

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего  
образования «Самарский национальный исследовательский университет  
имени академика С.П. Королева»  
(Самарский университет)

Институт информатики и кибернетики  
Кафедра программных систем

**ОТЧЕТ ПО ПРАКТИКЕ**

Вид практики: учебная

Тип практики: Научно-исследовательская работа  
(получение первичных навыков научно-исследовательской работы)

Сроки прохождения практики: с 20.06.2025 г. по 03.07.2025 г.  
(в соответствии с календарным учебным графиком)

по направлению подготовки 02.03.02 «Фундаментальная информатика  
и информационные технологии»(уровень бакалавриата)  
направленность (профиль) «Информационные технологии»

Обучающийся группы № 6201-020302D \_\_\_\_\_ Е.А. Элекина

Руководитель практики  
от университета, доцент кафедры  
программных систем, к.т.н., доцент \_\_\_\_\_ Д.А. Попова-Коварцева

Дата сдачи 03.07.2025 г.  
Дата защиты 03.07.2025 г.

Оценка \_\_\_\_\_

Самара 2025

## СОДЕРЖАНИЕ

Задание (я) по практике для выполнения определенных видов работ,.....	3
связанных с будущей профессиональной деятельностью .....	3
(сбор и анализ данных и материалов, проведение исследований) .....	3
Введение .....	6
1      Описание и анализ предметной области.....	7
1.1    Основные понятия и определения.....	7
1.2    Обзор существующих систем-аналогов .....	8
1.2.1 Система «1С:Университет» .....	8
1.2.2 Система «АИС Расписание».....	9
1.2.3 Система «TimeTable» (FET) .....	10
1.2.4 Сравнительный анализ систем-аналогов.....	11
1.2.5 Постановка задачи .....	11
2      Проектирование системы .....	13
2.2    Выбор и обоснование архитектуры системы .....	13
2.3    Выбор комплекса программных средств.....	16
Заключение.....	18
Список использованных источников.....	19
Приложение А Слайды презентационного материала.....	20

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»  
(Самарский университет)

Институт информатики и кибернетики  
Кафедра программных систем

Задание (я) по практике для выполнения определенных видов работ, связанных с будущей профессиональной деятельностью  
(сбор и анализ данных и материалов, проведение исследований)

Обучающейся Элекиной Екатерине Анатольевне группы 6201-020302D  
Направлен на практику приказом по университету от 11.06.2025 г. № 309-ПР в Самарский университет на кафедру программных систем.

Планируемые результаты освоения образовательной программы (компетенции)/индикаторы)	Выполнение определенных видов работ, связанных с будущей профессиональной деятельностью (сбор и анализ данных и материалов, проведение исследований)	Результаты практики (планируемые результаты обучения при прохождении практики)
<p><b>ОПК-1 Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности</b></p> <p><b>ОПК- 1.1</b> Использует основные положения и концепции в области математических и естественных наук, базовые теории коммуникации; знает основную терминологию</p> <p><b>ОПК-1.2</b> Осуществляет первичный сбор и анализ материала, интерпретирует различные математические объекты</p> <p><b>ОПК-1.3</b> Осуществляет первичный сбор и анализ материала, интерпретирует</p>	<p>1) Ознакомиться с концепциями и терминологией по теме научно-исследовательской работы.</p> <p>2) Осуществить поиск материала по теме научно-исследовательской работы в сети Интернет, электронных библиотечных системах и базах данных.</p> <p>3) Ознакомиться со стандартом оформления текстовых учебных документов Самарского университета.</p> <p>4) Закрепить знания материала по стандарту оформления текстовых документов с помощью прохождения теста в системе Moodle.</p> <p>5) Оформить отчет по результатам прохождения практики в строгом</p>	<p>1) Ознакомилась с основными концепциями и терминологией, касающихся разработки мобильного приложения.</p> <p>2) Осуществила поиск материала по теме разработки мобильных приложений с дополненной реальностью в сети Интернет, электронных библиотечных системах и базах данных.</p> <p>3) Ознакомилась со стандартом оформления текстовых учебных документов Самарского университета.</p> <p>4) Закрепила знания материала по стандарту оформления текстовых</p>

различные математические объекты	соответствии со стандартом оформления текстовых учебных документов.	документов с помощью прохождения теста в системе Moodle.
<p><b>ОПК-2 Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности</b></p> <p><b>ОПК-2.1</b> Использует основные положения и концепции в области программирования, архитектуру языков программирования, теорию коммуникации, основную терминологию, знаком с содержанием единого реестра российских программ</p> <p><b>ОПК-2.2</b> Анализирует код на типовых языках программирования, может составлять программы</p> <p><b>ОПК-2.3</b> Применяет опыт решения задач анализа, интеграции различных типов программного обеспечения, анализа типов коммуникаций</p>	<ol style="list-style-type: none"> <li>1) Ознакомиться с рекомендациями по подготовке презентационного материала.</li> <li>2) Подготовить презентацию по теме научно-исследовательской работы в соответствии с общими правилами оформления презентаций.</li> <li>3) Закрепить знания материала по подготовке презентационного материала с помощью прохождения теста в системе Moodle.</li> </ol>	<ol style="list-style-type: none"> <li>1) Ознакомилась с рекомендациями по подготовке презентационного материала.</li> <li>2) Подготовила презентацию по теме научно-исследовательской работы в соответствии с общими правилами оформления презентаций.</li> <li>3) Закрепила знания материала по подготовке презентационного материала с помощью прохождения теста в системе Moodle.</li> </ol>
<p><b>ОПК-3 Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз</b></p>	<ol style="list-style-type: none"> <li>1) Ознакомиться с материалом по построению схем алгоритмов.</li> <li>2) Закрепить знания материала по построению схем алгоритмов с помощью прохождения теста в системе Moodle.</li> </ol>	<ol style="list-style-type: none"> <li>1) Ознакомилась с материалом по построению схем алгоритмов.</li> <li>2) Закрепила знания материала по построению схем алгоритмов с помощью прохождения теста в системе Moodle.</li> </ol>

<p><b>данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям</b></p> <p><b>ОПК-3.1</b> Понимает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей</p> <p><b>ОПК-3.2</b> Соотносит знания в области программирования, интерпретирует прочитанное, определяет и создает информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем</p> <p><b>ОПК-3.3</b> Имеет практический опыт применения разработки программного обеспечения</p>		
---	--	--

Срок представления на кафедру отчета о практике 03.07.2025 г.

Руководитель практики от университета,  
доцент кафедры программных  
систем, к.т.н., доцент

\_\_\_\_\_ Д.А. Попова-Коварцева  
(подпись)

Задание принял к исполнению  
обучающийся группы № 6201-020302D

\_\_\_\_\_ Е.А. Элекина  
(подпись)

## ВВЕДЕНИЕ

Современные цифровые технологии активно трансформируют сферу образования, открывая новые возможности для оптимизации учебного процесса. Особенно актуальна эта тенденция для высших и средних специальных учебных заведений, где составление расписания остается одной из наиболее сложных и трудоемких административных задач. В условиях роста численности студентов, увеличения количества образовательных программ и ужесточения требований к качеству организации учебного процесса, традиционные методы ручного планирования становятся неэффективными.

Актуальность разработки специализированного веб-приложения подтверждается тем, что в среднем администраторы вузов тратят от 20 до 40 часов в неделю на составление и корректировку расписаний. При этом около 30% учебных заведений сталкиваются с регулярными конфликтами ресурсов - наложением занятий, нерациональным использованием аудиторного фонда и перегрузкой преподавателей. Рынок программных решений для автоматизации этого процесса, несмотря на наличие коммерческих продуктов, остается недостаточно насыщенным, особенно в сегменте доступных и адаптируемых под российские образовательные стандарты систем.

Разрабатываемое веб-приложение предназначено для широкого круга образовательных учреждений - от крупных университетов до колледжей и школ.

Целью работы является создание системы автоматизации составления учебных расписаний, способной учитывать комплекс ограничений: ресурсные (вместимость аудиторий, их оснащенность), временные (рабочие часы преподавателей), педагогические (логика изучения дисциплин). Научная новизна проекта заключается в разработке гибридного алгоритма оптимизации, сочетающего методы генетического программирования и машинного обучения для прогнозирования нагрузок. Практическая значимость работы определяется потенциальной возможностью сократить время составления расписаний на 80-90%, минимизировать количество организационных конфликтов и повысить эффективность использования ресурсов образовательного учреждения.

Реализация проекта включает несколько этапов: анализ предметной области и существующих решений, проектирование архитектуры системы, разработку алгоритмов оптимизации, создание пользовательского интерфейса и тестирование на реальных данных. В результате будет предложено технологическое решение, способное существенно повысить эффективность организации учебного процесса в образовательных учреждениях различного типа и масштаба.

## 1 Описание и анализ предметной области

Предметная область – автоматизация процесса составления учебных расписаний в образовательных учреждениях с учетом ограничений по аудиториям, преподавателям, времени и другим ресурса [1].

### 1.1 Основные понятия и определения

Учебное расписание – документ, определяющий последовательность и время проведения занятий, распределение учебных нагрузок между преподавателями, а также использование аудиторий и оборудования.

Автоматизация составления расписания – применение алгоритмов и программного обеспечения для оптимизации процесса распределения учебных занятий с учетом заданных ограничений (например, доступность преподавателей, вместимость аудиторий, соблюдение санитарных норм).

Веб-приложение для автоматизации расписаний – программное обеспечение, доступное через браузер, которое позволяет администраторам учебных заведений создавать, редактировать и оптимизировать расписания без ручных расчетов [2].

Минимально жизнеспособный продукт (MVP) – базовая версия системы с ключевыми функциями, достаточными для тестирования гипотез и сбора обратной связи.

Конфликт расписания – ситуация, при которой одно и то же время или ресурс (аудитория, преподаватель) одновременно назначаются для нескольких занятий, что делает расписание некорректным.

Интеграция с внешними системами – возможность подключения к CRM, LMS (Learning Management System) или ERP-системам учебного заведения для автоматического импорта данных о преподавателях, студентах и аудиториях.

Интерфейс администратора – специализированная панель управления, предоставляющая сотрудникам учебного заведения инструменты для настройки параметров генерации расписания, ручной корректировки результатов и управления системой.

Отчетность по расписанию – автоматически формируемые документы и аналитические материалы, отражающие различные аспекты составленного учебного графика. Система предоставляет возможность экспорта данных в стандартные форматы для последующего использования в других информационных системах образовательного учреждения.

Автоматизация составления расписания – это современный технологический подход к организации учебного процесса, который использует специальные алгоритмы и программные решения для эффективного распределения занятий. Такой подход позволяет оптимально учитывать множество факторов: доступность преподавателей, загруженность

аудиторий, особенности учебных программ, требования к оборудованию и другие важные параметры.

Ресурсные ограничения – условия, накладываемые на процесс составления расписания, такие как максимальная нагрузка преподавателей, вместимость аудиторий, требования к оборудованию и другие организационные правила, факторы, влияющие на составление расписания.

Персонализация расписаний – возможность настройки правил под конкретное учебное заведение (например, учет сменности, блокировка определенных временных окон).

Гибкость системы – способность адаптироваться к изменениям (добавление новых групп, отмена занятий, временная замена преподавателей).

Алгоритмы оптимизации расписания – математические методы (например, генетические алгоритмы, метод ветвей и границ, эвристики), применяемые для поиска наиболее эффективного распределения занятий.

Онлайн-доступ и совместная работа – возможность редактирования (изменения, добавления, удаления) расписания несколькими администраторами с синхронизацией изменений в реальном времени.

## 1.2 Обзор существующих систем-аналогов

### 1.2.1 Система «1С:Университет»

«1С:Университет» – это масштабное ERP-решение для комплексной автоматизации всех аспектов деятельности высших учебных заведений, где модуль составления расписаний представляет собой профессиональный инструмент для планирования учебного процесса с учетом специфики вузовской организации.

На рисунке 1 показан интерфейс системы «1С:Университет».

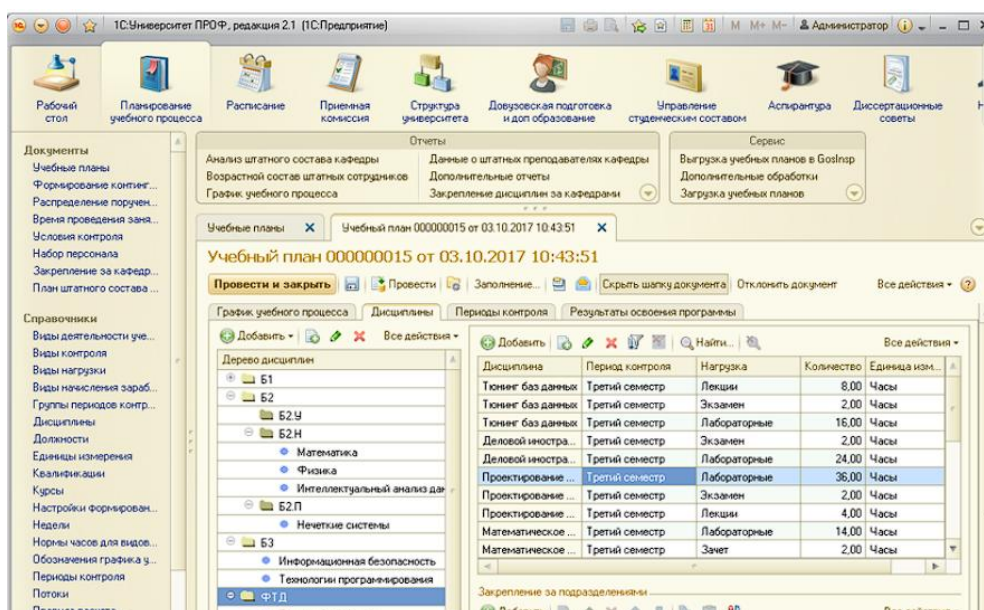


Рисунок 1 – Интерфейс работы программы «1С:Университет»

Ключевые особенности:

- интеграция с бухгалтерскими и кадровыми модулями 1С;
- учет аудиторного фонда, преподавательской нагрузки и специальных требований к занятиям;
- поддержка корректировки расписания в ручном режиме.

Достоинства системы:

- глубокая проработка требований российских вузов;
- возможность формирования различных отчетных форм;
- стабильная работа с большими объемами данных.

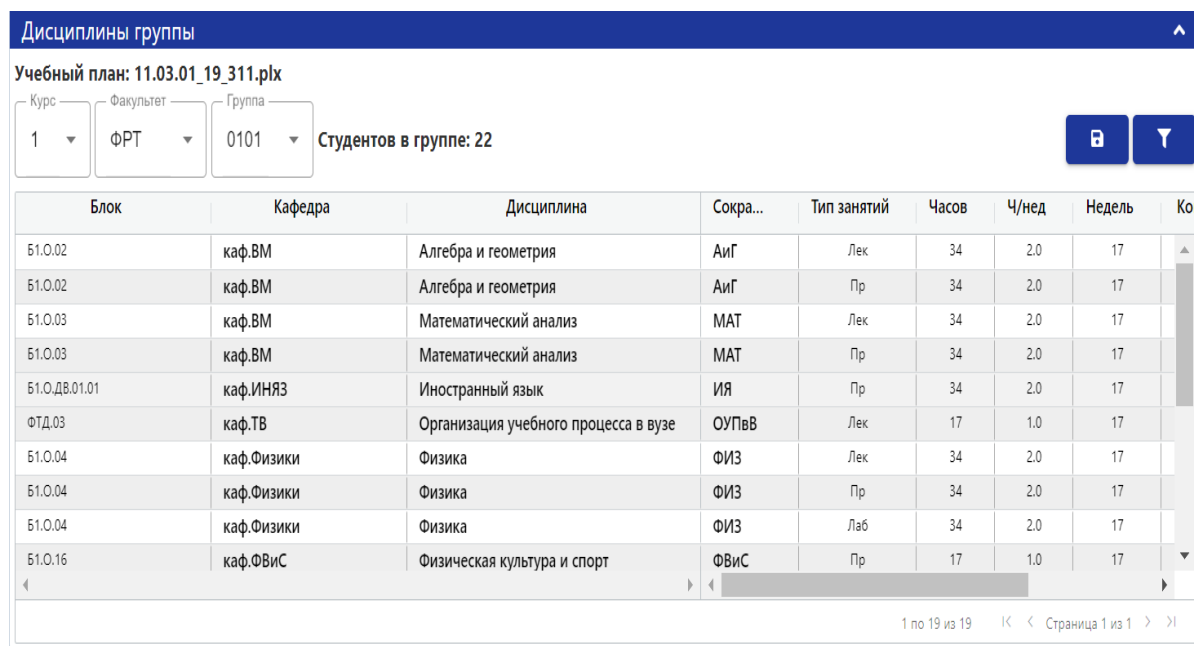
Недостатки системы:

- высокая стоимость внедрения и сопровождения;
- требуется обучение персонала для работы с системой;
- ограниченные возможности облачного взаимодействия.

### 1.2.2 Система «АИС Расписание»

«АИС Расписание» – это комплексное веб-решение нового поколения для автоматизированного проектирования и управления учебными расписаниями в учреждениях общего и профессионального образования (школах и колледжах). Система представляет собой интеллектуальную платформу.

На рисунке 2 показан интерфейс системы «АИС Расписание».



Дисциплины группы

Учебный план: 11.03.01\_19\_311.plx

Курс: 1 | Факультет: ФРТ | Группа: 0101 | Студентов в группе: 22

Блок	Кафедра	Дисциплина	Сокра...	Тип занятий	Часов	Ч/нед	Неделя	Ко
Б1.О.02	каф.ВМ	Алгебра и геометрия	АиГ	Лек	34	2.0	17	
Б1.О.02	каф.ВМ	Алгебра и геометрия	АиГ	Пр	34	2.0	17	
Б1.О.03	каф.ВМ	Математический анализ	МАТ	Лек	34	2.0	17	
Б1.О.03	каф.ВМ	Математический анализ	МАТ	Пр	34	2.0	17	
Б1.О.ДВ.01.01	каф.ИНЯЗ	Иностранный язык	ИЯ	Пр	34	2.0	17	
ФТД.03	каф.ТВ	Организация учебного процесса в вузе	ОУПвВ	Лек	17	1.0	17	
Б1.О.04	каф.Физики	Физика	ФИЗ	Лек	34	2.0	17	
Б1.О.04	каф.Физики	Физика	ФИЗ	Пр	34	2.0	17	
Б1.О.04	каф.Физики	Физика	ФИЗ	Лаб	34	2.0	17	
Б1.О.16	каф.ФВиС	Физическая культура и спорт	ФВиС	Пр	17	1.0	17	

1 по 19 из 19 | < > | Страница 1 из 1

Рисунок 2 – Интерфейс работы программы «АИС Расписание»

Ключевые особенности:

- учет сменности и кабинетной системы;
- контроль нагрузки преподавателей;

- автоматическое обнаружение конфликтов.

Достоинства системы:

- простота использования;
- доступ через браузер без установки;
- поддержка мобильных устройств.

Недостатки системы:

- ограниченные возможности для вузов;
- нет интеграции с популярными LMS;
- минимальная аналитика по загрузке ресурсов.

### 1.2.3 Система «TimeTable» (FET)

FET (Free Timetabling Software) – это бесплатная программа с открытым исходным кодом, предназначенная для автоматического составления учебных расписаний. Разработанная как альтернатива коммерческим решениям, FET предлагает мощный функционал для генерации расписаний с учетом множества ограничений и параметров. Система использует продвинутые алгоритмы оптимизации

На рисунке 3 показан интерфейс программы «TimeTable».

Рисунок 3 – Интерфейс работы программы «TimeTable»

Ключевые особенности:

- поддержка сложных ограничений;
- экспорт в HTML, XML и CSV;
- кроссплатформенность (Windows, Linux, macOS).

Достоинства системы:

- бесплатность и открытый исходный код;
- мощный алгоритм оптимизации;
- подходит для школ и небольших вузов.

Недостатки системы:

- ограниченный бесплатный период (14 дней);
- отсутствие веб-версии;
- нет интеграции с внешними сервисами.

#### 1.2.4 Сравнительный анализ систем-аналогов

В таблице 1 приведены характеристики систем-аналогов.

Таблица 1 – Сравнительные характеристики систем-аналогов

Показатели	Системы для сравнения		
	1С:Университет	АИС Расписание	TimeTable
Автоматическое расписание	+	+	+
Бесплатный доступ	–	+	–
Интеграция с сервисами	1С, Excel	–	–
Учет ресурсных ограничений	+	+	+
Веб-доступ	–	+	–
Понятность интерфейса (1-5)	3	4	2

#### 1.2.5 Постановка задачи

В рамках выпускной квалификационной работы необходимо разработать веб-приложение для автоматизации составления учебных расписаний с учетом ресурсных ограничений. Приложение предназначено для администраторов учебных заведений и позволит оптимизировать процесс распределения занятий, аудиторий и преподавателей, минимизируя конфликты и ручной труд. Автоматизация этого процесса позволяет сократить временные затраты и минимизировать человеческий фактор, обеспечивая оптимальное распределение занятий с учетом всех ограничений.

Приложение должно предоставлять пользователям следующие возможности:

- автоматическое составление расписания на основе заданных параметров;
- ручная корректировка сгенерированного расписания;
- контроль ресурсных ограничений с выявлением конфликтов;
- формирование отчетов в форматах PDF, Excel.

Для начала работы с системой пользователю необходимо пройти регистрацию, указав:

- имя (от 2 до 30 символов);
- фамилию (от 4 до 40 символов);
- электронную почту (должна соответствовать формату email);
- пароль (от 8 символов, включая цифры и буквы).

Система должна проверять корректность введенных данных и выводить соответствующие сообщения об ошибках при их наличии.

При работе с веб-приложением пользователи смогут:

- проходить регистрацию и авторизацию в системе;
- создавать расписание;
- редактировать расписание;
- просматривать статистику и аналитику;
- редактировать данные профиля.

Вся информация, необходимая для работы системы, будет храниться в базе данных, структура которой включает следующие основные сущности:

- пользователи (id, имя, фамилия, email, пароль, роль);
- аудитории (id, номер, вместимость, оборудование);
- преподаватели (id, ФИО, специализация, график работы);
- группы (id, название, учебный план);
- занятия (id, дисциплина, преподаватель, аудитория, время, группа);
- ограничения (id, тип, описание).

Необходимо обеспечить целостность и безопасность хранимых данных.

Таким образом, система должна решать следующие основные задачи:

1) функции системы:

- аутентификация и авторизация пользователей;
- генерация расписания с учетом всех ограничений;
- визуализация расписания (неделя/месяц, цветовая маркировка);
- уведомления об изменениях (email, push-уведомления);

2) технические требования:

- веб-интерфейс с адаптивным дизайном (ПК, планшеты, смартфоны);
- поддержка одновременной работы 100+ пользователей;
- резервное копирование данных.

Система должна быть реализована как веб-приложение с адаптивным интерфейсом, доступным с различных устройств, и обеспечивать стабильную работу при одновременном использовании множеством пользователей.

## 2 Проектирование системы

### 2.2 Выбор и обоснование архитектуры системы

Архитектура системы представляет собой фундаментальную организацию программного обеспечения, определяющую структуру взаимодействия между его компонентами и принципы их совместной работы. В контексте разработки веб-приложений для автоматизации составления расписаний рассматриваются три основных типа архитектур: монолитная, микросервисная и сервис-ориентированная (SOA).

Монолитная архитектура предполагает единую tightly-coupled структуру приложения, где все компоненты (интерфейс, бизнес-логика и доступ к данным) объединены в единое целое. Такой подход отличается простотой разработки и развертывания, но становится проблематичным при масштабировании и внедрении изменений в крупных системах [3].

Микросервисная архитектура декомпозирует приложение на набор слабосвязанных специализированных сервисов, каждый из которых отвечает за конкретную бизнес-функцию. Этот подход обеспечивает высокую масштабируемость и гибкость, но требует сложной оркестрации сервисов и создает дополнительные накладные расходы.

Сервис-ориентированная архитектура (SOA) представляет собой компромиссный вариант, где функциональность системы реализуется в виде набора взаимодействующих сервисов с четко определенными интерфейсами. Такая архитектура обеспечивает баланс между гибкостью и управляемостью системы.

Для разрабатываемого веб-приложения автоматизации составления учебных расписаний была выбрана гибридная архитектура, сочетающая элементы монолитного ядра (для базовых функций) и микросервисного подхода (для специализированных сервисов оптимизации расписаний).

#### 2.1.1 Выбор систем для разработки клиентской части

Разработка клиентской части веб-приложения требует тщательного подхода к выбору технологий. Потому что от этого зависит вид, востребованность приложения, визуальная составляющая всего приложения очень важна. В современной веб-разработке существует несколько популярных фреймворков, каждый из которых предлагает уникальные возможности для создания интерактивных, технологичных пользовательских интерфейсов. Основное внимание было уделено трем наиболее востребованным решениям:

- React.js — библиотека Javascript от Facebook, позволяющая создавать динамические интерфейсы с использованием компонентного подхода;
- Vue.js — прогрессивный фреймворк, отличающийся простотой освоения и гибкостью;

- Angular — полнофункциональный фреймворк от Google для разработки сложных корпоративных приложений.

После детального анализа требований проекта и сравнения характеристик различных технологий, выбор пал на React.js. Это решение обусловлено рядом ключевых факторов. Во-первых, компонентная архитектура react.js идеально подходит для образовательной платформы, где многие элементы интерфейса (карточки курсов, блоки заданий, элементы прогресса) могут быть выделены в переиспользуемые компоненты. Во-вторых, виртуальный dom обеспечивает высокую производительность при работе с большими объемами данных, что критически важно для системы, которая будет обрабатывать множество учебных материалов и результатов учащихся.

Для расширения возможностей react.js и реализации всех требований к интерфейсу образовательной платформы был сформирован следующий технологический стек:

- Redux — для централизованного управления состоянием приложения;
- Material-ui — библиотека компонентов для создания единого визуального стиля;
- Axios — для организации взаимодействия с серверным API;
- Jest — для комплексного тестирования компонентов интерфейса.

Преимущества выбранного технологического стека особенно ярко проявляются в контексте образовательной платформы. Во-первых, React.js в сочетании с material-ui позволяет создавать интуитивно понятные интерфейсы с плавными переходами и анимациями, что улучшает пользовательский опыт. Во-вторых, использование redux обеспечивает предсказуемость состояния приложения, что упрощает отладку и дальнейшее развитие системы. В-третьих, богатая экосистема react предоставляет множество готовых решений для типовых задач, таких как загрузка файлов, отображение прогресса или организация сложных форм.

Важным аспектом выбора стало сообщество разработчиков вокруг React.js. Большое количество обучающих материалов, активное развитие инструментов и регулярные обновления гарантируют, что платформа будет оставаться современной и поддерживаемой в течение длительного времени. Кроме того, популярность React.js среди российских разработчиков упрощает процесс формирования команды для дальнейшего развития проекта.

### 2.1.2 Выбор систем для разработки серверной части

Серверная часть образовательной платформы требует тщательного подхода к выбору технологий, так как от этого зависит надежность, производительность всего решения. Рассмотрим основные варианты, доступные для реализации нашего проекта.

При анализе backend-фреймворков мы рассматривали четыре основных варианта. Node.js с фреймворком Express показал наилучший баланс характеристик для нашего случая. Его event-loop архитектура особенно хорошо подходит для образовательной платформы, где важна обработка множества одновременных подключений студентов и преподавателей. В отличие от PHP, который демонстрирует более низкую производительность, или Java, требующей значительных ресурсов, Node.js обеспечивает оптимальное соотношение скорости работы и потребления памяти.

Особенно стоит отметить поддержку TypeScript в Node.js-экосистеме. Это важное преимущество, позволяющее разрабатывать более надежный и поддерживаемый код по сравнению с Python или PHP-решениями. Для образовательного проекта, где важна стабильность работы и возможность долгосрочной поддержки, это критически важно.

Таблица 3 – Сравнительный анализ серверных технологий

Особенности	Node.js + Express	Python + Django	Java + Spring
Производительность	высокая (event-loop)	средняя	высокая
Экосистема	богатая (npm)	хорошая	очень богатая
Кривая обучения	низкая	средняя	высокая
Поддержка TypeScript	полная	частичная	через дополнительные инструменты
Сообщество	очень активное	активное	крупное
Использование памяти	низкое	среднее	высокое

Для хранения данных я выбрала гибридный подход, сочетающий PostgreSQL и MongoDB. PostgreSQL идеально подходит для структурированных данных: информации о пользователях, расписании, предметах и преподавателях, аудиториях. Его надежная система транзакций гарантирует целостность данных при одновременном доступе множества пользователей.

MongoDB, в свою очередь, будет использоваться для хранения менее структурированных данных: чатов, комментариев, медиафайлов и других данных. Его гибкая схема позволяет легко адаптировать хранилище под меняющиеся требования образовательного процесса. Такой комбинированный подход дает нам преимущества обоих миров: надежность реляционной базы и гибкость NoSQL-решения.

Выбранный технологический стек - Node.js + Express на Google Cloud Platform с гибридной базой данных - представляет собой оптимальное решение. Он сочетает в себе:

- высокую производительность и масштабируемость;
- гибкость для реализации сложной образовательной логики;

- надежное хранение и защиту данных;
- экономическую эффективность;
- простоту интеграции с другими образовательными сервисами.

Этот выбор позволяет нам создать платформу, которая будет стабильно работать при любых нагрузках, обеспечивая бесперебойный образовательный процесс для всех участников. При этом архитектура системы предусматривает возможность дальнейшего масштабирования и добавления новых функций по мере развития проекта.

### 2.3 Выбор комплекса программных средств

Для разработки веб-приложения автоматизации составления учебных расписаний был подобран оптимальный технологический стек, учитывающий специфику работы с ресурсными ограничениями и современные подходы веб-разработки. Основой клиентской части стала связка React.js и TypeScript, выбранная благодаря компонентной архитектуре, идеально подходящей для визуализации сложных расписаний. TypeScript обеспечивает статическую типизацию, критически важную для надежной работы с большими объемами данных, информации о предметах, днях, преподавателях, аудиториях, учебных планов, а виртуальный DOM React гарантирует высокую производительность при частых изменениях в расписании [4].

Серверная часть реализована на Node.js с фреймворком Express.js, что обеспечивает единую среду разработки для клиентской и серверной частей, а также эффективную обработку множества одновременных запросов благодаря event-loop архитектуре. Для ресурсоемких алгоритмов оптимизации расписаний выделен отдельный микросервис на Python с использованием библиотек NumPy и Pandas, позволяющий применять сложные математические методы и генетические алгоритмы.

Система хранения данных построена на комбинированном подходе: PostgreSQL отвечает за структурированные данные пользователей, аудиторий, преподавателей и учебных планов, MongoDB используется для хранения истории изменений расписаний и кэширования промежуточных результатов, а Redis применяется для временного хранения сессий и очередей задач. Такая архитектура обеспечивает как надежность хранения критически важных данных, так и гибкость работы с динамической информацией, всё это очень важно в разрабатываемом проекте.

Инфраструктура развернута на Google Cloud Platform, что предоставляет масштабируемые вычислительные ресурсы для алгоритмов оптимизации, надежное хранение данных и готовые решения для аутентификации. Дополнительные инструменты включают Docker для контейнеризации компонентов системы, Kubernetes для оркестрации

микросервисов, Git для контроля версий и Jest с Cypress для комплексного тестирования. Используются методы линейного программирования

Выбранный технологический стек сочетает несколько ключевых преимуществ: для разработчиков это единая среда на TypeScript с четким разделением ответственности компонентов и богатой экосистемой инструментов; для администраторов - гибкость настройки алгоритмов, простота развертывания и понятный мониторинг; для пользователей - высокая скорость работы, интуитивный интерфейс и надежность хранения данных. Такой подход обеспечивает оптимальный баланс между производительностью, гибкостью и надежностью системы, работающей с сотнями ограничений и ресурсов в режиме реального времени.

## ЗАКЛЮЧЕНИЕ

В ходе проведенного исследования была разработана концепция веб-приложения для автоматизации составления учебных расписаний с учетом ресурсных ограничений. Анализ современных подходов к созданию подобных систем позволил сформировать обоснованный технологический выбор, оптимально соответствующий специфическим требованиям задачи оптимизации учебного процесса.

Для клиентской части приложения был выбран стек на основе React.js и TypeScript, который обеспечивает создание производительного и интуитивно понятного интерфейса для работы со сложными расписаниями. Компонентная архитектура React в сочетании с виртуальным DOM позволяет эффективно визуализировать изменения расписания в реальном времени, что критически важно для пользователей системы.

Серверная часть проекта реализована на базе Node.js с использованием Express.js, что обеспечивает высокую производительность при обработке множества одновременных запросов. Для решения сложных задач оптимизации расписаний разработан специализированный микросервис на Python с применением методов математического программирования, что позволяет учитывать сотни различных ограничений и параметров.

Анализ подходов к хранению данных подтвердил эффективность гибридного решения, сочетающего реляционную СУБД PostgreSQL для структурированных данных и MongoDB для работы с динамической информацией. Развертывание системы на Google Cloud Platform обеспечивает необходимую масштабируемость и надежность работы алгоритмов оптимизации при пиковых нагрузках.

Проведенное исследование продемонстрировало, что предложенная архитектура и выбранные технологические решения позволяют создать эффективный инструмент для автоматизации составления учебных расписаний, способный учитывать комплекс ресурсных ограничений. Дальнейшая работа будет направлена на практическую реализацию предложенной концепции с учетом выявленных в ходе исследования требований и ограничений.


В приложении А приведены презентационные материалы для наглядного ознакомления с изученной темой.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Петров, И.В. Алгоритмы оптимизации учебных расписаний в вузах / И.В. Петров, А.С. Козлов - Москва: Издательство "Инфра-Инженерия", 2023. - 192 с. (дата обращения: 25.06.2025)
- 2 Архитектура информационных систем [Электронный ресурс]. URL: [https://kpfu.ru/staff\\_files/F\\_1284155032/Arkitektura\\_informacionnykh\\_sistem.pdf](https://kpfu.ru/staff_files/F_1284155032/Arkitektura_informacionnykh_sistem.pdf) (дата обращения: 24.06.2025)
- 3 Соловов, А.В. Проектирование и разработка систем управления обучением (LMS). / А.В. Соловов – Санкт-Петербург: Издательство «БХВ-Петербург», 2022. – 416 с. (дата обращения: 24.06.2025)
- 4 Документация по React.js [Электронный ресурс] / URL: <https://react.dev/> (дата обращения: 24.06.2025)

## ПРИЛОЖЕНИЕ А

### Слайды презентационного материала

 САМАРСКИЙ УНИВЕРСИТЕТ  
Институт информатики и кибернетики  
Кафедра программных систем

**НАЗВАНИЕ ПРЕЗЕНТАЦИИ:**  
**РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ  
СОСТАВЛЕНИЯ УЧЕБНЫХ РАСПИСАНИЙ С УЧЕТОМ РЕСУРСНЫХ  
ОГРАНИЧЕНИЙ**

Научно-исследовательская работа бакалавра (получение первичных навыков научно-исследовательской работы)

Выполнил: студент 6201-0203020  
Элекина Е.А.  
Научный руководитель: доцент кафедры  
программных систем, к.т.н. доцент  
Д. А. Попова-Коварцева

Самара 2025

Рисунок А.1 – Титульный слайд

**ПОСТАНОВКА ЗАДАЧИ**

Цель работы:

Разработка веб-приложения для автоматизации составления учебных расписаний

Задачи:

- изучить предметную область;
- изучить методы, технологии, алгоритмы и методики, значимые для рассматриваемой предметной области;
- провести анализ актуальности выбранной задачи;
- рассмотреть системы-аналоги;
- составить список функций разрабатываемого приложения;
- произвести выбор архитектуры разрабатываемой системы и средств реализации.

Разработка веб-приложения для автоматизации составления учебных расписаний с учетом ресурсных ограничений 2/11

Рисунок А.2 – Постановка задачи

## ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

**Учебное расписание** – документ, определяющий последовательность и время проведения занятий, распределение учебных нагрузок между преподавателями, а также использование аудиторий и оборудования.

**Веб-приложение для автоматизации расписаний** – программное обеспечение, доступное через браузер, которое позволяет администраторам учебных заведений создавать, редактировать и оптимизировать расписания без ручных расчетов.



Разработка веб-приложения для автоматизации составления учебных расписаний с учетом ресурсных ограничений

3/11

Рисунок А.3 – Описание предметной области

## АНАЛИЗ СИСТЕМ-АНАЛОГОВ

### 1. «1С:Университет»

Функционал:

- автоматическое составление расписаний с учетом аудиторного фонда и преподавательской нагрузки;
- интеграция с бухгалтерскими и кадровыми модулями 1С;
- формирование отчетных документов по учебному процессу.



Разработка веб-приложения для автоматизации составления учебных расписаний с учетом ресурсных ограничений

4/11

1. Рисунок А.4 – Анализ приложения «1С:Университет»



## АНАЛИЗ СИСТЕМ-АНАЛОГОВ

Показатели	Системы для сравнения		
	1С:Университет	АИС Расписание	TimeTable
Автоматическое расписание	+	+	+
Бесплатный доступ	–	+	–
Интеграция с сервисами	1С, Excel	–	–
Учет ресурсных ограничений	+	+	+
Веб-доступ	–	+	–
Понятность интерфейса (1-5)	3	4	2

Разработка веб-приложения для автоматизации составления учебных расписаний с учетом ресурсных ограничений

7/11

Рисунок А.7 – Сравнительная характеристика систем аналогов

## ПЕРЕЧЕНЬ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

### Функции системы:

- аутентификация и авторизация пользователей;
- генерация расписания с учетом всех ограничений;
- визуализация расписания (неделя/месяц, цветовая маркировка);
- уведомления об изменениях (email, push-уведомления).

Разработка веб-приложения для автоматизации составления учебных расписаний с учетом ресурсных ограничений

8/11

Рисунок А.8 – Перечень функций системы

## ПЕРЕЧЕНЬ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

Требования:

- веб-интерфейс с адаптивным дизайном (ПК, планшеты, смартфоны);
- поддержка одновременной работы 100+ пользователей;
- резервное копирование данных.

Разработка веб-приложения для автоматизации составления учебных расписаний с учетом ресурсных ограничений

9/11

Рисунок А.9 – Перечень функций

## ВЫБОР СИСТЕМ ДЛЯ РАЗРАБОТКИ

**React.js**

обеспечит компонентный подход и высокую производительность за счет виртуального DOM.



**TypeScript**

добавит статическую типизацию, повысив надежность и поддерживаемость кода.



**Node.js**

обеспечит высокую производительность и масштабируемость для обработки множества одновременных подключений.



**PostgreSQL**

для структурированных данных (пользователи, курсы, задания).



Разработка веб-приложения для автоматизации составления учебных расписаний с учетом ресурсных ограничений

10/11

Рисунок А.10 – Выбор систем для разработки

### **Заключение / Выводы**

В результате разработки проекта были выполнены следующие задачи:

1. Изучена предметная область.
2. Изучены методы, технологии, алгоритмы и методики, значимые для рассматриваемой предметной области.
3. Проведен анализ актуальности выбранной задачи.
4. Рассмотрены системы-аналоги.
5. Составлен список функций разрабатываемого приложения.
6. Произведен выбор архитектуры разрабатываемой системы и средств реализации.

**БЛАГОДАРЮ ЗА ВНИМАНИЕ!**

Разработка веб-приложения для автоматизации составления учебных расписаний с учетом ресурсных ограничений

11/11

Рисунок А.11 – Итоги работы