

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных
систем

Нафикова Лиана Ирековна

Разработка интерфейса для валидатора выпускных квалификационных работ

Учебная практика

Научный руководитель:
ассистент кафедры ИАС Чернышев Г. А.

Санкт-Петербург
2023

Оглавление

Введение	3
1. Постановка задачи	4
2. Обзор текущего приложения	5
3. Обзор используемых инструментов	6
3.1. React	6
3.2. Mantine UI	7
4. Реализация	8
4.1. Дизайн	8
4.2. Описание работы с приложением	9
4.3. Описание реализации	12
Заключение	15
Список литературы	16

Введение

Приложение Mundane Assignment Police представляет собой инструмент, разработанный специально для проверки работ студентов на наличие ошибок оформления. МАР обеспечивает оценку академических работ, помогая студентам повысить качество своих ВКР. Приложение сканирует текст работы и подчёркивает проблемные участки. Таким образом, студенты получают незамедлительную обратную связь и имеют возможность улучшить текст.

Важной составляющей приложения, обрабатывающего данные от пользователя, является интерфейс взаимодействия с ним. Приложение должно давать пользователю возможность сделать то, что он не может сделать иным способом, а если и может, то предоставлять способ сделать это более удобно, обеспечивать неопытного пользователя доступными инструкциями [1]. Интерфейсом могут быть различные формы, меню, поля для ввода и загрузки данных. Без интерфейса пользователь не сможет настроить программу, а программа корректно обработать введённые данные.

Пользовательский интерфейс МАР на данный момент предоставляет лишь базовый функционал отображения текста и ошибок оформления в работах студентов. При этом в приложение постоянно добавляются новые правила проверки текста, для каждой работы также теперь есть возможность предоставить статистику наиболее часто встречающихся слов и размеров секций. Однако клиентская часть приложения оставалась неизменной и не давала возможности отобразить новый функционал. Именно поэтому в этом семестре было решено заниматься обновлением интерфейса приложения.

1. Постановка задачи

Целью данной работы является разработка нового интерфейса приложения Mundane Assignment Police. Для её достижения были поставлены следующие задачи:

1. выполнить обзор текущей версии приложения с целью выявить недочёты;
2. провести обзор технологий, изучить и выбрать лучшие для использования;
3. создать и реализовать макет интерфейса.

2. Обзор текущего приложения

В этой секции будет произведён обзор текущей версии приложения.

На главной странице описаны основная функциональность и некоторые из поддерживаемых на данный момент правил без описания или комментариев. С помощью нажатия на кнопку можно начать загрузку файлов (рис. 1).

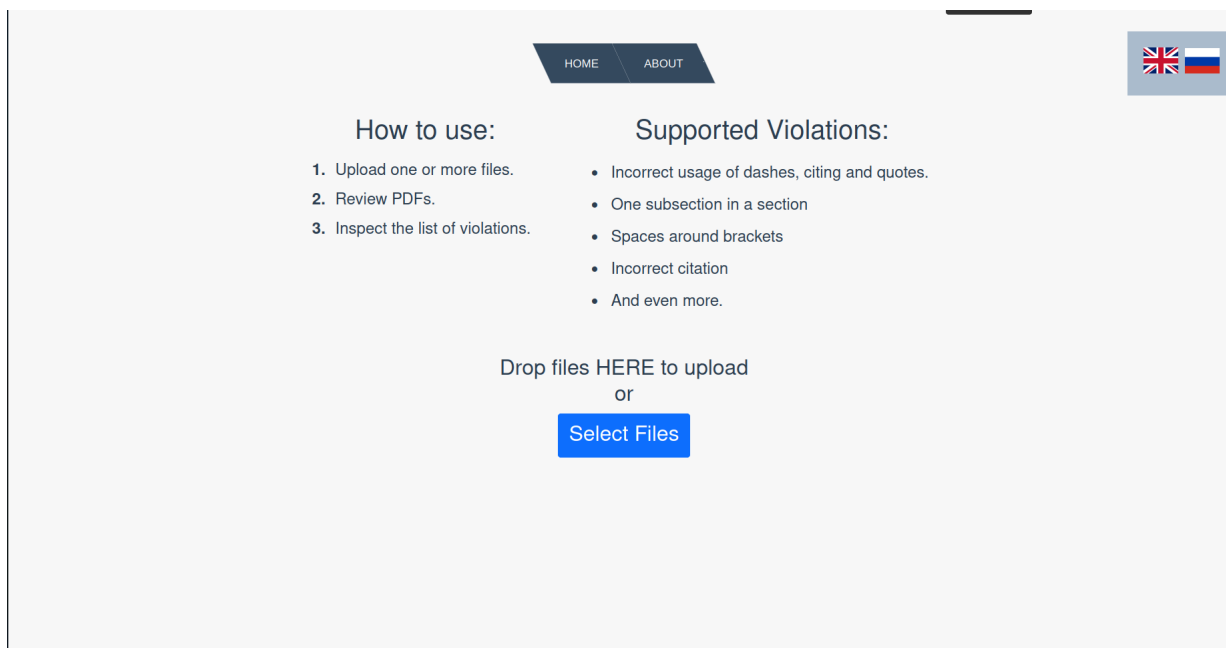


Рис. 1: Главная страница текущей версии приложения

На странице обработки (рис. 2) нарушения не сгруппированы, также присутствует большой размер страницы относительно размера экрана. То есть, в случае если находится слишком много однотипных нарушений, то переключаться между ними становится неудобно. Нет возможности отключить те или иные правила, хотя некоторые из них могут требовать достаточно много времени на проверку или быть нерелевантными для конкретной работы. Очевидным недостатком является отсутствие единого стилистического решения.

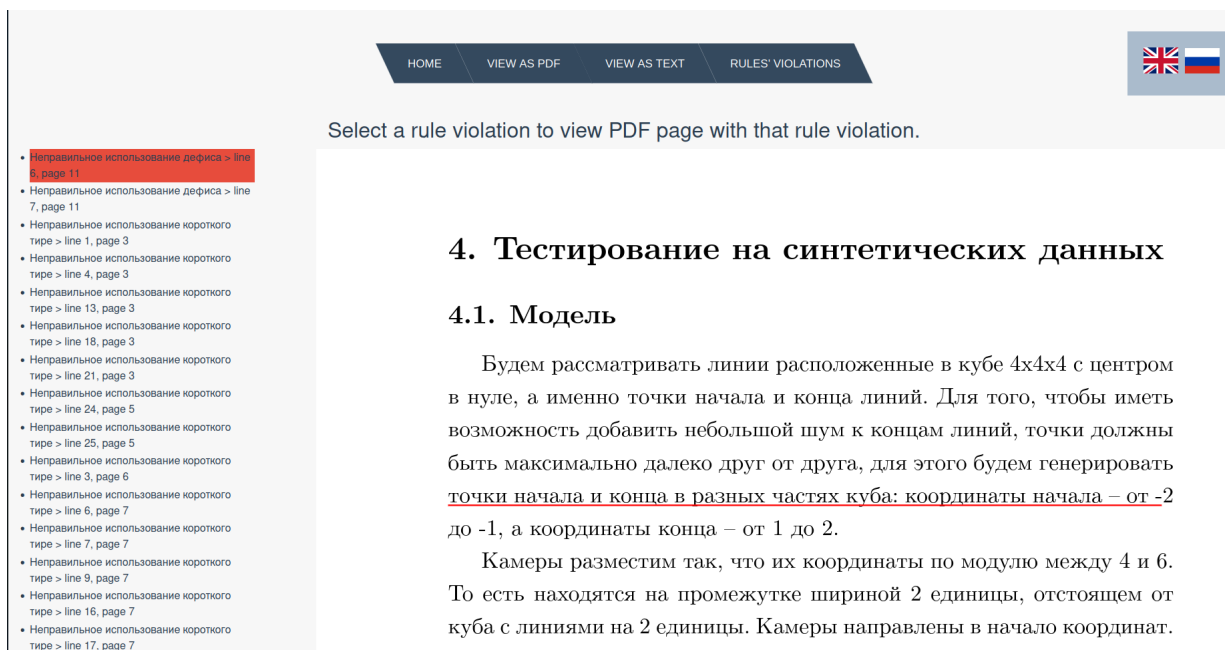


Рис. 2: Страница представления результатов обработки файла

3. Обзор используемых инструментов

В этой части рассматриваются основные технологии, используемые для реализации интерфейса приложения.

3.1. React

Первоначально клиентская часть приложения была реализована с помощью Vue.js [2]. Этот выбор обосновывался тем, что для небольших приложений Vue.js показывал лучшую производительность чем другие популярные фреймворки, такие как React [3] и Angular [4]. С того момента размер приложения увеличился, поэтому согласно [5] теперь целесообразнее использовать React. Также Vue и Angular пользуются меньшей популярностью, чем React [6, 7], что значит, что найти нужную информацию для React будет намного проще.

React — библиотека для создания пользовательских интерфейсов, которая работает по принципу компонентного подхода. Создаваемые компоненты вкладываются друг в друга и переиспользуются для формирования сложных интерфейсов. React упрощает взаимодействие с DOM-деревом, так

как сам следит за обновлением интерфейса, позволяя сосредоточиться разработчику на работе с данными. Реализован такой механизм взаимодействия с помощью построения дерева элементов: когда в каких-то узлах этого дерева произошли изменения, строится новое дерево и сравнивается с предыдущим, после чего происходит стадия отрисовки, и на странице появляются новые данные.

3.2. Mantine UI

Для реализации графической составляющей приложения была выбрана UI-библиотека с открытым исходным кодом Mantine [8]. Она проста в освоении, часто обновляется, предоставляет множество готовых компонентов, используемых для реализации веб-приложений. Также библиотека предоставляет удобную работу с формами и React Hooks.

4. Реализация

Основная функциональность проекта — отображение ошибок оформления в выпускных и курсовых работах студентов. Для удобства пользователя необходимо реализовать возможность удобной загрузки документов, окно обработки отдельного документа, которая будет предоставлять статистику работы и показывать ошибки в работе. Также необходима возможность фильтрации по нарушению, по странице. Преимуществом будет также создание и добавление конфигураций правил. Каждому правилу необходимо описание.

4.1. Дизайн

Перед началом разработки приложения была продумана его структура. Работа началась с создания эскиза пользовательского интерфейса в приложении Figma [9]. Проект доступен по ссылке¹.

Это помогло определить количество страниц и расположение элементов на них. Было решено, что для правильной работы сервиса потребуются следующие страницы:

- Главная страница;
- Страница с контактами;
- Страница, предоставляющая настройку правил и добавление новых конфигураций правил;
- Страница для загрузки файлов;
- Страница с результатами обработки (нарушениями, статистикой) отдельного документа.

¹<https://www.figma.com/file/M0UFSp9f1FS3hyet7PljEr/Untitled?type=design&node-id=29%3A2633&mode=design&t=X00f2aVt58WuN9ad-1> — репозиторий проекта в Figma (дата обращения: 29.08.2023)

4.2. Описание работы с приложением

Пользователь переходит на сайт. Открывается главная страница (рис. 3) приложения. На ней он видит описание Mundane Assignment Police, список поддерживаемых правил и кнопку с предложением начать работу, которая переводит его на страницу с настройками правил.

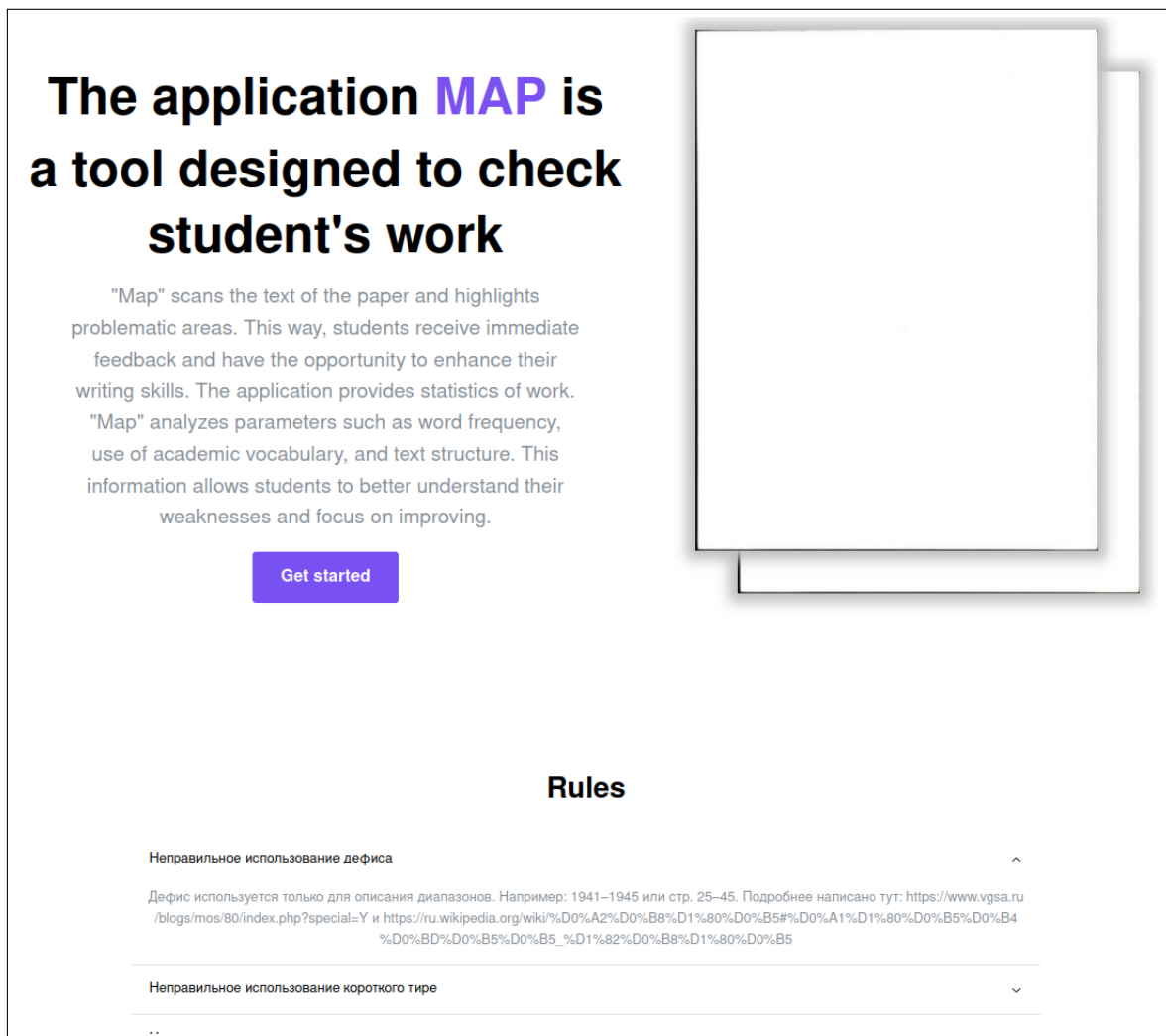


Рис. 3: Главная страница приложения

Здесь (рис. 4) пользователь может выбрать те конфигурации, которые уже присутствуют, либо составить свою собственную с помощью заполнения формы.

После того, как выбрана конфигурация, пользователь переходит на страницу загрузки файлов. Она имеет схожую с предыдущей страницей структуру, только здесь вместо конфигураций располагаются загруженные фай-

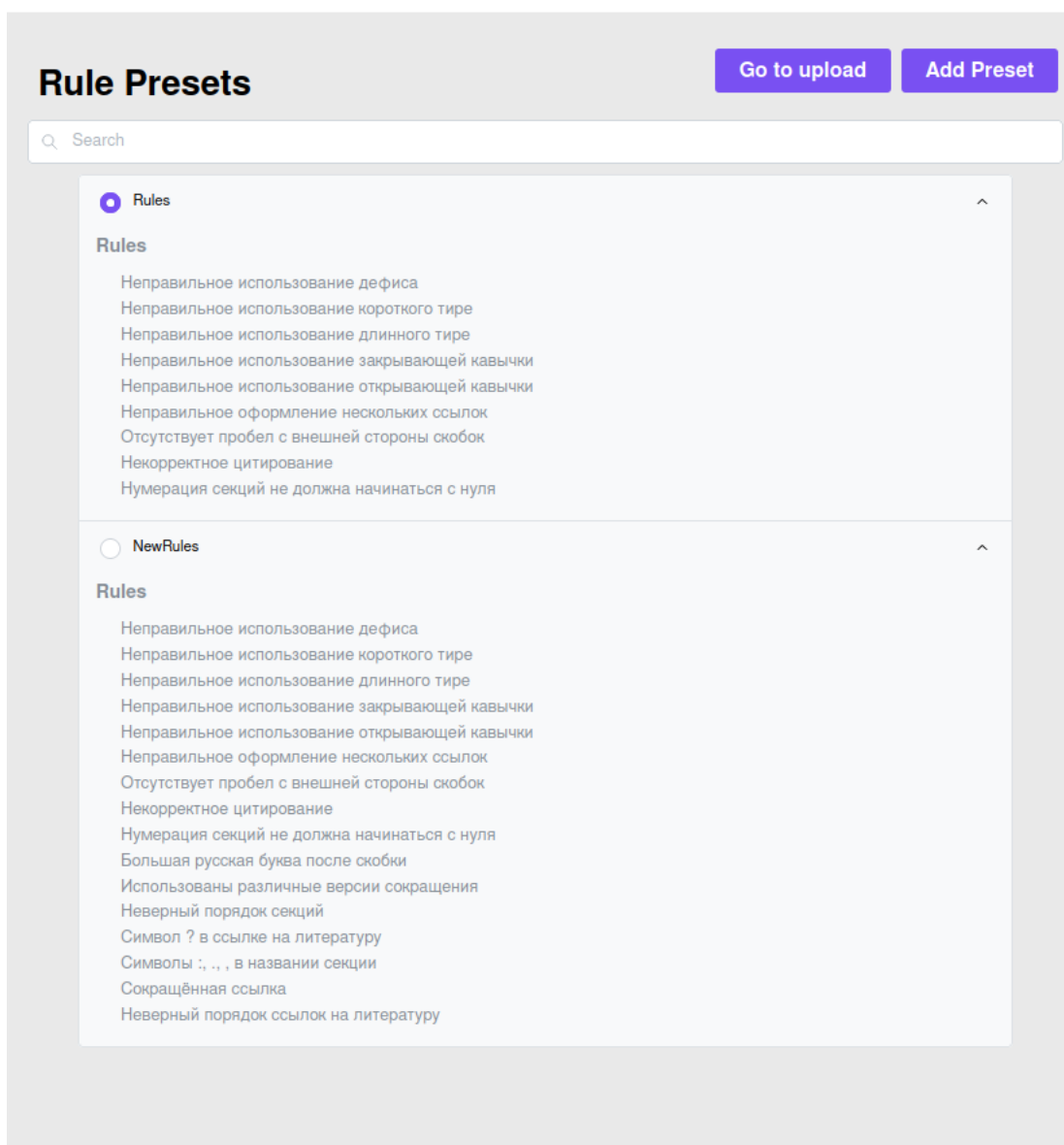


Рис. 4: Страница с настройками конфигураций правил

лы. После загрузки каждый файл обрабатывается с учётом выбранной конфигурации, и теперь пользователь может просмотреть ошибки, найденные в его работе, путём перехода по кнопке.

На странице обработки файлов (рис. 5) в левой части расположены все подключенные в конфигурации правила, в центральной — страница с текущим выбранным нарушением, а в правой — все найденные в работе нарушения, которые можно разбить по группам, в зависимости от страницы или названия правила. В левой части также можно выбрать, отображать правило из конфигурации или нет. На этой же странице можно посмотреть

статистику работы.

The screenshot displays a web application for processing files, specifically focusing on document violations. The interface is divided into three main sections: a left sidebar for rules, a central document viewer, and a right sidebar for violations.

Left Sidebar (Rules): Contains a list of rules with checkboxes for enabling/disabling them. The rules are categorized by severity: ERROR (red), WARNING (yellow), and INFO (blue). The visible rules are:

- ERROR:** ☒ Неправильное оформление нескольких ссылок
- ERROR:** ☐ Отсутствует пробел с внешней стороны скобок
- WARNING:** ☒ Большая русская буква после скобки
- ERROR:** ☒ Неправильное написание целых чисел от 1 до 9

Central Document Viewer: Displays a document with the following content:

3. Реализованные правила

В данном разделе описывается процесс реализации правил заявленных в постановке задачи.

3.1. Оформление ссылок на разные статьи

Нередко люди пишущие ВКР или другие работы используя несколько ссылок оформляют это так:

В рассматриваемых играх [3] [6] присутствуют элементы RPG.

Однако корректно было бы оформить это так:

В рассматриваемых играх [3, 6] присутствуют элементы RPG.

Для реализации этого правила было использовано правило проверяющее соседние символы. В нем задаются два символа которые не могут стоять рядом и затем все соседние символы проверяются. Первым символом был выбран символ "[", после чего проверяются только его правые соседи на то, равны ли они "]". При этом символы запятой и пробела отмечаются как игнорируемые для проверки.

3.2. Выход символа за поля

Часто при оформлении ВКР и других работ по невнимательности, или ввиду иной ошибки может получиться так, что один или несколько символов выйдут за поля. Для фиксации подобных случаев будут использоваться координаты символов.

Однако проверять координату каждого символа будет излишне, поэтому проверку начинать следует с конца строки, и при выявлении вы-

Right Sidebar (Violations): Shows a list of violations found in the document. It includes a "Sort" dropdown set to "page" and a list of violations with expand/collapse icons:

- page 6: ☐ Неправильное оформление нескольких ссылок
- page 9: ☐ Большая русская буква после скобки
- page 9: ☐ Большая русская буква после скобки
- page 9: ☐ Неправильное написание целых чисел от 1 до 9
- page 9: ☐ Неправильное написание целых чисел от 1 до 9
- page 9: ☐ Неправильное написание целых чисел от 1 до 9

Рис. 5: Страница обработки файла

Диаграмма переходов между страницами представлена на рис. 6.

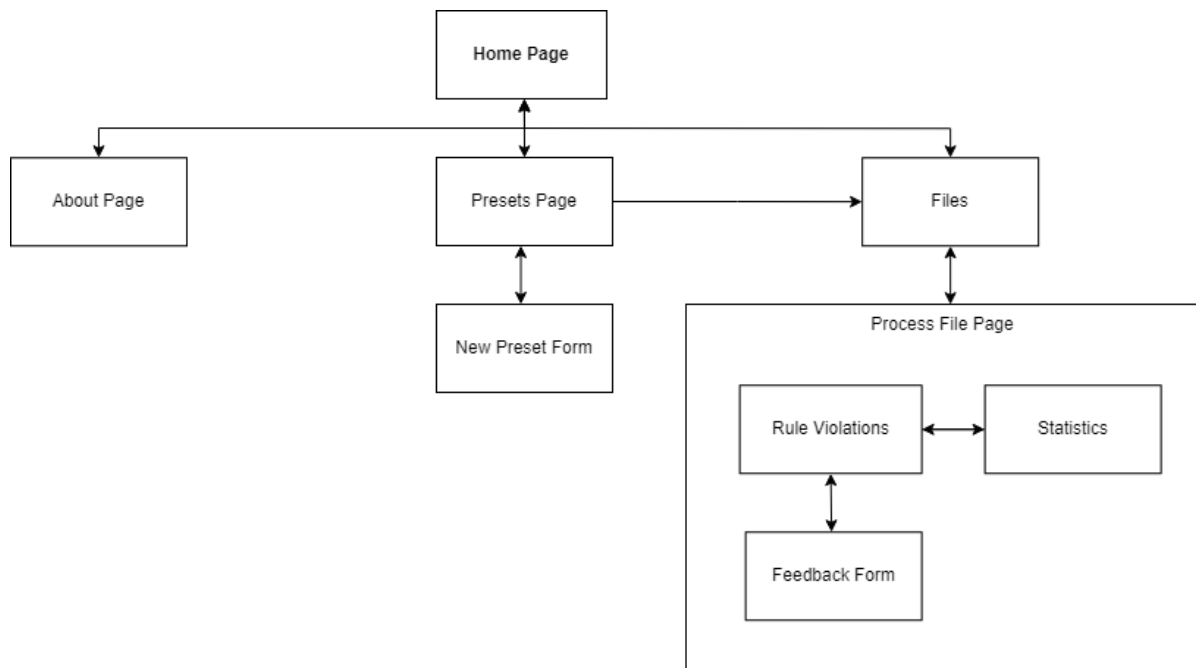


Рис. 6: Диаграмма переходов

4.3. Описание реализации

Для реализации макета интерфейса было необходимо реализовать множество компонентов для каждой страницы. Были реализованы как компоненты, которые лишь отображают данные (карточка с контактами, компоненты главной страницы), так и компоненты, которые осуществляют поиск среди файлов, загрузку, выполняют сортировки и фильтрацию элементов. Также некоторые компоненты было необходимо связать друг с другом для корректного отображения информации. Например, на странице, отображающей ошибки в работе, предоставлена возможность выбрать отображение нарушения лишь определённых правил. Связь компонентов происходит с помощью использования встроенных в React функций (React Hooks), которые следят за состоянием одного компонента, и при его изменении производят необходимые вычисления для изменения и перерисовки другого компонента.

Некоторые части интерфейса отличаются лишь частью вложенной информации, поэтому они не требуют отдельных реализаций. Например, страницы конфигураций правил и загрузки файлов имеют схожую структуру, поэтому был создан один компонент, принимающий в качестве входных па-

раметров отличающиеся элементы, которые необходимо вставить на страницу (кнопки, название секции, список элементов, которые нужно отображать). Также общий компонент модального окна был реализован для формы добавления новой конфигурации правил и формы обратной связи. Список реализованных компонентов:

1. Компоненты-контейнеры страниц сайта:

- (a) Home (главная страница);
- (b) About (страница с контактами);
- (c) Download (страница загрузки файлов);
- (d) Presets (страница выбора конфигурации правил);
- (e) ProcessFile (страница обработки файла);

2. Компоненты навигации:

- (a) Menu для возможности переключения между страницами;
- (b) RuleViolationsHeader для возможности переключения между отображениями нарушений и статистики;

3. RuleModal для отображения модальных окон с формами;

4. Формы:

- (a) RuleForm — форма настройки конфигурации правила;
- (b) FeedbackForm — форма обратной связи по нарушению;

5. ListContainer для отображения списков файлов на странице загрузки и списков конфигураций на странице выбора конфигурации;

6. Списки:

- (a) PresetList — список конфигураций;
- (b) FileList — список файлов;
- (c) RuleList — список правил на странице с нарушениями;

- (d) RuleViolationList — список найденных нарушений на странице с нарушениями;
7. Элементы списков:
 - (a) Preset — конфигурация, элемент списка конфигураций;
 - (b) File — файл, элемент списка файлов;
 - (c) Rule — правило, элемент списка правил;
 - (d) RuleViolation — нарушение, элемент списка нарушений;
 8. RuleInput для поискового ввода на страницах конфигураций и загрузки файлов;
 9. RuleViolationsContent для отображения ошибок оформления в работе на странице обработки файла;
 10. Statistics для отображения статистики работы на странице обработки файла;
 11. FileView для отображения конкретной страницы файла с нарушением;
 12. RuleSelect для разделения нарушений на группы по названию, странице.

Заключение

В результате данной работы был разработан новый интерфейс приложения, помогающего студентам проверять свои работы на наличие ошибок оформления. Были выполнены следующие задачи:

1. выполнен обзор текущей версии приложения с целью выявить недочёты;
2. проведён обзор технологий, изучены и выбраны лучшие для использования;
3. создан и реализован макет интерфейса.

Открытый код проекта доступен по ссылке на репозиторий² GitHub.

²<https://github.com/gwyndollne/map-react-front> — репозиторий проекта на GitHub (дата обращения: 28.10.2023).

Список литературы

- [1] Russ Unger and Carolyn Chandler. *A Project Guide To Ux Design: For User Experience Designers In The Field Or In The Making*. Pearson Education, 2009.
- [2] Documentation for vue.js. <https://ru.vuejs.org/index.html>. Accessed: 29.09.2023.
- [3] Documentation for react. <https://react.dev/learn>. Accessed: 29.09.2023.
- [4] Documentation for angular. <https://angular.io/>. Accessed: 26.10.2023.
- [5] Mattias Levlin. Dom benchmark comparison of the front-end javascript frameworks react. *Angular, Vue, and Svelte, Åbo Akademi*, 2020.
- [6] Stackoverflowtrends for react, angular, vue. <https://insights.stackoverflow.com/trends?tags=angular%2Creactjs%2Cvue.js>. Accessed: 26.10.2023.
- [7] npm trends for react, angular, vue. <https://npmtrends.com/angular-vs-react-vs-vue>. Accessed: 26.10.2023.
- [8] Documentation for mantine. <https://mantine.dev/>. Accessed: 29.09.2023.
- [9] Онлайн-сервис для разработки интерфейсов и прототипирования. <https://www.figma.com/>. Accessed: 29.09.2023.