УДК 004.62

**Разработка информационной системы поддержки для управления выпускными работами студента**

***Сибгатуллин М.Р.***

*(ФГБОУ ВПО Казанский национальный исследовательский технический   
университет им. А.Н. Туполева – КАИ)*

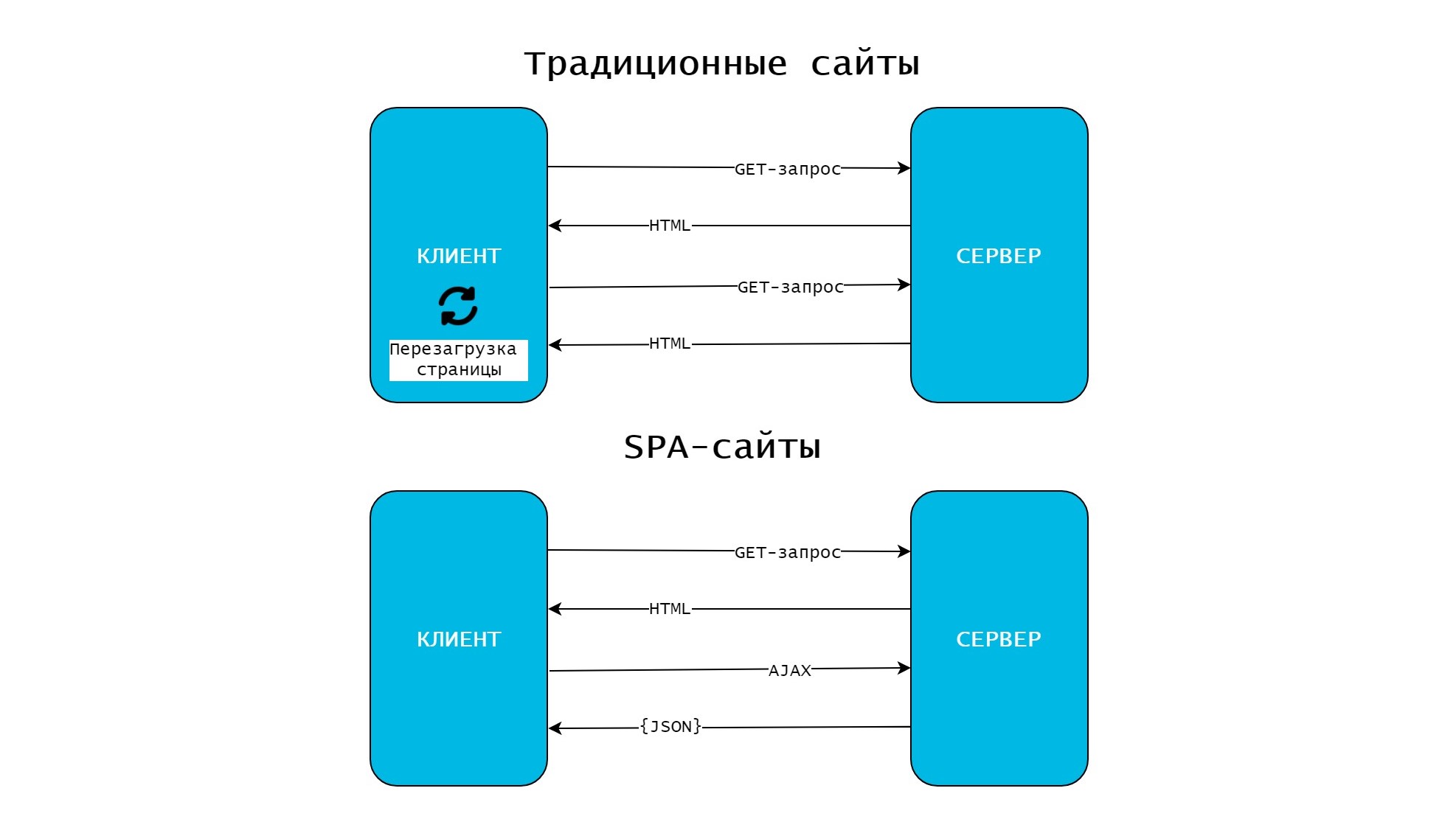
**Аннотация**

Данный доклад посвящен разработке информационного системы поддержки для комфортного управления выпускными работами. Целью работы является создание легко масштабируемой системы, способной хранить и отображать выпускные квалификационные работы. Разработка велась на языках JavaScript, HTML и CSS, так же использовался фреймворк написанный нами на Php7.

**Организация работы одностраничного приложения**

SPA (Single Page Application) — одностраничное JavaScript приложение, которое запускается и работает в браузере. В отличии от «традиционного» сайта, архитектура на SPA-сайтах построена так, что рендеринг страницы полностью происходит на стороне клиента, а не на стороне сервера.

В браузере пользователя запускается JavaScript-приложение, а все необходимое содержимое страниц динамически загружается с помощью AJAX. Навигация по сайту происходит без перезагрузки страниц. За счет такой архитектуры, SPA-сайты работают быстрее, чем «традиционные» сайты.



Рассмотрим детальнее, как происходит загрузка и рендеринг содержимого на SPA-сайтах:

1. Пользователь запрашивает HTML содержимое сайта.

2. В ответ приходит JavaScript-приложение.

3. Приложение определяет, на какой странице находится пользователь, и какое содержимое ему нужно отобразить.

4. С помощью AJAX пользователь получает необходимый контент: CSS, JS, HTML и текстовый контент.

5. JavaScript-приложение обрабатывает полученные данные и отображает их в браузере.

6. При навигации по сайту обновляется не вся страница, а только необходимое содержимое.

2.1.1 Серверная часть и фреймворк Laravel

На серверной стороне необходимо:

1. База данных.

2. Маршрутизация.

3. Хранить домашнюю страницу с файлами стилей и файлом клиентского js приложения.

4. При первом заходе отправлять домашнюю страницу.

5. Работать с базой данных и отвечать на API запросы от клиентской части.

6. Отправлять сообщения в web сокет.

7. Отправлять запрос на push уведомление сервису уведомлений.

Для этих задач был выбран фреймворк написанный нами. Laravel — бесплатный веб-фреймворк с открытым кодом, предназначенный для разработки с использованием архитектурной модели MVC. Он был выбран, потому что имеет удобные инструменты для работы с соответствующими задачами:

1. Миграции —система контроля версий для вашей базы данных. Они позволяют вашей команде изменять структуру БД, в то же время оставаясь в курсе изменений других участников. Миграции обычно идут рука об руку с построителем структур для более простого обращения с архитектурой вашей базы данных. Если вы когда-нибудь просили коллегу вручную добавить столбец в его локальную БД, значит вы сталкивались с проблемой, которую решают миграции БД.

2. Все маршруты (routes) Laravel определены в файлах маршрутов, которые расположены в каталоге routes. Эти файлы автоматически загружаются фреймворком. В файле routes/web.php определены маршруты для вашего web-интерфейса. Эти маршруты входят в группу посредников web, которые обеспечивают такие возможности, как состояние сессии и CSRF-защита. Маршруты из файла routes/api.php не поддерживают состояния и входят в группу посредников api.

3. Views (шаблоны) обычно содержат HTML-код вашего приложения и представляют собой удобный способ разделения бизнес-логики и логики отображения информации. Шаблоны хранятся в папке.

4. Гибкая система маршрутов позволяет без конфликтов использовать HTML5 history API.

5. ORM Eloquent - красивая и простая реализация паттерна ActiveRecord для работы с базой данных. Каждая таблица имеет соответствующий класс-модель, который используется для работы с этой таблицей. Модели позволяют читать данные из таблиц и записывать данные в таблицу. Создавая API, понадобится промежуточный слой между вашими моделями и JSON-ответами, которые возвращаются пользователям. API Resources в Laravel 5.5 позволяют быстро и легко трансформировать модели и коллекции моделей в желаемое JSON-представление.

6. Laravel broadcast позволяет отправлять события, на которые подписывается клиентское приложение. Позволяет создавать защищенные приватные и публичные каналы.

7. Guzzle – php HTTP клиент позволяющий просто отправлять любые запросы внешним интернет сервисам.

2.1.2 Клиентская часть и фреймворк Vue.js

На стороне клиента необходимо:

1. Хранить логику всего клиентского приложения.

2. Выводить шаблоны с оформлением.

3. Заполнять шаблоны содержимым.

4. Отправлять запросы на сервер.

5. Маршрутизация.

6. Хранение локальных данных.

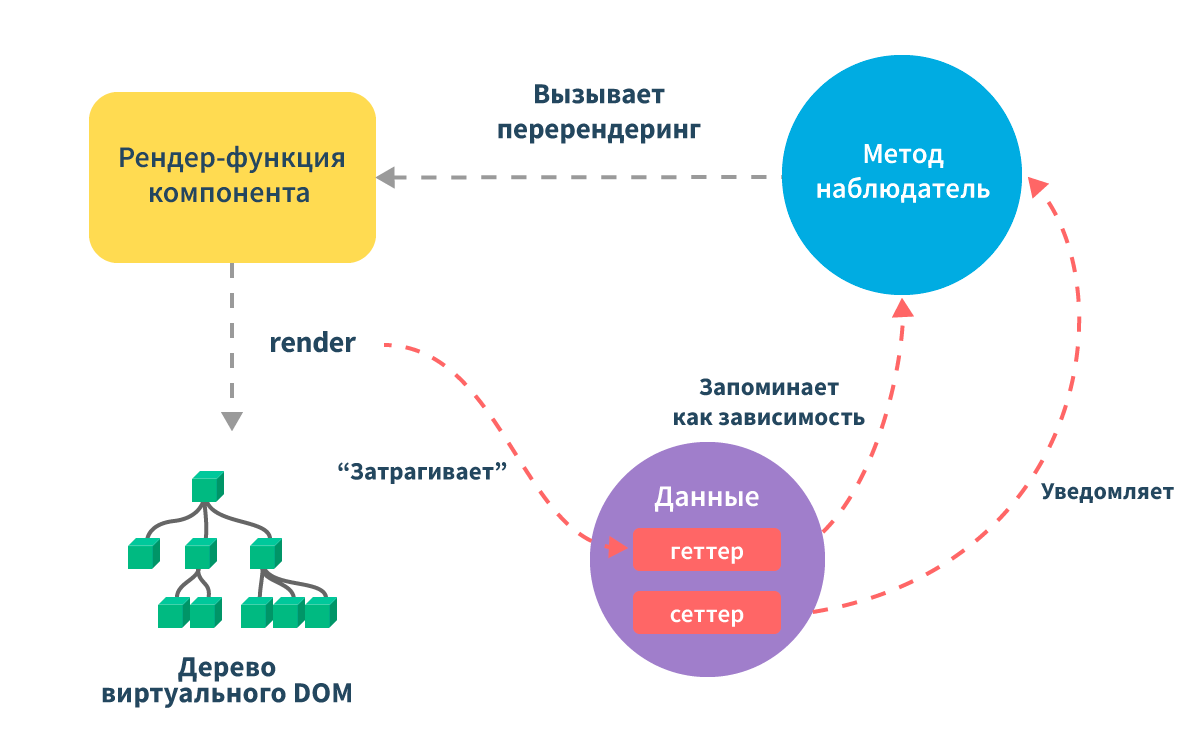
7. Подключаться к веб сокету и обрабатывать его события.

Для этого я выбрал Vue.js фреймворк. Vue.js — JavaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов. Легко интегрируется в проекты с использованием других JavaScript-библиотек. Может функционировать как веб-фреймворк для разработки одностраничных приложений в реактивном стиле. И позволяет решить соответствующие поставленные задачи:

1. Система сборки позволяет создавать цельное приложение из компонентов.

2. В современной разработке UI вместо разделения кодовой базы на три огромных слоя, что тесно переплетаются друг с другом, имеет больше смысла делить их на слабо связанные компоненты и компоновать уже их. Внутри компонента, его шаблон, логика и стили неразрывно связаны между собой, что позволяет сделать компонент более сплочённым и удобным в поддержке.

3. Система реактивности позволяет связывать значения элементов в DOM с переменными JS. Которое реализуется следующим образом.



4. Axios, основанный на промисах http клиента js позволяет асинхронно проводить запросы и обрабатывать полученные ответы. Автоматически преобразовывать данные в JSON и поддерживает защиту от CSRF.

5. Vue Router — официальная библиотека маршрутизации для Vue.js. Она глубоко интегрируется с ядром Vue.js, что позволяет создавать SPA-приложения с лёгкостью.

Включает следующие возможности:

• Вложенные маршруты/представления

• Модульная конфигурация маршрутизатора

• Доступ к параметрам маршрута, query, wildcards

• Анимация переходов представлений на основе Vue.js

• Удобный контроль навигации

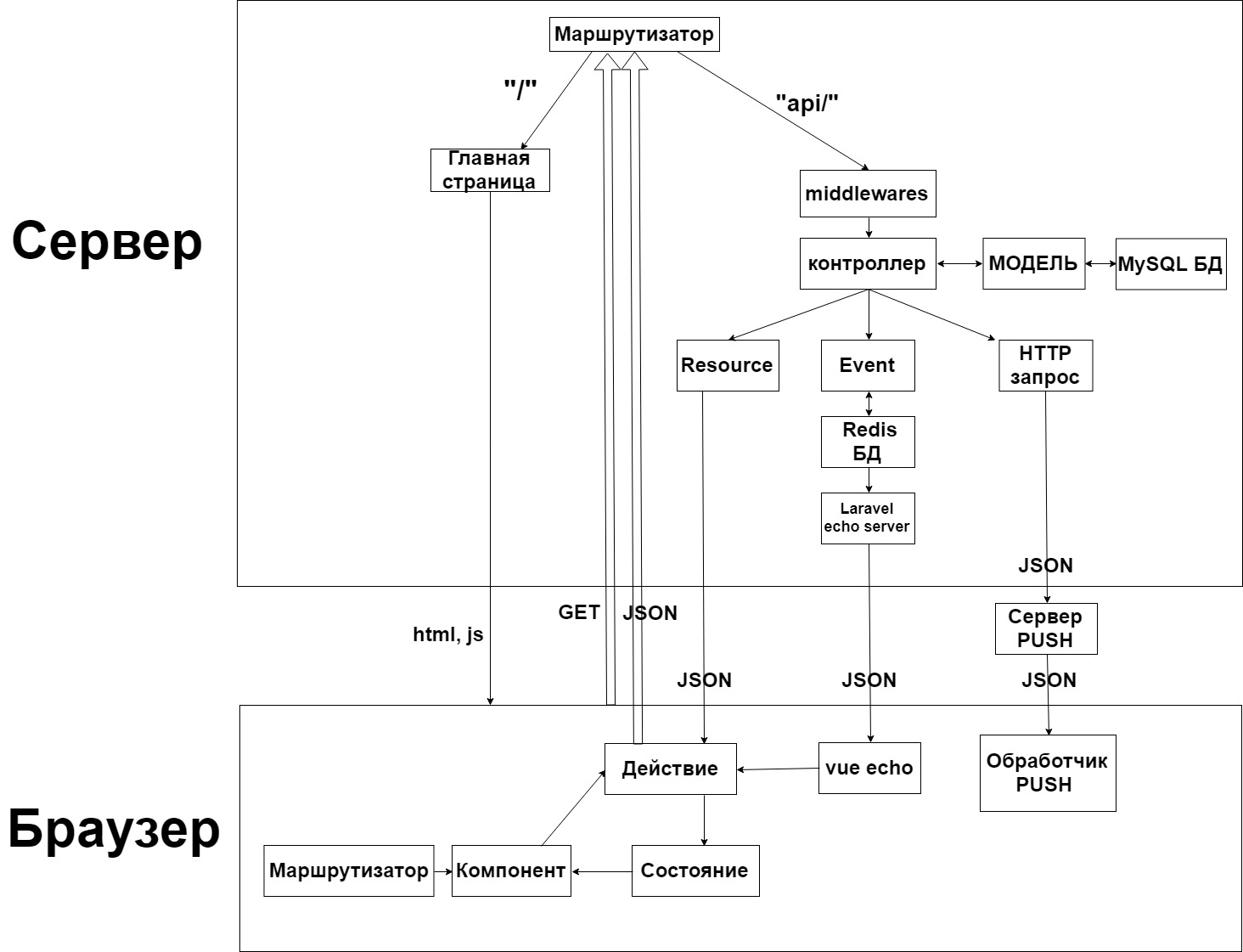
• Автоматическое проставление активного класса CSS для ссылок

• Режимы работы HTML5 history или хэш, с авто-переключением в IE9

• Настраиваемое поведение прокрутки страницы

6. Vuex — паттерн управления состоянием + библиотека для приложений на Vue.js. Он служит централизованным хранилищем данных для всех компонентов приложения с правилами, гарантирующими, что состояние может быть изменено только предсказуемым образом.

7. С помощью инструмента Laravel Echo легко использовать мощь WebSockets в своих Laravel-приложениях. Он упрощает самые необходимые и самые трудные аспекты построения сложных взаимодействий WebSockets. Echo состоит из двух частей: набора улучшений для системы вещания сообщений Laravel (Event broadcasting system), и нового пакета JavaScript.



**Литература**

1. *PHP 7. В подлиннике Игорь Симдянов, Дмитрий Котеров* 2016.
2. Laravel документация <https://laravel.com/>
3. Vue.js документация <https://vuejs.org/>
4. Алан Бьюли «Изучаем SQL» (2007)