

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**СОЗДАНИЕ WEB-ПРИЛОЖЕНИЯ ДЛЯ СОСТАВЛЕНИЯ
РАСПИСАНИЯ ПЕРЕСДАЧ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 551 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Сендерова Олега Александровича

Научный руководитель

зав. каф., к. ф.-м. н., доцент

С. В. Миронов

Заведующий кафедрой

к. ф.-м. н., доцент

С. В. Миронов

Саратов 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Задача составления расписания передач	4
1.1 Комплексные системы планирования ресурсов предприятия для образовательных учреждений	4
1.2 Зарубежные системы управления учебным процессом.....	4
1.3 Решения на основе табличных процессов.....	5
1.4 Результаты проведенного анализа существующих решений	5
1.5 Выбор технологий для реализации	6
1.6 Подробное описание выбранных технологий	8
1.6.1 Java.....	8
1.6.2 Библиотека Spring Framework	8
1.6.3 СУБД PostgreSQL	9
1.6.4 Vue.js.....	10
2 Создание приложения	12
2.1 Структура базы данных	12
2.2 Реализация серверной части приложения.....	12
2.3 Реализация клиентской части приложения	12
2.4 Администрирование системы	13
ЗАКЛЮЧЕНИЕ	14

ВВЕДЕНИЕ

Образовательный процесс в высших учебных заведениях требует современного подхода, в том числе это касается составления расписания занятий, экзаменов и пересдач.

Нынешние подходы к составлению подобных расписаний имеют ряд недостатков, таких как высокая нагрузка на административный отдел, непонятное распределение по времени и аудиториям как для преподавателей так и для студентов.

Целью настоящей работы является создание WEB-приложения для составления расписания пересдач, что в перспективе позволит снизить нагрузку на административный отдел, минимизировать количество допускаемых ошибок при составлении расписания, повысить удобство для всех участников образовательного процесса.

Задачи, которые предстоит решить в процессе реализации:

- проанализировать существующие решения, которые позволяют составлять расписания для учебных заведений;
- выбрать инструменты для разработки WEB-приложения;
- изучить технологии для разработки WEB-приложения;
- создать WEB-приложение с использованием выбранных технологий.

1 Задача составления расписания пересдач

Основная задача данной работы — реализация WEB-приложения для составления расписания пересдач.

В рамках поставленной задачи необходимо выделить основные бизнес-сущности приложения и их свойства, построить диаграмму взаимодействия этих сущностей, на основе выделенных взаимодействий определить структуру базы данных. Реализовать механизмы создания, удаления и редактирования пересдач, учесть расписание каждой доступной аудитории, настроить механизмы валидации создаваемых пересдач, так чтобы один преподаватель не мог иметь несколько записей на одно и то же время, студенты не могли иметь больше одной пересдачи в день. Дополнительно необходимо настроить безопасность приложения, выделив роли, которые будут присваиваться пользователю, в зависимости от которых пользователь будет иметь тот или иной набор прав и доступов.

1.1 Комплексные системы планирования ресурсов предприятия для образовательных учреждений

Современные системы управления учебным процессом, такие как ИАС «АВЕРС» и «1С:Университет», предлагают комплексные решения для автоматизации деятельности вузов. Данные платформы обеспечивают полный цикл управления образовательным процессом — от приема студентов до выпуска. Однако применительно к задаче организации пересдач эти системы демонстрируют ряд существенных ограничений.

Основные проблемы подобных систем:

- избыточная сложность конфигурации;
- недостаточная гибкость алгоритмов составления расписания;
- высокая стоимость внедрения и сопровождения;
- высокий порог входа для необученного персонала;

Особенно критичным является отсутствие учета специфики пересдач, что приводит к необходимости постоянной ручной корректировки автоматически сгенерированного расписания.

1.2 Зарубежные системы управления учебным процессом

Международные платформы (ExamSoft, Syllabus+) предлагают решения для автоматизации экзаменационных процессов. Анализ функциональных воз-

возможностей позволяет выделить следующие особенности:

- интеллектуальные алгоритмы распределения ресурсов;
- интеграция с системами календарного планирования;
- мобильный интерфейс;

Однако практическое применение этих решений в российских вузах имеет ряд ограничений:

- несоответствие нормативным требованиям российского образования;
- отсутствие локализации;
- проблемы интеграции с отечественными информационными системами;

1.3 Решения на основе табличных процессов

Широкое распространение в образовательных учреждениях получили табличные решения Google Sheet и Microsoft Excel. Их популярность обусловлена следующими факторами:

- минимальные требования к инфраструктуре;
- нулевая стоимость внедрения;
- низкий порог входа для персонала;

Однако проведенный анализ выявил ряд существенных недостатков:

- отсутствие механизма контроля целостности данных;
- отсутствие возможности реализации сложных бизнес-правил;
- высокий риск ошибки при работе с данными;

1.4 Результаты проведенного анализа существующих решений

На основе проведенного анализа у существующих решений были обнаружены следующие проблемы:

1. Отсутствие специализированных инструментов:

- недостаточное внимание к пересдачам;
- ограниченная возможность адаптации;

2. Экономические барьеры:

- высокая стоимость комплексных систем;
- расходы на поддержку;
- отсутствие окупаемости;

3. Технические ограничения:

- проблемы интеграции;
- сложность масштабирования;

— недостаточная надежность;

Таким образом, проведенный анализ подтверждает необходимость создания решения для составления расписания передач, которое устранит выявленные недостатки существующих систем.

1.5 Выбор технологий для реализации

Процесс создания WEB-приложения состоит из трех ключевых этапов:

- реализация пользовательского интерфейса;
- внедрение интерактивных элементов;
- интеграция с серверной частью приложения и СУБД;

Основой WEB-страниц служит HTML (HyperText Markup Language) — стандартный язык разметки, определяющий структуру и содержание страницы (текстовые блоки, медиа-элементы, формы ввода). Для стилизации HTML-элементов применяется CSS (Cascading Style Sheets), который контролирует визуальное представление: цветовую гамму, размеры, позиционирование и другие аспекты оформления.

Альтернативой чистому CSS могут выступать препроцессоры типа SCSS, предлагающие дополнительные возможности синтаксиса (вложенность, переменные, миксины), однако конечный результат все равно компилируется в стандартный CSS.

Для создания интерактивных компонентов используется JavaScript — основной язык программирования для frontend-разработки. Современная практика предполагает применение фреймворков (React, Vue.js, Angular), которые упрощают создание сложных динамических интерфейсов за счет компонентного подхода и готовых решений для управления состоянием приложения.

Финальный этап предполагает организацию взаимодействия между клиентской частью и сервером через API-запросы. Это обеспечивает:

- обмен данными с backend-системой;
- работу с базой данных;
- реализацию бизнес-логики приложения;

Такая архитектура позволяет создать полнофункциональное веб-приложение, где frontend отвечает за пользовательский интерфейс, а backend — за обработку данных и выполнение серверных операций.

В качестве основных требований к системе можно выделить три отдельные группы. Функциональные требования, которые определяют то как должна

вести себя система. Нефункциональные требования, которые должны определять ограничения накладываемые на систему. Эксплуатационные требования определяют характеристики, которые проявляются в процессе его функционирования.

1. Функциональные требования:

- обработка бизнес-логики планирования;
- поддержка множества одновременных пользователей;
- генерация отчетов в различных форматах;
- интеграция отчета с внешними системами;

2. Нефункциональные требования:

- надежность и отказоустойчивость;
- безопасность данных;
- масштабируемость;

3. Эксплуатационные требования:

- простота развертывания;
- легкость поддержки;
- возможность дальнейшего развития;

Среди основных критериев оценки выбираемых технологий следует выделить открытость исходного кода, безопасность, масштабируемость, простота разработки, открытая документация, объектно-ориентированная модель. Отталкиваясь от выделенных критериев и выявленных требований к приложению, можно выделить следующий список технологий, которые будут использоваться:

- Java SE(Standart Edition) обеспечивает открытость исходного кода, покрывает потребность объектно-ориентированной модели, а так же имеют большую, открытую документацию;
- Spring Framework предоставляет механизмы для удобного развертывания приложения, его разработки, а так же обеспечения безопасности;
- PostgreSQL система управления базами данных с открытым исходных кодом, которая имеет огромный список базового функционала покрывающего основные потребности в работе с данными;
- Vue.js библиотека языка программирования JavaScript, которая предоставляет удобные и простые механизмы для реализации пользовательского интерфейса;

1.6 Подробное описание выбранных технологий

1.6.1 Java

Java — это высокоуровневый объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems в начале 1990-х годов. Основной целью создания Java было предоставить язык, который был бы простым, гибким и удобным для разработки приложений, работающих на различных платформах.

Преимущества Java SE(Standart Edition):

- независимость от платформы: Java известен своей философией «написано один раз, работает везде», что означает, что код Java может выполняться на любом устройстве, где установлена виртуальная машина Java, независимо от базовой операционной системы;
- Java — объектно-ориентированный язык;
- Java имеет механизмы управления памятью (Garbage Collector);
- большая библиотека классов;
- строгая типизация;
- многопоточность;
- обратная совместимость;
- встроенные механизмы безопасности;

Области применения языка:

- Android разработка;
- Разработка динамических WEB-приложений, таких как онлайн-банкинг;
- Разработка программных средств (IntelliJ IDEA);
- Разработка трейдинговых приложений;

1.6.2 Библиотека Spring Framework

Spring Framework — это инструмент для разработки Java-приложений, который значительно упрощает создание масштабируемых, надежных и легко поддерживаемых приложений. Благодаря своей гибкости и модульности позволяет использовать только необходимые компоненты, что делает код более читаемым и простым в поддержке.

Еще одним значительным преимуществом является внедрение зависимостей. Благодаря чему компоненты приложения становятся слабо связанными, что значительно упрощает работу с кодом. Кроме того Spring обеспечивает

поддержку интеграций с различными базами данных.

Spring также предоставляет мощные инструменты для работы с безопасностью, что позволяет сосредоточиться на бизнес-логике, а не на инфраструктурных деталях. В дополнение развитая экосистема в частности модуль Spring Boot, значительно ускоряет процесс разработки, конфигурации и развертывания приложений.

Основные модули Spring:

- Spring Core — основной модуль, обеспечивающий механизм внедрения зависимостей и управления жизненным циклом объектов;
- Spring MVC — модуль для создания WEB-приложений, который реализует паттерн Model-View-Controller;
- Spring Data — упрощает работу с базами данных, предлагая абстракции для доступа к данным;
- Spring Security — модуль для реализации безопасности приложений, включая аутентификацию и авторизацию;
- Spring AOP — предоставляет возможности аспектно-ориентированного программирования, позволяя отделить кросс-функциональные задачи от основной бизнес-логики;
- Spring Boot — облегчает настройку и развертывание приложений, позволяя быстро создавать проекты с минимальной конфигурацией;
- Spring Cloud — набор инструментов для разработки облачных приложений и микросервисов, предлагающий решения для управления конфигурацией, сервисами и маршрутизацией;

1.6.3 СУБД PostgreSQL

PostgreSQL — объектно-реляционная система управления базами данных с открытым исходным кодом.

Основные задачи, которые решаются с помощью PostgreSQL:

- гибкий доступ к данным;
- управление данными — создание, редактирование, удаление, обновление;
- транзакционная отправка последовательных запросов;
- защита информации от потерь и утечек;
- контроль версий и организаций одновременного доступа к базе из разных источников;

Преимущества:

1. Объектно-реляционная модель. Данные в модели представляются в виде объектов их атрибутов, методов.
2. Поддержка множества типов данных. PostgreSQL поддерживает как классические типы данных, например, целые числа, числа с плавающей точкой, булевы значения, строки, так и поддержка денежных типов, многомерных массивов данных, битовых строк, а так же поддержка JSON, XML.
3. Работа с большими объемами данных. В PostgreSQL отсутствуют ограничения на объем хранимых данных.
4. Поддержка сложных запросов.
5. Одновременная модификация базы. В PostgreSQL используется технологии MVCC — Multiversion Concurrency Control, многоверсионное управление параллельным доступом. Эта технология позволяет вносить изменения в базу только после фиксации транзакции.
6. Соответствие ACID. ACID — это набор принципов для обеспечения целостности данных:
 - Atomicity — атомарность — гарантирует, что каждая транзакция будет выполнена полностью или не будет выполнено совсем.
 - Consistency — согласованность — транзакция не допускает промежуточных результатов.
 - Isolation — изоляция — параллельные транзакции не влияют на выполнение текущей транзакции.
 - Durability — надежность — если было получено подтверждение выполнения транзакции, то изменения не будут отменены из-за сбоя.
7. Открытость. PostgreSQL — это ПО с открытым исходным кодом.
8. Кроссплатформенность.
Недостатки:
 - сложность при настройке;
 - повышенное потребление ресурсов;
 - отсутствие ряда функций;

1.6.4 Vue.js

Vue.js — это фреймворк для разработки сайтов и веб-приложений, написанный на JavaScript. Во фреймворке Vue.js встроена логика кнопок и других элементов интерфейса; их внешний вид; механизмы обмена данными и другие необходимые для интерфейса вещи. Другими словами, в нем уже реализованы

все базовые элементы WEB-приложений.

Области применения фреймворка:

- быстрые сайты;
- сайты с высокой нагрузкой;
- адаптивные интерфейсы;
- разделы личных кабинетов;

Основные особенности:

- реактивность;
- прогрессивность;
- простота;
- быстрая разработка;
- доступность;

2 Создание приложения

2.1 Структура базы данных

Для создания структуры базы данных необходимо выделить основные сущности и обозначить их взаимодействие. В результате анализа предметной области были выделены следующие основные сущности:

- пользователь — сюда входят как студенты, так и преподаватели, и диспетчеры;
- передача — основная сущность приложения, которая будет в себе хранить всю необходимую информацию;
- группа — сущность, которая описывает номерные группы, в которых обучаются студенты;
- права пользователей — здесь будут описаны основные роли, которые будут присваиваться пользователям;
- предметы — описывает предмет, который будет включен в передачу;
- строения, аудитории — описывают где именно будут проходить передачи;

2.2 Реализация серверной части приложения

Для реализации серверной части приложения используется следующий список технологий

- JavaSE(Standard Edition) — как основной язык программирования;
- библиотека Spring Framework — как дополнительная технология для реализации, включающая все необходимые для этого компоненты;
- СУБД PostgreSQL — как основная система для работы с данными;
- система сборки Maven;

2.3 Реализация клиентской части приложения

Клиентская часть приложения реализована с помощью библиотеки Vue.js, для упрощения реализации отдельных частей компонентов интерфейса используется библиотека Vuetify, для реализации запросов на сервер используется библиотека axios.

Для выполнения запросов на сервер был создан экземпляр axiosInstance, в котором была определена конечная точка обращения(порт, хост сервера), а так же установлены необходимые заголовки запроса.

2.4 Администрирование системы

Для того, чтобы организовать работу серверной части приложения, необходимо, чтобы на сервере был установлен язык Java версии 17 или выше, docker и систему сборки Maven. Docker необходим в данном случае для того, чтобы выполнить инициализацию базы данных. В проекте представлен файл, который называется `docker-compose.yml` в этом файле находится конфигурация для базы, а так же необходимое программное обеспечение, которое требуется установить перед запуском.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы, были изучены и применены на практике принципы и методики создания WEB-приложения. Были подробно изучены возможности языков программирования Java, JavaScript, библиотек Spring Framework, Vue.js.

Были изучены представленные на рынке решения, рассмотрены их достоинства и недостатки, выявлена необходимость в разработке собственного решения, которое отвечало бы необходимым нормам, стандартам и разработано приложение, позволяющее составлять расписание пересдач, учитывая расписание студентов и преподавателей, а также реализован справочник аудиторий, в котором представлено расписание каждой представленной аудитории.