in range(1, 11):
            query55 = "'select sk.id,sk.kart,sk.type_sr_kr, sk.sr,sk.kr,sk.year_resurs, sk.year_"' +
str(
                num_year) + "'_narabotka,sk.year_"' + str(num_year) + "',
                (select nn.year_"' + str(num_year) + "'_narabotka from ravbook.sr_kr as nn where
nn.kart=sk.kart and nn.type_sr_kr = 'sr') as narab
                from ravbook.sr_kr as sk where year_"' + str(num_year) + "' = true and year_"' + str(
                num_year) + "'_narabotka >= kr and type_sr_kr = 'kr'
                union
                select id,kart,type_sr_kr, sr,kr,year_resurs, year_"' + str(
                num_year) + "'_narabotka,year_"' + str(
                num_year) + "',0 from ravbook.sr_kr where year_"' + str(num_year) + "' = true
                and type_sr_kr = 'sr' and kart not in (select kart from ravbook.sr_kr where year_"' +
str(
                num_year) + "' = true and year_"' + str(num_year) + "'_narabotka >= kr and
type_sr_kr = 'kr')
                order by year_"' + str(
                num_year) + "'_narabotka desc, narab desc, type_sr_kr,kart limit '" + str(limit)
            cursor.execute(query55)
            resulting = cursor.fetchall()

            query56 = "'select id from ravbook.sr_kr where year_"' + str(num_year) + "' = true "'
            cursor.execute(query56)
            ids = cursor.fetchall()
```

```
d = []
for c in ids:
   d.extend(c)

curent_year = []
for res in resulting:
   curent_year.append(res[0])
   d.remove(res[0])

query57 = '"update ravbook.sr_kr set year_'" + str(num_year) + '"=false'"
cursor.execute(query57)

for cur_year in curent_year:
   query58 = '"update ravbook.sr_kr set year_'" + str(
      num_year) + '"=true where id = '" + str(cur_year)
   cursor.execute(query58)

if num_year != 10:
   for next_year in d:
      query59 = '"update ravbook.sr_kr set year_'" + str(
         num_year + 1) + '"=true where id = '" + str(next_year)
      cursor.execute(query59)

   for res in resulting:
      narab_update = num_year
      mnoj = 1

      if res[2] == 'kr':
         new_time_sr = narab_update + sr[3]
         if new_time_sr < 11:
            query61 = '"update ravbook.sr_kr set year_'" + str(
               new_time_sr) + '"=true where kart = '" + str(res[1]) + '" and year_'" + str(
               narab_update) + '"=false'"
            cursor.execute(query61)

         new_time_kr = narab_update + sr[4]
         if new_time_kr < 11:
            query61 = '"update ravbook.sr_kr set year_'" + str(
               new_time_kr) + '"=true where kart = '" + str(
               res[1]) + '" and year_'" + str(narab_update) + '"=true'"
            cursor.execute(query61)

         for sr_update in range(narab_update, int(new_time_sr)):
            if sr_update < 11:
               query62 = '"update ravbook.sr_kr set year_'" + str(
                  sr_update) + '"=false where kart = '" + str(
                  res[1]) + '" and year_'" + str(narab_update) + '"=false'"
               cursor.execute(query62)

         for kr_update in range(narab_update + 1, int(new_time_kr)):
            if kr_update < 11:
```

```
                        query63 = '''update ravbook.sr_kr set year_''' + str(
                            kr_update) + '''=false where kart = ''' + str(
                            res[1]) + ''' and year_''' + str(narab_update) + '''=true'''
                        cursor.execute(query63)

                while narab_update < 10:
                    if res[2] == 'sr':
                        query60 = '''update ravbook.sr_kr set year_''' + str(
                            narab_update + 1) + '''_narabotka=''' + str(
                            res_with_cheks[5] * mnoj) + ''' where id = ''' + str(res[0])
                    else:
                        query60 = '''update ravbook.sr_kr set year_''' + str(
                            narab_update + 1) + '''_narabotka=''' + str(
                            res_with_cheks[5] * mnoj) + ''' where kart = ''' + str(res[1])

                    cursor.execute(query60)
                    narab_update += 1
                    mnoj += 1

        for year in range(1, 11):
            query61 = '''select kart, type_sr_kr from ravbook.sr_kr where year_''' + str(
                year) + ''' = true'''
            cursor.execute(query61)
            selected_res = cursor.fetchall()

            for select in selected_res:
                query62 = '''select year_''' + str(year) + ''' from ravbook.perspective_plan as pp ,
ravbook.nsi_ravbook as nsi
                    where nsi.id_rav_grid = pp.id_grid and nsi.id_part_rav = 0 and id_kart = ''' + str(
                    select[0]) + ''' and pp.id_vch in (''' + str(id_vch) + ''') and nsi.id_vch =''' + str(
                    id_vch_main) + ''' and pp.group_cat=''' + str(i)
                cursor.execute(query62)
                year_data = cursor.fetchone()
                year_data = year_data[0]

                if select[1] == 'sr':
                    year_data.append("СР")
                else:
                    year_data.append("КР")

                appending_string = ""
                for data in year_data:
                    appending_string += "\"" + str(data) + "\","
                appending_string = '''update ravbook.perspective_plan set year_''' + str(
                    year) + ''' ='{''' + str(appending_string[0:len(
                    appending_string) - 1]) + '''}' where id = (select pp.id from
ravbook.perspective_plan as pp ,
                    ravbook.nsi_ravbook as nsi where nsi.id_rav_grid = pp.id_grid and
nsi.id_part_rav = 0 and id_kart = ''' + str(
                    select[0]) + '''and pp.group_cat =''' + str(i) + ''' and pp.id_vch in('''+str(id_vch)+''')
and nsi.id_vch='''+str(id_vch_main)+''')'''
                cursor.execute(appending_string)
```

Файл «plan_er.js»

```
$(document).ready(function () {
    document.getElementById("Loading").style.display = 'block';
    document.getElementById("Loading_go").style.display = 'block';
    document.getElementById("Loading_nsi").style.display = 'block';

    let id_vch = 0
    var test = String(document.location.href).split('?');
    if (test.length === 2) {
        id_vch = parseInt(test[1]);
    } else {
        alert("Пареметры запуска приложения заданы неверно!")
    }

    let date_to = 0
    let status = 0

    function refresh_search() {
        grid_searching = null
        grid_searching_len = 0
        grid_searching_current_el = 0
    }

    $("#form1").on('click', function (event) {
        refresh_search()
        document.getElementById("main").style.display = 'none';
        document.getElementById("nsi").style.display = 'block';
        document.getElementById("day_of_to").style.display = 'none';
        document.getElementById("trudoemk_to").style.display = 'none';
        document.getElementById("queue_of_to").style.display = 'none';
        document.getElementById('form2').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('form3').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('form4').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('form1').style.cssText = "background: linear-gradient(to left,
#e8e8e8, 70%, #8d8d8d66);"
        document.getElementById("task_1_field").style.display = 'none';
        document.getElementById('task1').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById("task_2_field").style.display = 'none';
        document.getElementById('task2').style.cssText = "width: calc(100% - 20px);\n" +
```

```
        "    font-size: 17px;\n" +
        "    margin-top: 10px;\n" +
        "    padding-left: 17px;"
    })
    $("#form2").on('click', function (event) {
        refresh_search()
        document.getElementById("main").style.display = 'none';
        document.getElementById("trudoemk_to").style.display = 'block';
        document.getElementById("day_of_to").style.display = 'none';
        document.getElementById("queue_of_to").style.display = 'none';
        document.getElementById("nsi").style.display = 'none';
        document.getElementById('form1').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('form3').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('form4').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('form2').style.cssText = "background: linear-gradient(to left,
#e8e8e8, 70%, #8d8d8d66);"
        document.getElementById("task_1_field").style.display = 'none';
        document.getElementById('task1').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById("task_2_field").style.display = 'none';
        document.getElementById('task2').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
    })
    $("#form3").on('click', function (event) {
        refresh_search()
        document.getElementById("main").style.display = 'none';
        document.getElementById("day_of_to").style.display = 'block';
        document.getElementById("queue_of_to").style.display = 'none';
        document.getElementById("trudoemk_to").style.display = 'none';
        document.getElementById("nsi").style.display = 'none';
        document.getElementById('form2').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('form1').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('form4').style.cssText = "width: calc(100% - 20px);\n" +
```

```
        "    font-size: 17px;\n" +
        "    margin-top: 10px;\n" +
        "    padding-left: 17px;"
    document.getElementById('form3').style.cssText = "background: linear-gradient(to left,
#e8e8e8, 70%, #8d8d8d66);"
    document.getElementById("task_1_field").style.display = 'none';
    document.getElementById('task1').style.cssText = "width: calc(100% - 20px);\n" +
        "    font-size: 17px;\n" +
        "    margin-top: 10px;\n" +
        "    padding-left: 17px;"
    document.getElementById("task_2_field").style.display = 'none';
    document.getElementById('task2').style.cssText = "width: calc(100% - 20px);\n" +
        "    font-size: 17px;\n" +
        "    margin-top: 10px;\n" +
        "    padding-left: 17px;"
  })
  $("#form4").on('click', function (event) {
    refresh_search()
    document.getElementById("main").style.display = 'none';
    document.getElementById("queue_of_to").style.display = 'block';
    document.getElementById("day_of_to").style.display = 'none';
    document.getElementById("trudoemk_to").style.display = 'none';
    document.getElementById("nsi").style.display = 'none';
    document.getElementById('form2').style.cssText = "width: calc(100% - 20px);\n" +
        "    font-size: 17px;\n" +
        "    margin-top: 10px;\n" +
        "    padding-left: 17px;"
    document.getElementById('form3').style.cssText = "width: calc(100% - 20px);\n" +
        "    font-size: 17px;\n" +
        "    margin-top: 10px;\n" +
        "    padding-left: 17px;"
    document.getElementById('form1').style.cssText = "width: calc(100% - 20px);\n" +
        "    font-size: 17px;\n" +
        "    margin-top: 10px;\n" +
        "    padding-left: 17px;"
    document.getElementById("task_1_field").style.display = 'none';
    document.getElementById('task1').style.cssText = "width: calc(100% - 20px);\n" +
        "    font-size: 17px;\n" +
        "    margin-top: 10px;\n" +
        "    padding-left: 17px;"
    document.getElementById('form4').style.cssText = "background: linear-gradient(to left,
#e8e8e8, 70%, #8d8d8d66);"
    document.getElementById("task_2_field").style.display = 'none';
    document.getElementById('task2').style.cssText = "width: calc(100% - 20px);\n" +
        "    font-size: 17px;\n" +
        "    margin-top: 10px;\n" +
        "    padding-left: 17px;"
  })
  $("#task1").on('click', function (event) {
    refresh_search()
    document.getElementById("main").style.display = 'none';
    document.getElementById("task_1_field").style.display = 'block';
```

```
        document.getElementById("queue_of_to").style.display = 'none';
        document.getElementById("day_of_to").style.display = 'none';
        document.getElementById("trudoemk_to").style.display = 'none';
        document.getElementById("nsi").style.display = 'none';
        document.getElementById('form2').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('form3').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('form1').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('form4').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('task1').style.cssText = "background: linear-gradient(to left,
#e8e8e8, 70%, #8d8d8d66);"
        document.getElementById("task_2_field").style.display = 'none';
        document.getElementById('task2').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
    })
    $("#task2").on('click', function (event) {
        refresh_search()
        document.getElementById("main").style.display = 'none';
        document.getElementById("task_2_field").style.display = 'block';
        document.getElementById("queue_of_to").style.display = 'none';
        document.getElementById("day_of_to").style.display = 'none';
        document.getElementById("trudoemk_to").style.display = 'none';
        document.getElementById("nsi").style.display = 'none';
        document.getElementById('form2').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('form3').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('form1').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
        document.getElementById('form4').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
```

```
        document.getElementById('task2').style.cssText = "background: linear-gradient(to left,
#e8e8e8, 70%, #8d8d8d66);"
        document.getElementById("task_1_field").style.display = 'none';
        document.getElementById('task1').style.cssText = "width: calc(100% - 20px);\n" +
            "    font-size: 17px;\n" +
            "    margin-top: 10px;\n" +
            "    padding-left: 17px;"
    })

    let edizm
    let grid_searching = null
    let grid_searching_len = 0
    let grid_searching_current_el = 0

    let cellclass = function (row, columnfield, value) {
        return 'block';
    }

    let source = {
        datatype: "json",
        datafield: [
            {name: 'index_rav'},
            {name: 'id_part_rav'},
            {name: 'ed_izm'},
            {name: 'year_resource_norm_bg'},
            {name: 'year_resource_norm_pr'},
            {name: 'sr_remont_1'},
            {name: 'kap_remont_1'},
            {name: 'sr_remont_2'},
            {name: 'kap_remont_2'},
            {name: 'spisanie'},
            {name: 'to_1'},
            {name: 'to_2'},
            {name: 'to_1x'},
            {name: 'to_2x'},
            {name: 'to_2x_pkp'},
            {name: 'rto'},
            {name: 'gsm'},
            {name: 'ykr_brez'},
            {name: 'ab'},
            {name: 'tank_shlem'},
            {name: 'gys_shin'},
            {name: 'nr'},
            {name: 'selected_prior', type: 'bool'}
        ],
        localdata: 0,
        id: 'id',
    }

    let dataAdapter = new $.jqx.dataAdapter(source, {
        downloadComplete: function (data, status, xhr) {
        },
```

```
    loadComplete: function (data) {
    },
    loadError: function (xhr, status, error) {
    }
});

let cellsrenderer = function (row, column, value, event) {
    return '<div style="text-align: center; margin-top: 8px;">' + value + '</div>';
}
let columnrenderer = function (value) {
    return '<div style="text-align: center;">' + value + '</div>';
}

// Грид
$("#nsi_grid"
).jqxGrid({
    groupable: true,
    columnsheight: '60px',
    source: dataAdapter,
    scrollBarSize: 10,
    autoRowHeight: false,
    columnsResize: true,
    width: 'calc(100% - 2px)',
    height: 'calc(100% - 50px)',
    editable: true,
    columns: [
        {
            text: 'id',
            dataField: 'index_rav',
            width: 130,
            renderer: columnrenderer,
            cellsrenderer: cellsrenderer,
        },
        {
            text: 'Индекс техники и его<br>составные части',
            dataField: 'id_part_rav',
            width: 220,
            renderer: columnrenderer,
            cellsrenderer: cellsrenderer,
        },
        {
            text: 'Единица<br> измерения<br> ресурса',
            dataField: 'ed_izm',
            width: 150,
            renderer: columnrenderer,
            cellsrenderer: cellsrenderer,
        },
        {
            text: 'Приоритет<br>определения<br>СР и КР<br>по наработке',
            dataField: 'selected_prior',
            width: 150,
            renderer: columnrenderer,
```

```
    columntype: 'checkbox',
    cellbeginedit: function (row, datafield, columntype, event) {
        row_val = $('#nsi_grid').jqxGrid('getrowdata', row)
        id = row_val.id
        id_main = id.substr(0, id.length / 2) + id.substr(0, id.length / 2)
        if ((id.substr(0, id.length / 2)) == (id.substr(id.length / 2)))
            return false;
        else {
            if (event == false) {
                a = $.grep(JSON.parse(source.localdata), function (n, i) {
                    return ((n.index_rav == row_val.index_rav) && (n.id != id_main))
                })
                for (i = 0; i <= a.length - 1; i++) {
                    index_of = $('#nsi_grid').jqxGrid('getrowboundindexbyid', a[i].id);
                    $("#nsi_grid").jqxGrid('setcellvalue', index_of, "selected_prior", false)
                }
                $("#nsi_grid").jqxGrid('setcellvalue', id, "selected_prior", true)
            }
        }
    }
},

{
    text: 'Обычная группа<br>эксплуатации',
    columngroup: 'yearsource',
    dataField: 'year_resource_norm_bg',
    width: 110,
    cellsformat: 'n',
    renderer: columnrenderer,
    cellsrenderer: cellsrenderer,
},
{
    text: 'С повышенным<br>расходом',
    columngroup: 'yearsource',
    dataField: 'year_resource_norm_pr',
    width: 110,
    cellsformat: 'n',
    renderer: columnrenderer,
    cellsrenderer: cellsrenderer,
},

{
    text: 'СР',
    dataField: 'sr_remont_1',
    width: 80,
    cellsformat: 'n',
    renderer: columnrenderer,
    cellsrenderer: cellsrenderer,
    columngroup: 'remont'
},
{
    text: 'КР',
```

```
    dataField: 'kap_remont_1',
    width: 80,
    cellsformat: 'n',
    renderer: columnrenderer,
    cellsrenderer: cellsrenderer,
    // cellclassname: cellclass,

    columngroup: 'remont',
},
{
    text: 'ТО-1',
    dataField: 'to_1',
    width: 80,
    cellsformat: 'n',
    renderer: columnrenderer,
    cellsrenderer: cellsrenderer,
    columngroup: 'to',
},
{
    text: 'ТО-2',
    dataField: 'to_2',
    width: 80,
    cellsformat: 'n',
    renderer: columnrenderer,
    cellsrenderer: cellsrenderer,
    columngroup: 'to',
},
{
    text: 'ТО-1х',
    dataField: 'to_1x',
    width: 80,
    cellsformat: 'n',
    renderer: columnrenderer,
    cellsrenderer: cellsrenderer,
    columngroup: 'to',
},
{
    text: 'ТО-2х',
    dataField: 'to_2x',
    width: 80,
    cellsformat: 'n',
    renderer: columnrenderer,
    cellsrenderer: cellsrenderer,
    columngroup: 'to',
},
{
    text: 'ТО-2х<br>с<br>ПКП',
    dataField: 'to_2x_pkp',
    width: 80,
    cellsformat: 'n',
    renderer: columnrenderer,
    cellsrenderer: cellsrenderer,
```

```
    columngroup: 'to'

},
{
    text: 'РТО',
    dataField: 'rto',
    width: 80,
    cellsformat: 'n',
    renderer: columnrenderer,
    cellsrenderer: cellsrenderer,
    columngroup: 'to'

},
{
    text: 'УБ',
    dataField: 'ykr_brez',
    width: 80,
    cellsformat: 'n',
    renderer: columnrenderer,
    cellsrenderer: cellsrenderer,
    columngroup: 'meropriyatiya'

},
{
    text: 'АБ',
    dataField: 'ab',
    width: 80,
    cellsformat: 'n',
    renderer: columnrenderer,
    cellsrenderer: cellsrenderer,
    columngroup: 'meropriyatiya'

},
{
    text: 'ВБ<br>(ППО,<br>ОУ)',
    dataField: 'balon',
    width: 80,
    cellsformat: 'n',
    renderer: columnrenderer,
    cellsrenderer: cellsrenderer,
    columngroup: 'meropriyatiya'

},
{
    text: 'Гус / Шина',
    dataField: 'gys_shin',
    width: 80,
    cellsformat: 'n',
    renderer: columnrenderer,
    cellsrenderer: cellsrenderer,
    columngroup: 'meropriyatiya'
```

```
            },
            {
                text: 'НР',
                dataField: 'nr',
                width: 80,
                cellsformat: 'n',
                renderer: columnrenderer,
                cellsrenderer: cellsrenderer,
            },

        ],
        columngroups: [
            {text: 'Проводимые мероприятия', align: 'center', name: 'meropriyatiya'},
            {text: 'Годовая норма расхода ресурсов', align: 'center', name: 'yearsource'},
            {text: 'Ремонт', parentgroup: 'meropriyatiya', align: 'center', name: 'remont'},
            {text: 'Техническое обслуживание', parentgroup: 'meropriyatiya', align: 'center', name:
'to'},
        ],
        groups: ['index_rav'],
        ready: function () {
            $('#nsi_grid').jqxGrid('expandallgroups');
            $('#nsi_grid').jqxGrid('hidecolumn', 'index_rav');
            $('#nsi_grid').jqxGrid('pincolumn', "id_part_rav");
            $('#nsi_grid').jqxGrid('pincolumn', "ed_izm");
            $('#nsi_grid').jqxGrid('setcolumnproperty', 'ed_izm', 'editable', false);
            $('#nsi_grid').jqxGrid('setcolumnproperty', 'id_part_rav', 'editable', false);                    },

    });

    function update_grid_data() {
        $.ajax({
            async: false,
            method: 'GET',
            url: 'load_nsi_grid',
            data: {
                'id_vch': id_vch,
            },
            success: function (data) {
                document.getElementById("Loading_nsi").style.display = 'none';
                document.getElementById("Loading_trud").style.display = 'block';
                build_trudoemkost_grid()
                document.getElementById("Loading_trud").style.display = 'none';
                document.getElementById("Loading_queue").style.display = 'block';
                build_queue_grid()
                document.getElementById("Loading_queue").style.display = 'none';
                document.getElementById("Loading_day").style.display = 'block';
                build_date_grid()
                document.getElementById("Loading_day").style.display = 'none';

                index_of_slice = data.indexOf('],')

                edizm = data.substr(index_of_slice + 2)
```

```
            data = data.substr(0, index_of_slice + 1)
            data = data.replace(/None/g, '')
            source.localdata = data

            let dataAdapter = new $.jqx.dataAdapter(source, {
               downloadComplete: function (data, status, xhr) {
               },
               loadComplete: function (data) {
               },
               loadError: function (xhr, status, error) {
               }
            });
            $("#nsi_grid").jqxGrid({source: dataAdapter});
            $("#nsi_grid").jqxGrid({groups: ['index_rav']});
            if (document.getElementById("nsi").style.display === 'block') {
               $('#nsi_grid').jqxGrid('expandallgroups');
            }

         },
      })

   }

   update_grid_data()

   // Поиск в гриде
   $("#nsi_search_input").jqxInput({
      placeHolder: "Поиск по наименовнию"
   })
   $("#nsi_search_input").on('input', function () {
      grid_searching = null
      grid_searching_len = 0
      grid_searching_current_el = 0
      document.getElementById('nsi_search_input').style.cssText = "background: #fff;";
   })
   $("#nsi_search_btn").jqxButton({})
   $("#nsi_search_btn").on('click', function () {
      let input_value = $("#nsi_search_input").val().toLocaleLowerCase().replace(/["
"/*+,\{\}\[<\]>.-]/g, "");
      if (input_value !== '') {
         document.getElementById('nsi_search_input').style.cssText = "background: #fff;";
         if (grid_searching === null) {
            let rows = $('#nsi_grid').jqxGrid('getrows');
            rows = rows.filter(function (item) {
                  return (((item.id_part_rav != "орудие") && (item.id_part_rav != "шасси") &&
(item.id_part_rav != "двигатель")) && (item.id_part_rav.toLocaleLowerCase().replace(/["
"/*+,\{\}\[<\]>.-]/g, "").includes(input_value)))
               }
            )
            if (rows.length === 0) {
               document.getElementById('nsi_search_input').style.cssText = "background: #dc3030;";
            } else {
```

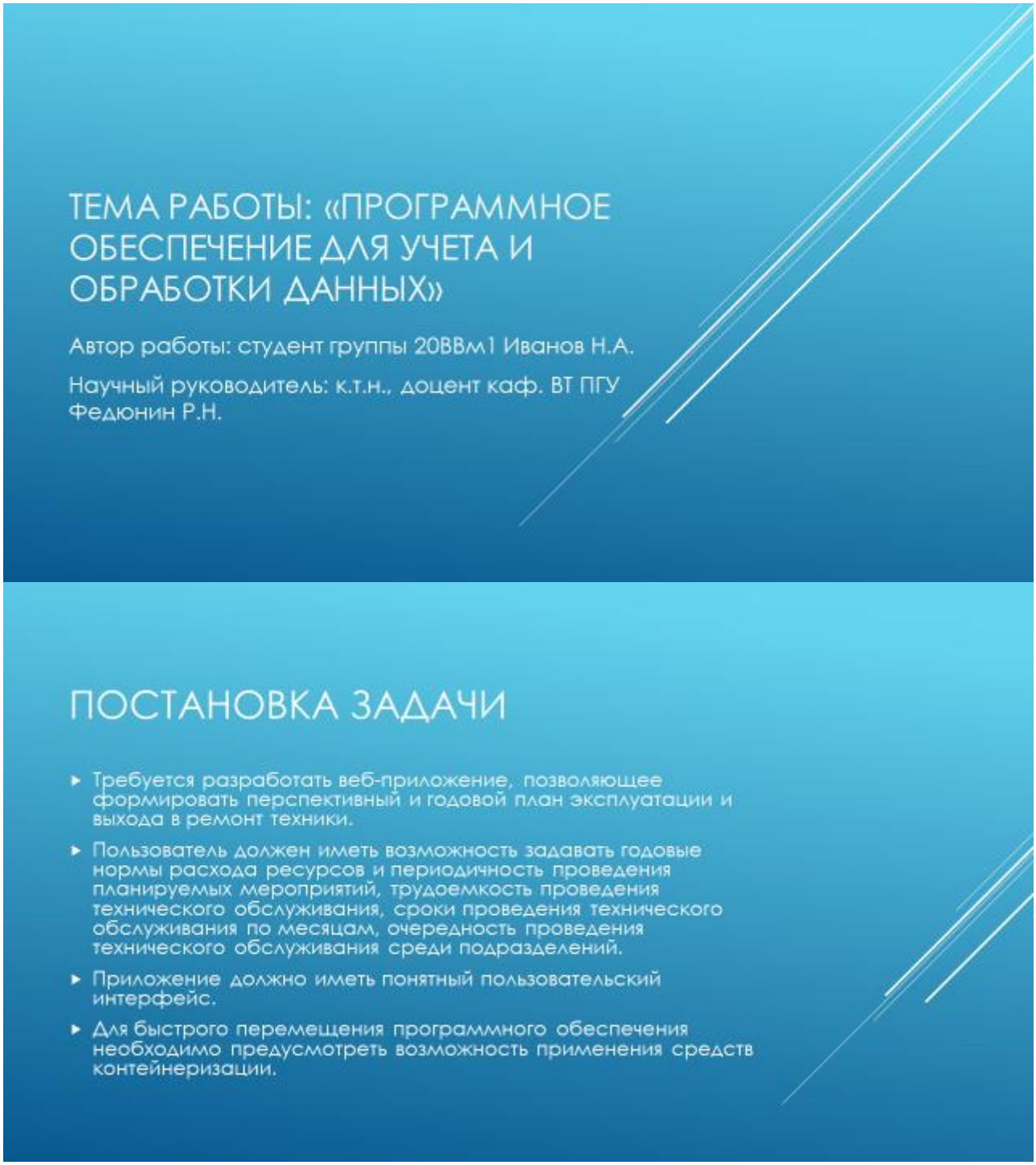
```
                grid_searching = rows
                grid_searching_len = grid_searching.length
                grid_searching_current_el = 0

                let boundindex = $('#nsi_grid').jqxGrid('getrowboundindexbyid', grid_searching[0].id);
                $('#nsi_grid').jqxGrid('selectrow', boundindex)
                $("#nsi_grid").jqxGrid('ensurerowvisible', boundindex);
                grid_searching_current_el++
            }
        } else {
            if (grid_searching_len !== grid_searching_current_el) {
                document.getElementById('nsi_search_input').style.cssText = "background: #fff;";
                let boundindex = $('#nsi_grid').jqxGrid('getrowboundindexbyid',
grid_searching[grid_searching_current_el].id);
                $('#nsi_grid').jqxGrid('selectrow', boundindex)
                $("#nsi_grid").jqxGrid('ensurerowvisible', boundindex);
                grid_searching_current_el++

            } else {
                let boundindex = $('#nsi_grid').jqxGrid('getrowboundindexbyid',
grid_searching[grid_searching_current_el - 1].id);
                $('#nsi_grid').jqxGrid('unselectrow', boundindex);
                grid_searching_current_el = 0
                document.getElementById('nsi_search_input').style.cssText = "background: #ff3131;";
            }
        }
    }
})

$("#nsi_close_btn").jqxButton({})
$("#nsi_refresh_btn").jqxButton({})

$("#nsi_grid").on('cellvaluechanged', function (event) {
    let args = event.args;
    let dataField = event.args.datafield;
    let rowBoundIndex = event.args.rowindex;
    let value = args.value;
    let oldvalue = args.oldvalue;
    let data = $('#nsi_grid').jqxGrid('getrowdata', rowBoundIndex);
    let id = data.id
    let ed = data.ed_izm

    if (value != oldvalue) {
        if (dataField != "ed_izm") {
            $.ajax({
                async: false,
                method: 'GET',
                url: 'update_nsi_grid',
                data: {
                    'id': id,
                    'datafield': dataField,
```

```
                'ed_izm': ed,
                'value': value,
                'id_vch': id_vch,

            },
        })
    } else {
        $.ajax({
            async: false,
            method: 'GET',
            url: 'update_nsi_grid',
            data: {
                'id': id,
                'datafield': 'cur_ed_izm',
                'ed_izm': ed,
                'value': ed,
                'id_vch': id_vch,
            },
            success: function (data) {
                data = data.replace(/None/g, '')
                source.localdata = data

                let dataAdapter = new $.jqx.dataAdapter(source, {
                    downloadComplete: function (data, status, xhr) {
                    },
                    loadComplete: function (data) {
                    },
                    loadError: function (xhr, status, error) {
                    }
                });
                $("#nsi_grid").jqxGrid({source: dataAdapter});
                $("#nsi_grid").jqxGrid({groups: ['index_rav']});
                if (document.getElementById("nsi").style.display === 'block') {
                    $('#nsi_grid').jqxGrid('expandallgroups');
                }
            },
        })
    }
  }
});

$("#nsi_close_btn").on('click', function (event) {
  document.getElementById("nsi").style.display = 'none';
  document.getElementById('form1').style.cssText = "width: calc(100% - 20px);\n" +
    "    font-size: 17px;\n" +
    "    margin-top: 10px;\n" +
    "    padding-left: 17px;"
  document.getElementById("main").style.display = 'block';
})

$("#nsi_refresh_btn").on('click', function (event) {
  update_grid_data()
```

```
})
```

ПРИЛОЖЕНИЕ Б

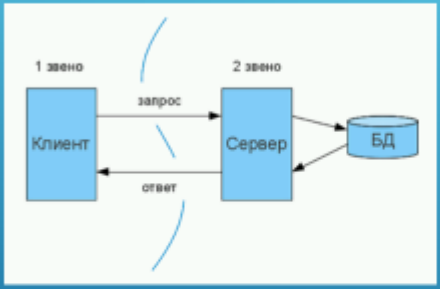
Презентация

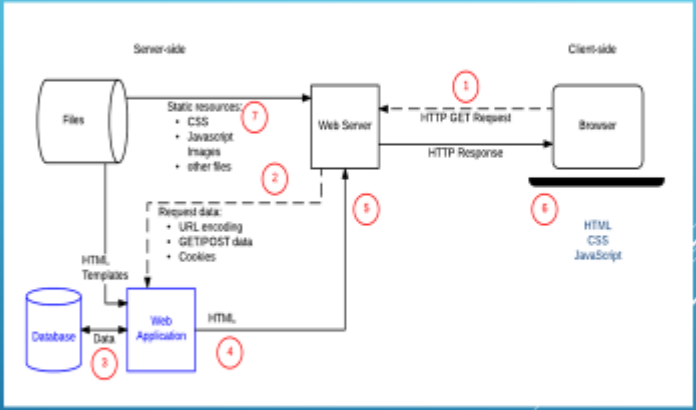


ЦЕЛЬ СОЗДАНИЯ СИСТЕМ ПЛАНИРОВАНИЯ

- минимизация затрат на поддержание работоспособности техники,
- сокращение количества отказа оборудования,
- переход от экстренных ремонтов к плановым,
- предупреждение остановки производственных процессов.

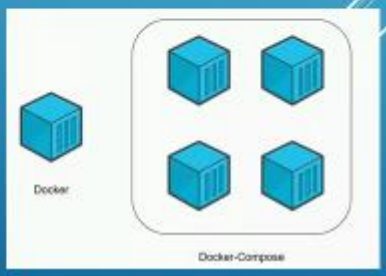
ВЕБ-ПРИЛОЖЕНИЕ



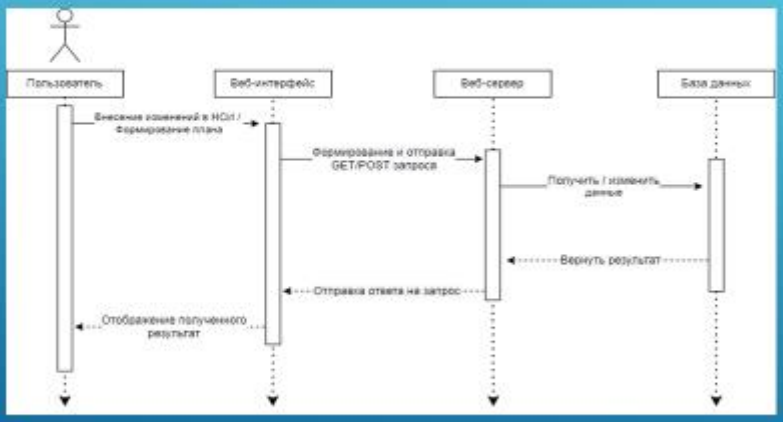


DOCKER

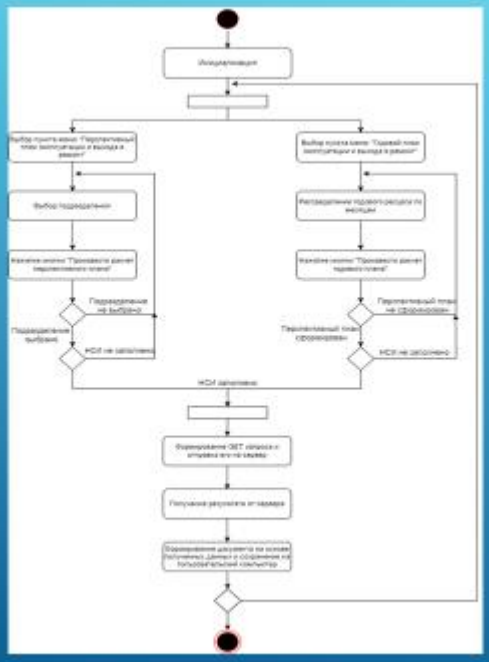
- позволяет ускорить процесс разработки за счет отсутствия необходимости установки вспомогательных инструментов, так как их можно запускать в контейнерах,
- простое масштабирование,
- удобная инкапсуляция приложений,
- предоставляет возможность мониторинга контейнеров.



UML-ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ



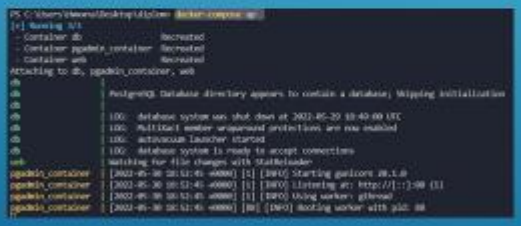
UML-ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ



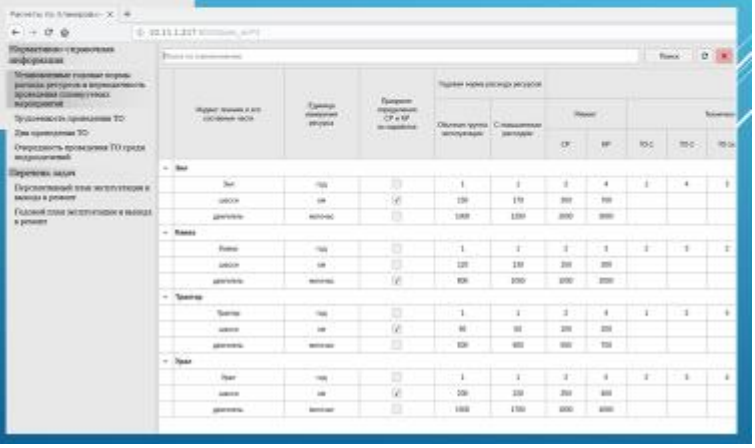
UML-ДИАГРАММА ДЕЯТЕЛЬНОСТИ ФОРМИРОВАНИЯ ПЛАНА СО СТОРОНЫ ПОЛЬЗОВАТЕЛЯ

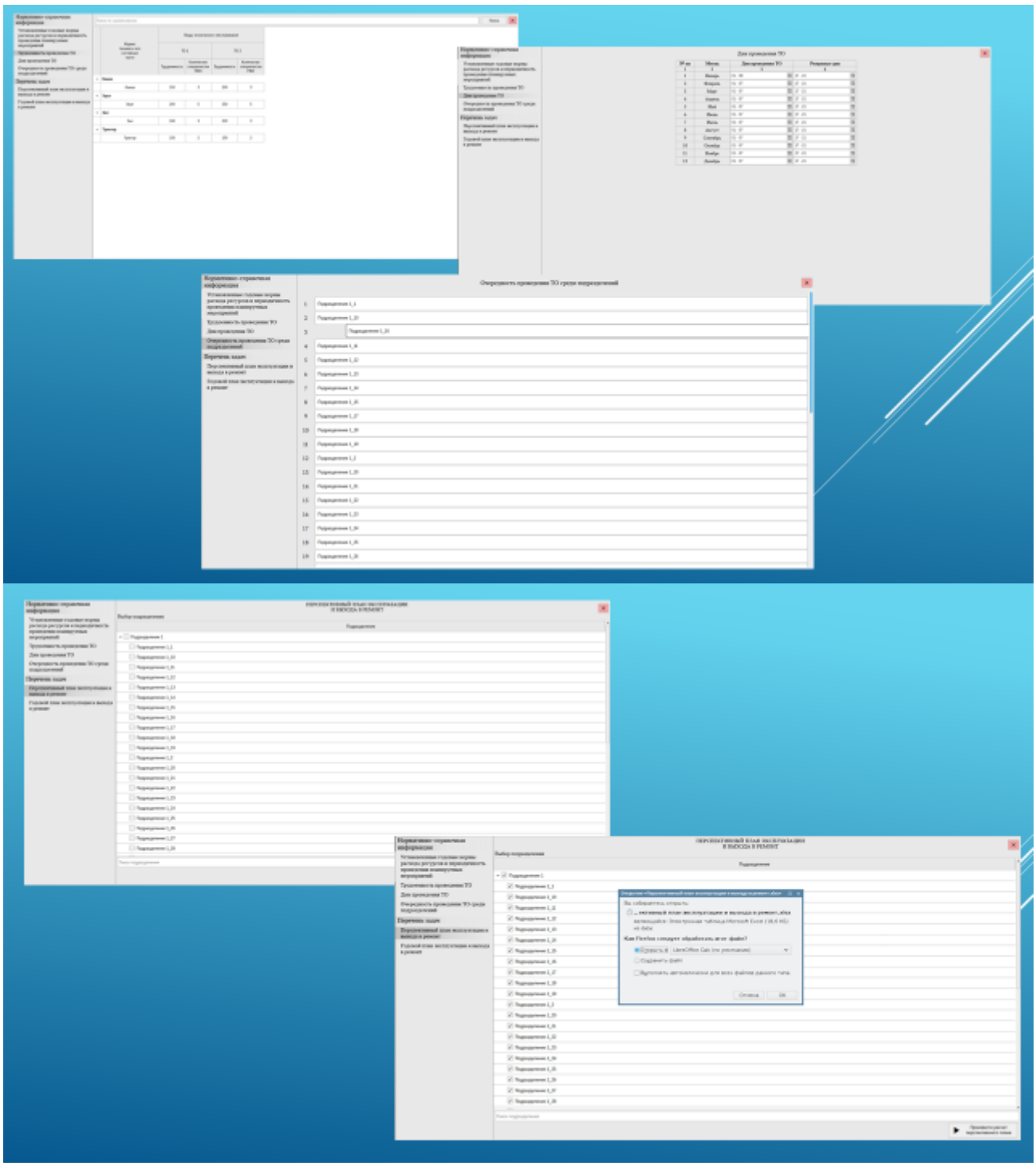
ОТЛАДКА И ТЕСТИРОВАНИЕ

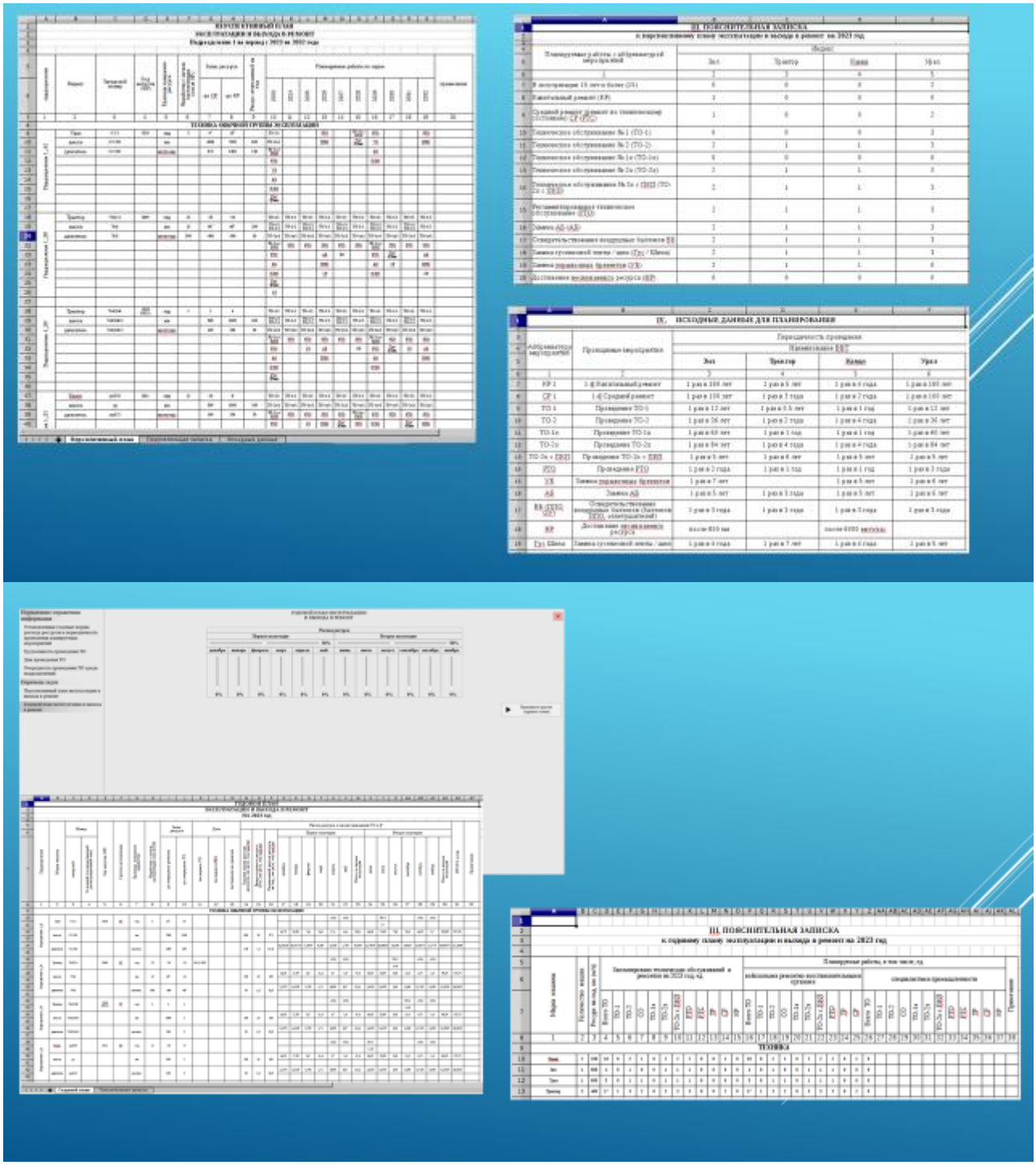
Была выполнена экспериментальная проверка работоспособности путем запуска приложения в контейнере



ГЛАВНОЕ ОКНО ПРИЛОЖЕНИЯ







ЗАКЛЮЧЕНИЕ

- Результатом выполненной работы является веб-приложение, позволяющее формировать перспективный и годовой план эксплуатации и выхода в ремонт техники. Разработанное программное обеспечение предусматривает возможность использования средств контейнеризации.
- В ходе выполнения магистерской диссертации были изучены современные веб-фреймворки и возможность их применения для разработки веб-приложений. Также были получены навыки объектно-ориентированного программирования на языке python.

Спасибо за внимание!