

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт кибернетики и информационных технологий

Кафедра "Программная инженерия"

Дю Е.Ю.

Разработка динамического редактора документов

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

Специальность 5В070400 – Вычислительная техника и программное
обеспечение

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт кибернетики и информационных технологий

Кафедра "Программная инженерия"

5B070400 – Вычислительная техника и программное обеспечение

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой ПИ

канд. техн. наук, доцент,

ассистент-профессор

_____ Р.Юнусов

" ____ " _____ 2020 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

На тему: " Разработка динамического редактора документов"

по специальности 5B070400 – Вычислительная техника и программное
обеспечение

Выполнила

Дю Е.Ю.

Рецензент

канд. техн. наук

_____ А.А. Пак

" ____ " _____ 2020 г.

Научный руководитель

канд. техн. наук, доцент

_____ Р. Юнусов

" ____ " _____ 2020 г.

Алматы 2020

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Дю Елена Юрьевна

Название: Динамический редактор документов

Координатор: Жибек Алибиева

Коэффициент подобия 1: 5

Коэффициент подобия 2: 0,7

Замена букв: 0

Интервалы: 0

Микропробелы: 0

Белые знаки: 0

После анализа Отчета подобия констатирую следующее:

- ☐ обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- ☐ обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- ☐ обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....

.....
Дата

.....
Подпись Научного руководителя

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт кибернетики и информационных технологий

Дю Елена Юрьевна

5B070400 – Вычислительная техника и программное обеспечение

ОТЗЫВ НАУЧНОГО РУКОВОДИТЕЛЯ

к дипломному проекту

Тема: «Разработка динамического редактора документов».

Работа самостоятельна и выполнена на высоком уровне. Используются современные технологии и программные платформы.

Работа состоит из 4-х разделов, включая введение, заключение и приложение.

В первом разделе работы представлено общее описание. Проведен анализ и описание предметной области, также раздел раскрывает цель разработки динамического редактора.

Во втором разделе представлены технологии для создания динамического редактора. Проведено сравнение используемых технологий и выбраны лучшие для данного проекта.

В третьем разделе описана проектная часть дипломного проекта. Представлена архитектура системы взаимодействия. Имеется диаграмма деятельности, которая визуально представляет последовательность действий. Также в разделе представлено проектирование пользовательского интерфейса. Наглядно продемонстрированы и описаны все окна редактора.

В четвертом разделе был описан функционал динамического редактора, и то, как данные функции были реализованы.

Дипломный проект выполнен с учетом всех требований, предъявляемых к дипломному проекту по специальности 5B070400 – «Вычислительная техника и программное обеспечение», студентка Дю Е. Ю. рекомендована к защите дипломного проекта и заслуживает присвоения академической степени «бакалавра» по специальности 5B070400 – «Вычислительная техника и программное обеспечение».

Научный руководитель: канд. техн. наук, доцент, ассистент-профессор
_____ Р.Юнусов.

"__" _____ 2020г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт информационных и телекоммуникационных технологий

Кафедра "Программная инженерия"

5B070400 – Вычислительная техника и программное обеспечение

УТВЕРЖДАЮ

Заведующий кафедрой ПИ
канд. техн. наук, доцент,
ассистент-профессор

_____ Р. Юнусов
" ____ " _____ 2020 г.

ЗАДАНИЕ

на выполнение дипломного проекта

Обучающемуся Дю Елене Юрьевне

Тема: Разработка динамического редактора документов

Исходные данные к дипломному проекту: Описание необходимых функций проекта.

Перечень подлежащих разработке в дипломном проекте вопросов:

- а) реализация функциональных возможностей динамического редактора
- б) проектирование и разработка пользовательского интерфейса;
- в) разработка, отладка и тестирование динамического редактора;

Перечень графического материала (с точным указанием обязательных чертежей): представлены 14 слайдов презентации.

ГРАФИК

подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
1. Анализ предметной области, разработка технического задания		
2. Выбор базового компонента		
3. Разработка дизайна интерфейса		
4. Разработка функционала динамического редактора		
5. Тестирование динамического редактора		
6. Написание пояснительной записки к дипломному проекту		

Подписи

консультантов и нормоконтролера на законченный дипломный проект
с указанием относящихся к ним разделов проекта

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Программное обеспечение	Кайырбеков А. М. магистр техн. наук, лектор		
Нормоконтролер	Алибиева Ж. М. магистр техн. наук, лектор		

Научный руководитель _____ Р. Юнусов

Задание принял к исполнению обучающийся _____ Дю Е. Ю.

Дата " ____ " _____ 2020г.

АННОТАЦИЯ

Данный дипломный проект посвящен разработке динамического редактора документов в рамках проекта LIMS. LIMS (сокр. от англ. Laboratory Information Management System, система управления лабораторной информацией) – это программное обеспечение, которое позволяет управлять образцами, результатами испытаний, документами и связанными данными для повышения производительности лаборатории.

Разработанный инструмент позволит упростить работу пользователей, работающих с документами в LIMS. Удобство заключается в сокращении времени создания и редактирования документов, пользователю нет необходимости отслеживать актуальность данных так как все данные в документах обновляются автоматически, при изменении их в системе.

Для реализации системы использовался фреймворк для web-разработки Angular 8. В качестве языка программирования Angular использует TypeScript, который представляет собой язык программирования от Microsoft. Web-приложение разрабатывалось в редакторе исходного кода Visual Studio Code.

АНДАТПА

Бұл дипломдық жоба LIMS жобасының бөлігі ретінде динамикалық құжат редакторын әзірлеуге арналған. LIMS (қысқаша ағылшын тілінен Laboratory Information Management System, зертханалық ақпаратты басқару жүйесі) - бұл зертхананың өнімділігін арттыру үшін үлгілерді, сынақ нәтижелерін, құжаттар мен соған сәйкес деректерді басқаруға мүмкіндік беретін бағдарламалық жасақтама.

Әзірленген құрал LIMS құжаттарымен жұмыс жасайтын пайдаланушылардың жұмысын жеңілдетеді. Жасақтаманың ерекшелігі - құжаттарды құру және өңдеу уақытын азайту, құжаттардағы барлық мәліметтер жүйеде өзгерген кезде автоматты түрде жаңартылатындықтан, пайдаланушыға деректердің маңыздылығын бақылау қажет еместігінде.

Жүйені жүзеге асыру үшін Angular 8 веб-әзірлеу фреймворкі қолданылды, Angular программалау тілі ретінде Microsoft-тан бағдарламалау тілі болып табылатын TypeScript-ті қолданады. Веб-бағдарлама Visual Studio Code бастапқы код редакторында жасалған.

ANNOTATION

This graduation project is dedicated to the development of a dynamic document editor as part of the LIMS project. LIMS (short for Laboratory Information Management System, Laboratory Information Management System) is software that allows you to manage samples, test results, documents and related data to increase laboratory productivity.

The developed tool will simplify the work of users working with documents in LIMS. Convenience is to reduce the time of creating and editing documents, the user does not need to monitor the relevance of the data, since all data in the documents is updated automatically when they change in the system.

To implement the system, the Angular 8 web development framework was used. Angular uses TypeScript, which is a programming language from Microsoft, as the programming language. The web application was developed in the Visual Studio Code source code editor.

СОДЕРЖАНИЕ

	Введение	9
1	Исследовательский раздел	10
1.1	Цель разработки	10
1.2	Термины и сокращения	10
1.3	Предметная область	12
1.3.1	LIMS	12
1.3.2	Основные преимущества LIMS	13
1.3.3	Основные функции LIMS	13
2	Технологический раздел	15
2.1	Фреймворк Angular	15
2.2	Среда разработки	15
2.3	Редакторы HTML	15
3	Проектная часть	18
3.1	Архитектура системы взаимодействия	18
3.2	Описание диаграммы деятельности	18
3.3	Разработка пользовательского интерфейса	21
4	Экспериментальный раздел	27
4.1	Интеграция CkEditor4 и фреймворка Angular	28
4.2	Метод обновления элементов	29
4.2.1	Атрибуты маркеров	29
4.2.2	Получение HTML компонентов	30
	Заключение	32
	Список использованной литературы	33
	Приложение А. Техническое задание	34

ВВЕДЕНИЕ

Лабораторная информационная система управления (LIMS), иногда называемая лабораторной информационной системой или лабораторной системой управления, является программным решением с функциями, поддерживающими работу современной лаборатории. Ключевые функции включают, но не ограничиваются, поддержкой рабочих процессов и отслеживания данных, гибкую архитектуру и интерфейсы обмена данными.

В последнее время функциональность LIMS вышла за пределы первоначальной цели управления. Управление и анализ данных были добавлены во многие системы LIMS, что позволяет полностью автоматизировать работу лаборатории в рамках единого программного решения.

Помимо ключевых функций управления выборками, интеграции инструментов и приложений и электронного обмена данными, в LIMS существует множество дополнительных операций, которыми можно управлять. Одна из них — это управление документами. Эта операция включает в себя обработку и преобразование данных в определенные форматы, управление тем, как документы редактируются.

Любая деятельность компании отражается в документах, и, чтобы улучшить качество рабочих бизнес-процессов, необходимо улучшить процесс работы с документами, т. е. оптимизировать его.

Актуальность данного проекта заключается в том, что он упрощает работу пользователей LIMS, облегчает процесс формирования отчетов и протоколов, повышает эффективность рабочего процесса. Кроме того, разработанный инструмент может быть модернизирован и расширен при появлении новых требований, а также использован при решении схожих задач.

1 Исследовательский раздел

1.1 Цель разработки

Необходимо разработать динамический редактор документов с возможностью:

- стандартной работы с фрагментами текста документа (копирование, удаление, перемещение, корректировка текста в режимах замены или вставки символов и т.д.);
- автоматической загрузки шаблона документа;
- блокировки документа при завершении исследования;
- отрисовки Angular компонентов в редакторе;
- добавления в документ необходимых данных из системы;
- добавления содержания, списков таблиц, графиков и изображений;
- автоматического добавления новых элементов, если они появляются в системе и должны присутствовать в отчете;
- навигации по тексту;

Все данные и элементы должны автоматически обновляться, при их изменении в системе.

Достоинство данного инструмента состоит в сокращении времени работы с документами и избавлении пользователя от необходимости отслеживания актуальности данных.

Для получения готового отчета, пользователю всего лишь нужно его открыть и при надобности отредактировать или добавить новые данные. При внесении изменений в документ, его можно сохранить и в системе будет храниться исправленная версия. Также есть возможность скачать документ в формате docx.

1.2 Термины и сокращения

Термины и сокращения, использованные при разработке, а также связанные с реализацией и используемыми технологиями описаны в таблице 1.

Таблица 1 - Термины, сокращения, и их определения

Сокращение или термин	Определение
LIMS	(сокр. от англ. Laboratory Information Management System) – это лабораторная система управления информацией

Фреймворк	Каркас, который состоит из множества различных библиотек, которые облегчают разработку программного продукта или сайта
IDE	(сокр. от англ. Integrated development environment) Интегрированная среда разработки — это программное приложение, которое предоставляет компьютерным программистам широкие возможности для разработки программного обеспечения.
HTML	(сокр. от англ. Hypertext Markup Language) Язык гипертекстовой разметки, является стандартным языком разметки для документов, предназначенных для отображения в веб-браузере
CSS	(сокр. от англ. Cascading Style Sheets) Каскадные таблицы стилей - это язык таблиц стилей, используемый для описания представления документа, написанного на языке разметки, например HTML.
XML	(сокр. от англ. Extensible Markup Language) Расширяемый язык разметки - это язык разметки, который определяет набор правил для кодирования документов в формате, удобном для чтения и считываемом компьютером.
FTP	(сокр. от англ. File Transfer Protocol) протокол передачи файлов по сети.
WebDav	(сокр. от англ. Web Distributed Authoring and Versioning) Это расширение протокола передачи гипертекста (HTTP), которое позволяет клиентам выполнять удаленные операции по созданию веб-контента.
Subversion	Свободная централизованная система управления версиями.

Git	Это бесплатная система управления версиями с открытым исходным кодом, предназначенная для быстрой и эффективной работы с небольшими и очень крупными проектами.
API	(сокр. от англ. Application programming interface) Интерфейс прикладного программирования — это вычислительный интерфейс к программному компоненту или системе, который определяет, как другие компоненты или системы могут его использовать.
UML	(сокр. от англ. Unified Modeling Language) Унифицированный язык моделирования является универсальным языком моделирования для разработки в области разработки программного обеспечения, предназначенным для обеспечения стандартного способа визуализации проектирования системы.
UI	(сокр. от англ. User interface) Пользовательский интерфейс.
npm	(сокр. от англ. Node Package Manager) Менеджер пакетов для языка программирования JavaScript.
HTTP	(сокр. от англ. Hypertext Transfer Protocol) Протокол передачи гипертекста — это прикладной протокол для распределенных, совместных, гипермедиа информационных систем.

1.3 Предметная область

1.3.1 LIMS

Перед началом разработки необходимо изучить предметную область и дать определение понятию LIMS. В процессе исследования выяснилось, что LIMS (сокр. от англ. Laboratory Information Management System) – это лабораторная система управления информацией, другими словами, это программное обеспечение, которое позволяет эффективно управлять образцами и данными. Лабораторная система управления информацией LIMS позволяет эффективно управлять потоком проб и связанных данных для повышения эффективности лаборатории. LIMS помогает стандартизировать рабочие процессы, тесты и процедуры, обеспечивая при этом точное управление процессом. Приборы могут быть интегрированы в LIMS для автоматизации сбора данных испытаний, гарантируя, что они должным образом откалиброваны и эксплуатируются только обученным персоналом.

Лабораторная система управления информацией

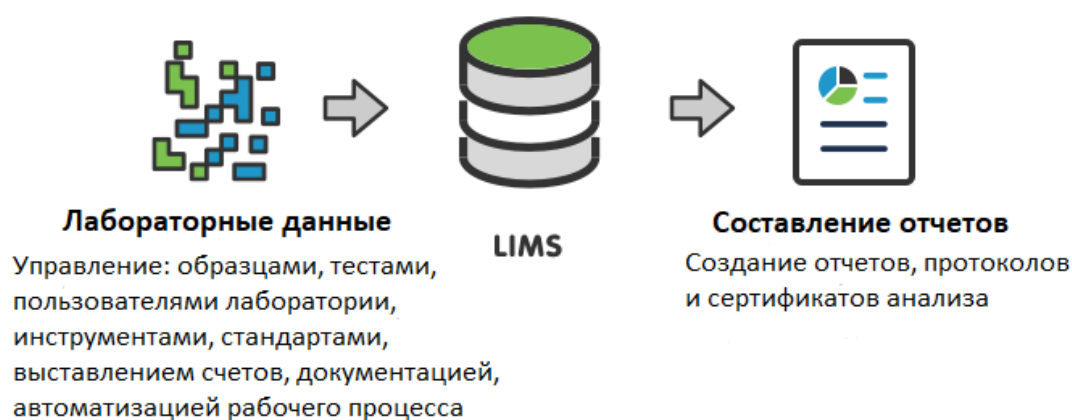


Рисунок-1.3.1 – Лабораторная система управления информацией

1.3.2 Основные преимущества LIMS

LIMS (или лабораторная система управления информацией в ее полной форме) помогает в трех основных направлениях:

- **точность** - LIMS помогает лабораториям получать точные, воспроизводимые результаты;

- эффективность - LIMS помогает лабораториям автоматизировать и ускорить трудоемкие задачи;
- управление - LIMS помогает лабораториям хранить и быстро получать результаты.

1.3.3 Основные функции LIMS

Основная цель ЛИМС - повысить эффективность и точность лабораторных работ за счет сокращения ручных операций. Система LIMS будет выполнять ряд основных функций. Обычно они включают в себя:

- управление образцами;

LIMS обычно используются в Sample Management для точного учета каждого образца. Подробная информация может быть записана при создании образца или его первом поступлении в лабораторию, что может быть улучшено и расширено на протяжении всего жизненного цикла. Обычно это включает в себя источник образца, имена исследователей лаборатории, работающих над ним, и какие части рабочего процесса он прошел. Также информация включает в себя информацию, такую как, как образец должен быть сохранен и любые сроки годности. Использование LIMS для хранения этой информации снижает риск потери, загрязнения или истечения образца.

- управление инвентарем;

Отслеживание инвентаря является важной частью повседневного управления лабораторией. Система LIMS может полностью автоматизировать этот процесс, гарантируя, что расходные материалы заказываются заранее в соответствии с требованиями. Это поддерживает рабочий процесс лаборатории и предотвращает задержки или проблемы, возникающие из-за нехватки запасов.

- управление рабочим процессом;

LIMS можно использовать для упрощения лабораторного рабочего процесса, от автоматического назначения задач исследователям до определения того, куда необходимо передать образец для его следующего шага в рабочем процессе. Он даже может быть запрограммирован, чтобы предлагать инструменты и оборудование, основываясь на заранее установленных правилах и критериях.

- составление отчетов;

LIMS является важным инструментом для поддержания точной и своевременной отчетности. Система может автоматизировать множество ежедневных, еженедельных и ежемесячных отчетов. Кроме того, можно быстро создавать специальные мгновенные отчеты, что позволяет работникам лаборатории принимать обоснованные решения в режиме реального времени. Более того, показания могут быть взяты непосредственно с оборудования и

поданы непосредственно в систему для создания отчетов с максимальной возможной точностью.

2 Технологический раздел

2.1 Фреймворк Angular

Angular - это JavaScript-фреймворк, разработанный Google в 2009 году для помощи разработчикам в инструментах для создания мобильных и веб-приложений. Фреймворк также позволяет интегрироваться с внешними библиотеками и легко расширяется.

Angular написан на TypeScript. Он реализует основные и дополнительные функции в виде набора библиотек TypeScript, которые вы импортируете в свои приложения.

2.2 Среда разработки

Интегрированная среда разработки (IDE) — это программное обеспечение, которое вы можете загрузить на свой компьютер. Они предназначены для упрощения процесса веб-разработки, как упоминалось ранее.

Рассмотрим лучшие IDE для Angular, которые необходимо использовать и которые могут помочь в разработке веб-приложений.

Webstorm - это мощный редактор кода, созданный IntelliJ, разработанный JetBrains, и отличный выбор для кодирования приложений Angular на основе TypeScript. Он включает в себя не только автозаполнение кода и оперативное обнаружение ошибок, но также имеет отладчик, модульное тестирование и множество интеграций, таких как Git, GitHub, Mercurial и другие. Бесплатная пробная версия действует 30 дней, далее необходимо оплатить лицензию.

Visual Studio Code - это редактор исходного кода, который поддерживает TypeScript, включая отладку, интеллектуальное завершение кода на основе типов переменных, функций и импортированных модулей. Он предоставляет различные функции, такие как фрагменты кода и рефакторинг кода. Visual Studio Code предлагает подсветку синтаксиса и автозаполнение с IntelliSense. Visual Studio IDE — это один из самых популярных и лучших вариантов веб-разработки IDE.

Sublime Text — это сложный текстовый редактор для кода и разметки. Имеет приятный пользовательский интерфейс, необычные функции и хорошую производительность. Sublime Text можно скачать и оценить бесплатно, однако для дальнейшего использования необходимо приобрести лицензию.

Рассмотрев все вышеперечисленные IDE, выбор был сделан в пользу Visual Studio Code.

2.3 Редакторы HTML

Редактор HTML — это программа для редактирования HTML, разметки веб-страницы. Хотя разметкой HTML на веб-странице можно управлять с помощью любого текстового редактора, специализированные редакторы HTML могут предложить удобство и дополнительную функциональность. Например, многие редакторы HTML обрабатывают не только HTML, но и связанные технологии, такие как CSS, XML и JavaScript.

Существует два основных варианта редакторов HTML: текстовые и WYSIWYG.

WYSIWYG (является аббревиатурой от англ. What You See Is What You Get, «что видишь, то и получишь») – HTML редакторы предоставляют интерфейс редактирования, который показывает, как страница будет выглядеть в веб-браузере. Поскольку использование редактора WYSIWYG может не требовать каких-либо знаний HTML, неопытному пользователю компьютера зачастую легче начать работу с ним. WYSIWYG-представление достигается путем встраивания механизма компоновки. Это может быть написано на заказ или основано на том, что используется в веб-браузере. Цель состоит в том, чтобы во время редактирования отображаемый результат представлял то, что будет видно позже в обычном веб-браузере.

Для реализации поставленных задач были рассмотрены WYSIWYG редакторы Editor.js, CkEditor4 и CkEditor5.

Editor.js – это блочный редактор для мультимедийных историй. Он выводит чистые данные в JSON вместо тяжелой HTML-разметки. И еще важнее то, что Editor.js разработан с возможностью расширения и подключения API.

CkEditor 4 – это проверенный, WYSIWYG HTML-редактор корпоративного уровня с широкой совместимостью с браузерами, включая устаревшие браузеры. Имеет ряд преимуществ таких как, вставка из Word и Excel, проверка орфографии, проверка доступности, таблицы, автозаполнение, виджеты, фрагменты кода, Emoji плагины. Полный контроль над контентом: фильтрация HTML, режим просмотра исходного кода. Долгосрочная поддержка (сокр от англ. LTS - Long-term support) до 2023 года. В редакторе легко настроить панель инструментов с помощью конфигуратора панели инструментов.

CKEditor 5 - это проект, который позволяет вам быстро и легко инициализировать один из многих типов редакторов, которые он предлагает в вашем приложении. В то же время, это основа для создания индивидуальных решений для редактирования текста. Прежнее требование выполнено благодаря CKEditor 5 Builds. Последнее - благодаря CKEditor 5 Framework.

По техническим характеристикам наиболее подходящими оказались редакторы CkEditor4 и CkEditor5.

CkEditor5 предоставляет больше технических возможностей благодаря CkEditor5 framework, однако разработка собственных решений вызывает трудности по причине необходимости осваивать CkEditor5 framework, также при

разработке возникли сложности интеграции с Angular компонентами. С другой стороны, CkEditor4 даёт возможность использования редактора в качестве компонента Angular, с последующим изменением конфигурации и добавления собственного функционала, посредством библиотеки ng2-ckeditor.

3 Проектная часть

3.1 Архитектура системы взаимодействия

Разрабатываемый инструмент является частью лабораторной системы управления информацией. В рамках процесса отчетности, первая часть взаимодействия выглядит следующим образом: по запросу frontend части, разработанной с использованием фреймворка Angular версии 8, с сервера приходит HTML-текст документа. На следующем этапе фреймворк взаимодействуя с редактором Sckeditor4, обрабатывает документ. Последний этап при взаимодействии — это сохранение документа: обновленный HTML-текст документа отправляется на сервер. Архитектура инструмента изображена на рисунке 3.1.

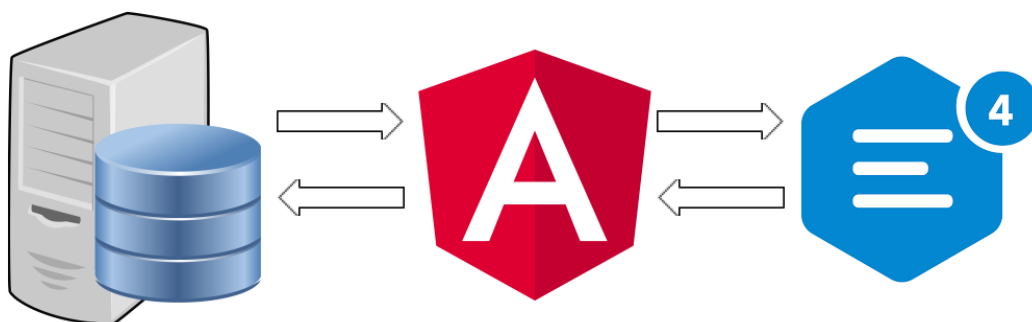


Рисунок-3.1.1 – Архитектура системы взаимодействия

3.2 Описание диаграммы деятельности

Диаграмма деятельности (диаграмма активностей) – UML диаграмма, которая визуально представляет последовательность действий или поток управления в системе, аналогичный блок-схеме или диаграмме потока данных. Диаграммы деятельности часто используются при моделировании бизнес-процессов. Они также могут описать шаги в диаграмме вариантов использования. Моделируемые действия могут быть последовательными и параллельными. В обоих случаях диаграмма действий будет иметь начало (начальное состояние) и конец (конечное состояние).

Для разработки диаграммы деятельности было использовано бесплатное автономное приложение draw.io для создания диаграмм, созданное технологическими лидерами в области веб-диаграмм, компанией Google.

Работа редактора начинается с получения документа с сервера. На сервере хранятся шаблоны документов. Шаблоны имеют свою структуру и весь набор данных необходимый для конкретного документа.

Элементы, которые автоматически обновляются, имеют атрибуты, которые содержат информацию, используемую для корректного обновления.

В процессе разработки формат атрибутов терпел некоторые изменения, по этой причине в редакторе предусмотрено обновление атрибутов до актуальной версии.

После обновления атрибутов, начинается поиск элементов для обновления, затем в конце HTML текста документа находится комментарий, который содержит массивы таблиц и графиков, которые были обработаны пользователем, т.е. те, что пользователь увидел в отчете и посчитал нужным оставить, или же удалить. Если в системе для этого документа существует таблицы или графики, которые отсутствуют в массиве обработанных, то эти элементы отображаются за пределами видимости пользователя. Этот момент реализуется для получения HTML элементов. После получения HTML, происходит добавление новых элементов в документ. Следующим действием является обновления комментария, содержащего массивы обработанных таблиц и графиков. После обновления комментария обновляются все элементы, имеющие соответствующие атрибуты.

В редакторе реализована функция разметки заголовков разных уровней. Заголовок каждого уровня имеет соответствующий атрибут. В заключении процесса обновления документа начинается процесс поиска элементов с атрибутами, указывающими на то, что этот элемент является заголовком определенного уровня. Найденные элементы сортируются в массивы, в зависимости от уровня и типа. Существует четыре вида заголовков: заголовок 1 уровня, 2 уровня, заголовок таблицы, графика и изображения. Заголовки 1 и 2 уровней отображаются в панели навигации. В дальнейшем эти массивы используются для отображения компонента содержания. Впоследствии редактор получает HTML компонента и добавляет его в документ.



Рисунок-3.2.1 – Диаграмма деятельности

3.3 Разработка пользовательского интерфейса

Разработка пользовательского интерфейса — это разработка веб-сайтов, веб-приложений, мобильных приложений и разработка программного обеспечения. «Пользовательский интерфейс» играет ключевую роль в жизненном цикле разработки программного обеспечения. Большинство людей предполагают, что решения для разработки пользовательского интерфейса (UI) создают веб-сайты и пишут HTML, CSS и JavaScript, но пользовательский интерфейс выходит далеко за рамки этих технических терминов. Цель пользовательского интерфейса - сделать взаимодействие с пользователем максимально простым и эффективным с точки зрения достижения целей пользователя.

Для разработки пользовательского интерфейса было использовано приложение Figma. Figma - это инструмент для веб-дизайна с возможностью совместной работы в режиме реального времени.

В рамках системы LIMS редактор реализован в виде модального окна. Левая часть окна содержит навигационную панель, в правой части окна находится сам редактор. В верхней части редактора, находится панель инструментов, которая содержит 20 кнопок. Каждая кнопка отвечает за определенный функционал. Часть кнопок представляют собой встроенные возможности CkEditor4. Рассмотрим функционал кнопок, реализованных в рамках данного проекта:

- Сохранить. Отправляет содержимое редактора на сервер;
- Скачать как Word. Скачивает документ в формате docx;
- Вставить таблицу / график. При нажатии на кнопку перед пользователем отобразится окно со списком таблиц и графиков. Пользователь может выбрать несколько элементов. Для выбора элемента, необходимо кликнуть checkbox левой клавишей мыши. Затем нажать кнопку «Вставить» и элемент добавиться в документ в место текущего положения курсора.
- Заголовки. Ниспадающее меню, предоставляющее выбор видов заголовка. При выборе одного из видов, элемент на котором в данный момент сфокусирован курсор, получит атрибут заголовка соответствующего типа.
- Вставить содержание. При нажатии на кнопку, на экране появится модальное окно, в котором пользователь может выбрать тип содержания (содержание, список таблиц, список графиков, список изображений) и добавить его в документ. Содержание добавиться в место текущего положения курсора.
- Вставить изображение. При нажатии на кнопку, на экране отобразится окно для выбора изображения с устройства. После выбора изображение добавиться в документ в место текущего положения курсора.
- Вставить значение из таблицы / заявки. При нажатии на кнопку перед пользователем отобразится окно с информацией из системы. Каждый элемент, вставка которого реализована в редакторе, при наведении курсора выделяется

подчеркнутой линией. Для выбора элемента, необходимо кликнуть на него левой клавишей мыши. Затем нажать кнопку «Вставить» и элемент добавиться в документ в место текущего положения курсора.

– Удалить маркер. Если в содержимое шаблона содержит элементы для обновления, помеченные определенными атрибутами, то пользователь не может их изменить вручную. Такой элемент можно только удалить. Кнопка становится активной, только при наведении курсора на маркера т.е. на элемент имеющий атрибут для автоматического обновления.

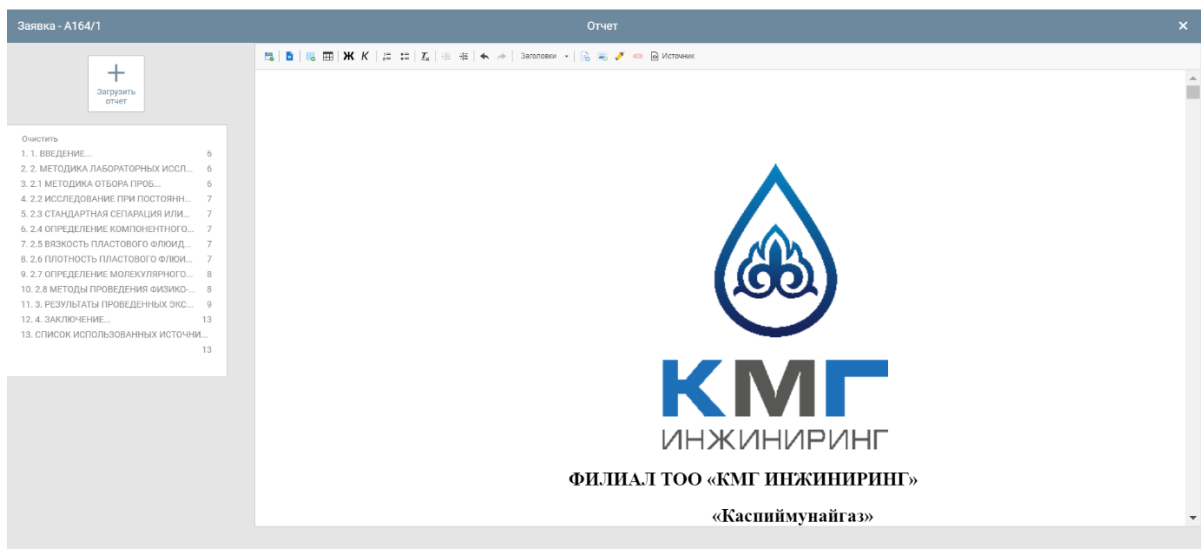


Рисунок-3.3.1 – Пользовательский интерфейс редактора

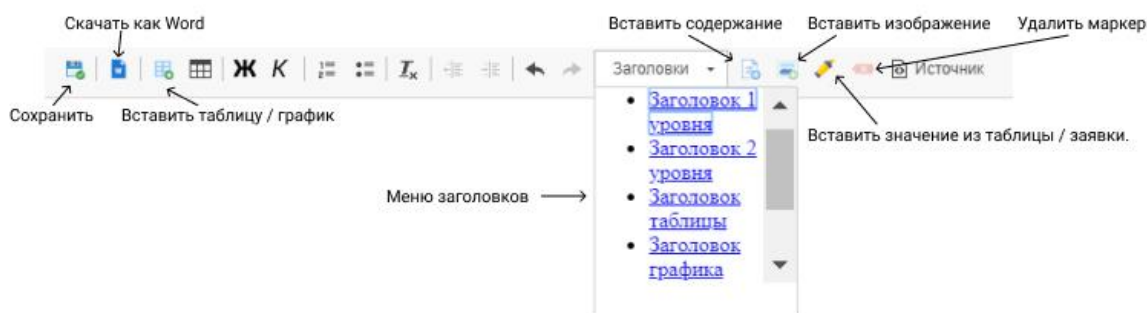


Рисунок-3.3.2 – Панель управления редактора

Модальном окне «Вставка объекта» для реализации функции добавления таблиц или графиков, содержит два вертикальный аккордеона, разделяющих таблицы и графики. При нажатии на любой из аккордеонов, мы увидим внизу список соответствующих данных, в виде таких же аккордеонов. Для выбора элемента для добавления в документ, пользователь должен кликнуть по checkbox слева.

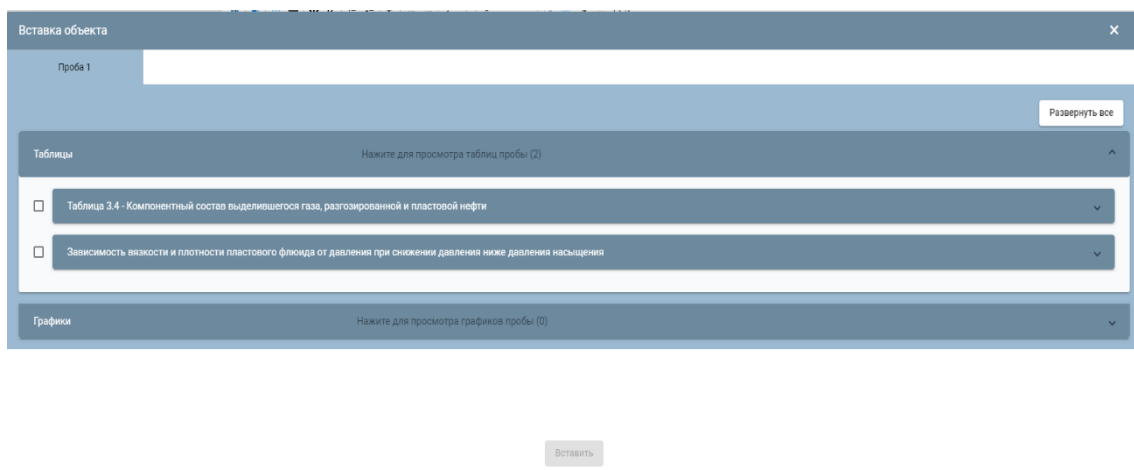


Рисунок-3.3.3 – Модальное окно для реализации вставки таблицы / графика

При клике на сам аккордеон с названием элемента, внизу появится изображение элемента.

№ п/п	Компоненты	Выделившийся газ, мол.%	Разгазированная нефть, мол.%	Пластовая нефть, мол.%
1	1	2	3	4
2	Среднетяжелый			
3	Углеводородный газ			
4	Азот			
5	Метан			
6	Этан			
7	Пропан			
8	Изобутан			
9	Нормальный бутан			

Рисунок-3.3.4 – Модальное окно для реализации вставки таблицы / графика, при нажатии на название таблицы

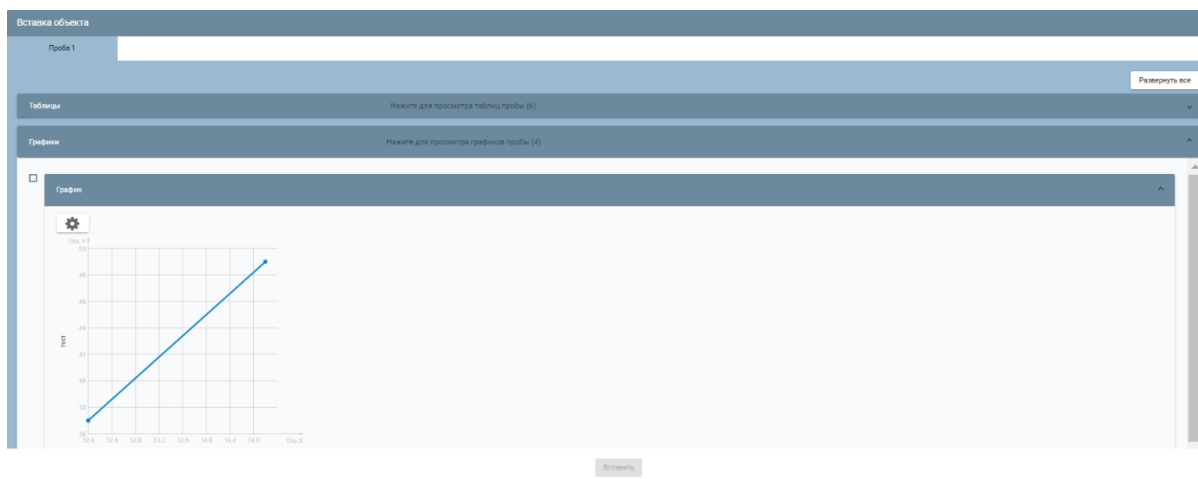


Рисунок-3.3.5 – Модальное окно для реализации вставки таблицы / графика, при нажатии на название графика

Модальное окно «Добавить содержание» состоит из 3 вкладок:

- Заголовки. Это заголовки, отмеченные атрибутами 1 и 2 уровней;
- Заголовки таблиц;
- Заголовки графиков;

Каждая вкладка содержит конкретный список заголовков. Нумерация заголовков происходит в зависимости от их расположения в документе.

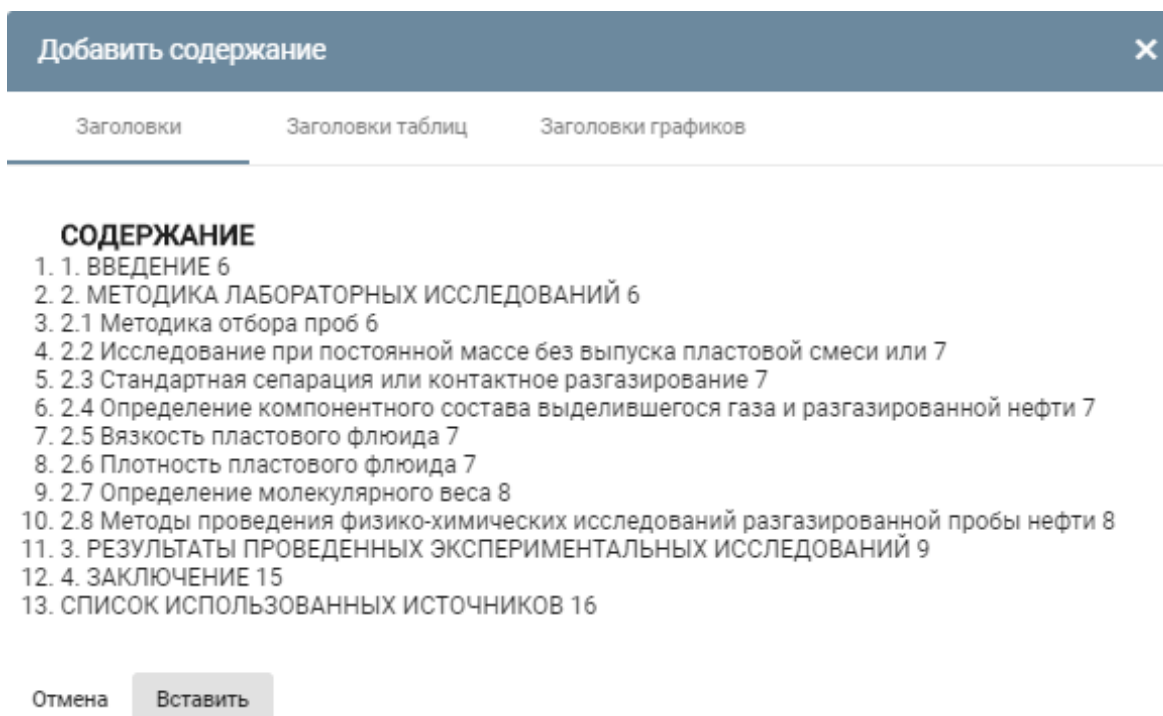


Рисунок-3.3.6 – Модальное окно для реализации вставки содержания

Модальное окно для выбора изображения, является стандартным системным окном.

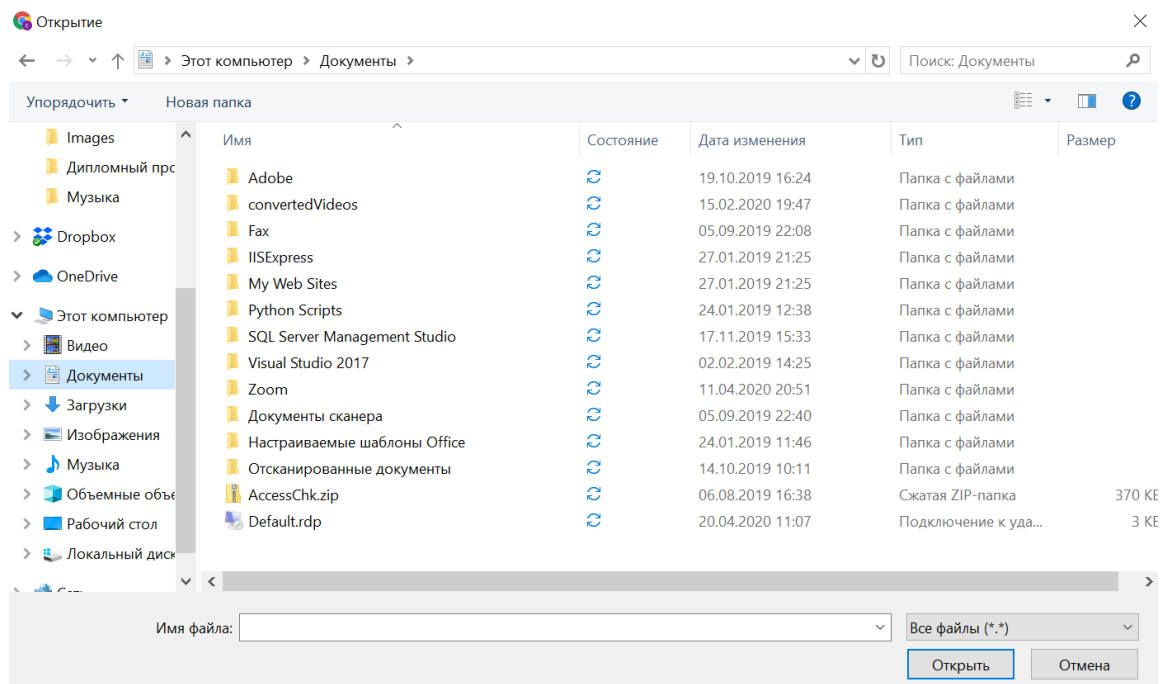


Рисунок-3.3.7 – Модальное окно для реализации вставки изображения

Модальное окно «Вставка данных» открывается, при нажатии на кнопку «Вставить значение из таблицы / заявки». Окно содержит ниспадающие меню, первое для выбора таблицы или заявки, второе для выбора конкретной таблицы или заявки. При выборе таблицы, она отобразится в модальном окне. Пользователь может выбрать одно, любое значение, которое подсвечивается серым фоном.

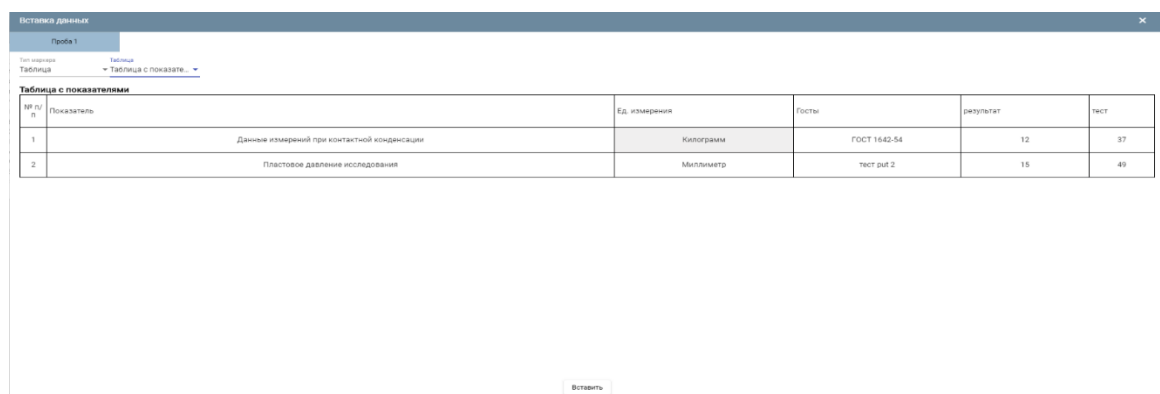


Рисунок-3.3.8 – Модальное окно для реализации вставки значения из таблицы

При выборе заявки, в модальном окне появятся данные из заявки. Пользователь может выбрать одно, любое значение.

Заказчик	Данные
АО «Эмбаунайгаз»	
БИН	123456789123
№ договора	55555
Дата планового завершения	30.04.2020 г.
Местонахождение	С. Нурманлы
№ скважины	00000
Количество проб	1
Тип пробоотборника	
Интервал перфорации	-
Глубина отбора	
Рекомендованная НЕТ	
Температура (°C)	0
Давление (МПа)	0
Дата отбора	20.04.2020 г.
Дата поступления	20.04.2020 г.

ВСТАВИТЬ

Рисунок-3.3.9 – Модальное окно для реализации вставки значения из заявки

4 Экспериментальный раздел

4.1 Интеграция CkEditro4 и фреймворка Angular

CKEditor 4 предлагает встроенную интеграцию через Angular компонент CKEditor 4. Он обеспечивает глубокую интеграцию CKEditor 4 и Angular, что позволяет использовать встроенные функции редактора WYSIWYG внутри компонента Angular. Компонент CKEditor 4 Angular совместим с версиями Angular 5.0 и более поздними.

Изучая методы интеграции CkEditor4 и Angular, в менеджере пакетов npm, была найдена библиотека ng2-ckeditor.

npm - крупнейший в мире реестр программного обеспечения. Разработчики с открытым исходным кодом со всех континентов используют npm для обмена и заимствования пакетов, а многие организации также используют npm для управления частной разработкой.

Авторы библиотеки ng2-ckeditor реализовали возможность добавления новых кнопок на панель инструментов. В дальнейшем весь функционал по нажатию кнопки может быть полностью разработан с использованием фреймворка Angular.

Для начала работы следует выполнить следующее:

- Добавление JavaScript файлов в приложение.

```
<script src="https://cdn.ckeditor.com/4.10.1/full/ckeditor.js"></script>
```

- Скачивание и установка библиотеки

JSPM : jspm install npm:ng2-ckeditor

NPM : npm install ng2-ckeditor

YARN: yarn add ng2-ckeditor

- Добавление CKEditorModule в основной модуль:

```
import { CKEditorModule } from 'ng2-ckeditor';
```

```
import { FormsModule } from '@angular/forms';
```

```
@NgModule({  
  // ...  
  imports: [CKEditorModule, FormsModule],  
  // ...  
})  
export class AppModule {}
```

После установки можно начать процесс разработки. Для использования CkEditor4, нужно добавить к Angular компоненту следующий код:

```
import { Component } from '@angular/core';
```

```
@Component({  
  selector: 'sample',  
  template: `
```

```

<ckeditor
  [(ngModel)]="ckeditorContent"
  [config]="{uiColor: '#99000'}"
  [readonly]="false"
  (change)="onChange($event)"
  (editorChange)="onEditorChange($event)" <!-- CKEditor change event -->
  (ready)="onReady($event)"
  (focus)="onFocus($event)"
  (blur)="onBlur($event)"
  (contentDom)="onContentDom($event)"
  (fileUploadRequest)="onFileUploadRequest($event)"
  (fileUploadResponse)="onFileUploadResponse($event)"
  (paste)="onPaste($event)"
  (drop)="onDrop($event)"
  debounce="500">
</ckeditor>
,
})
export class Sample {
  ckeditorContent: string = '<p>Some html</p>';
}

```

Теперь CkEditor4 можно использовать в качестве компонента Angular. Входные параметры компонента:

- конфигурация config: объект конфигурации для CKEditor см. документацию CkEditor4;
- debounce: есть возможность добавить задержку (мс) при обновлении ngModel;
- readonly: включено / отключено. Редактор работает только в режиме чтения;

Есть возможность использовать следующие директивы для добавления пользовательских кнопок на панель инструментов CKEditor и организации их в группы.

```

<ckeditor
  [(ngModel)]="ckeditorContent">
  <ckbutton [name]="saveButton"
    [command]="saveCmd"
    (click)="save($event)"
    [icon]="save.png"
    [label]="Save Document"
    [toolbar]="clipboard,1">
  </ckbutton>
</ckeditor>

```

Следует обратить внимание, что атрибуты `name` и `command` являются обязательными.

4.2 Метод обновления элементов

4.2.1 Атрибуты маркеров

Если в документе нужно обновить одно конкретное значение, то возникает проблема нахождения искомого элемента. Для обновления только определенных элементов в документе, они должны обладать некими уникальными свойствами. Для решения этой проблемы было решено добавлять к таким элементам различные атрибуты, которые будут хранить информацию для обновления значения. Все элементы требующие обновления получили название маркеры. Всего существует 4 типа маркеров: заголовки таблиц (`table-header`), таблицы (`table`), графики (`graph`), значения из заявки и пробы (`objectVariable`). Маркеры нельзя изменить вручную, можно только удалить, поэтому у них присутствует атрибут `readonly = "true"`. Каждый вид маркера также имеет свои собственные атрибуты.

Перечислим элементы и их атрибуты:

Заголовок таблицы:

- `data-marker-table` – ID таблицы
- `data-marker-table-templateid` – ID шаблона таблицы
- `data-marker-type="table-header"`.
- `class="marker"`

Таблица:

- `data-marker-table` – ID таблицы
- `data-marker-table-templateid` – ID шаблона таблицы
- `readonly="true"`
- `data-marker-type="table"`

График

- `class="marker"`
- `data-marker-graph` – ID графика
- `data-marker-type="graph"`

Значение из таблицы

- `data-marker-type="objectVariable"`
- `class="marker"`
- `readonly="true"`
- `data-marker-info` - содержит все данные о значении. (`markerType`: `tableCell` - указывает тип маркера, `tableId` – ID таблицы, `selectedTableRowId` – ID строки, в которой было выбрано значение,

selectedTableColumnId – ID колонки из которой было выбрано значение,
tableTemplateId – ID шаблона таблицы)

Значение из заявки

- data-marker-type="objectVariable"
- class="marker"
- readonly="true"
- data-marker-info - содержит все данные о значении.
(markerType:appVariable - указывает тип маркера , applicationId - ID заявки, applicationVariableName: CompanyTitle - название значения)

Значение из пробы

- data-marker-type="objectVariable"
- class="marker"
- readonly="true"
- data-marker-info – содержит данные о значении (markerType: appVariable указывает тип маркера, applicationId – ID заявки, sampleId – ID пробы, sampleVariableName: HoleNumber – название значения)

Заголовки

- header-type="level1" – указывает тип заголовка

4.2.2 Получение HTML компонентов

При добавлении элементов, путем использования панели инструментов, пользователь выбирает определенный элемент, редактору же для корректного отображения, передается HTML элемента.

```
clickOnElement(selectedElement: HTMLBaseElement) {  
    let control = selectedElement.getAttribute('formcontrolname');  
    let fullDate;  
    let variableName;  
    if (control === 'PlanEndDate') {  
        fullDate  
        =  
this.getConvertFullDate(this.applicationForm.controls[control].value);  
    } else if (control === 'PerforationIntervals') {  
        variableName = 'SamplePerforationIntervals';  
    }  
    this.controlName = control;  
    this.selElementWithAttributes.emit({  
        el: selectedElement, value: control === 'PlanEndDate' ? fullDate :  
this.applicationForm.controls[control].value,  
        attributes: { markerType: 'appVariable', applicationId: this.data.ID,  
applicationVariableName: variableName ? variableName : control }  
    });  
}
```

```
}
```

В этом случае все просто, в открытых модальных окнах выстраивается HTML компонентов и в последствии его нужно просто вставить в редактор. Но при обновлении документа возникают трудности.

Пользователь должен получить актуальную версию документа при первом же открытии. Когда пользователь открывает редактор с документом у нас есть все данные для обновления маркеров, кроме их HTML. Чтобы решить изложенную выше задачу было принято решение отображать компоненты за пределами видимости пользователя.

Angular отрисовывает компоненты в одном модальном окне с редактором, но с внесенными изменениями в стилях CSS:

```
:host {  
  position: absolute;  
  top: -22000px;  
  left: -22000px;  
  opacity: 0;  
}
```

ЗАКЛЮЧЕНИЕ

В рамках данного дипломного проекта поставленная цель выполнена успешно. Результатом работы является динамический редактор документов, реализованный с использованием фреймворка Angular версии 8 и WYSIWYG редактора SckEditor4. Данный инструмент в разы сокращает время работы с документами и освобождает пользователя от постоянного контроля актуальности данных. Разработанный инструмент реализован сугубо для работы внутри системы LIMS и не является свободно распространяемым ресурсом.

В процессе исследования предметной области была подробно изучена лабораторная система управления информацией. Было определено, что разработанный в рамках этой работы инструмент является важным инструментом для поддержания точной и своевременной отчетности в системе LIMS.

В процессе проектирования сформировалась архитектура системы, также был построен подробный алгоритм работы динамического редактора. Был проведен сравнительный анализ различных технологий, для выбора наиболее подходящей проекту.

Динамический редактор документов представляет законченный инструмент, который является частью уже работающей системы LIMS. Редактор является инструментом, который может легко дорабатываться в соответствии с новыми требованиями, что в свою очередь, может стать причиной его быстрого развития.

Программный код хранится в приватном удаленном репозитории tfs. При разработке активно использовалась система контроля версий – git.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Introduction to Angular concepts // Электронная версия на сайте <https://angular.io/guide/architecture>
2. Laboratory information management system // Электронная версия на сайте https://en.wikipedia.org/wiki/Laboratory_information_management_system
3. What is a LIMS? // Электронная версия на сайте <https://www.autoscribeinformatics.com/resources/blog/what-is-a-lims>
4. Base concepts Editor.js // Электронная версия на сайте <https://editorjs.io/base-concepts>
5. CKEditor 4 Documentation // Электронная версия на сайте <https://ckeditor.com/docs/ckeditor4/latest/index.html>
6. Integrated development environment // Электронная версия на сайте https://en.wikipedia.org/wiki/Integrated_development_environment

Приложение А

(обязательное)

Техническое задание

А1.1 Техническое задание на разработку динамического редактора документов

Настоящее техническое задание распространяется на разработку динамического редактора документов, отвечающего за отчетность в системе управления лабораторией LIMS. Предполагается что использовать данную систему будут только сотрудники компании. Данный инструмент позволит оптимизировать процесс работы с документами.

А1.1.1 Основания для разработки

Редактор разрабатывается на основании устного распоряжения Заместителя Председателя Правления АО «Институт цифровой техники и технологий» по разработке программных продуктов.

А1.1.2 Назначение

Разрабатываемый инструмент предназначен для удовлетворения внутренних потребностей компании заказчика и оптимизации работы с документами.

А1.1.3 Требования к функциональным характеристикам

Динамический редактор должен обеспечить возможность выполнения следующих функций и задач:

- стандартной работы с фрагментами текста документа (копирование, удаление, перемещение, корректировка текста в режимах замены или вставки символов и т.д.);
- автоматической загрузки шаблона документа;
- блокировки документа при завершении исследования;
- отрисовки Angular компонентов в редакторе;

Продолжение приложения А

- добавления в документ необходимых данных из системы;
- добавления содержания, списков таблиц, графиков и изображений;
- автоматического добавления новых элементов, если они появляются в системе и должны присутствовать в отчете;
- навигации по тексту;

А1.1.4 Требования к надежности

Предусмотреть запрет на редактирование обновляемых данных. Предусмотреть блокировку документа при завершении исследования в заявке. После блокирования заявки, в документе доступен режим только для чтения.