

Alla G. Kravets<sup>1,2</sup>, Natalia A. Salnikova<sup>3</sup>, Inna V. Strukova<sup>1</sup>

## DEVELOPMENT OF A CLIENT-SERVER APPLICATION FOR THE FORMATION OF A SCHEDULE OF MASS ADDITIONAL EDUCATION PROGRAMS

<sup>1</sup>Volgograd State Technical University, Volgograd, Russian Federation

<sup>2</sup>State University Dubna, Moscow region, Dubna, Russian Federation

<sup>3</sup>Volgograd Institute of Management - branch of the Russian Academy of National Economy and Public Administration under the President of the Russian Federation, Volgograd, Russian Federation  
ns3112@mail.ru

*The task of automating the process of creating schedules for the educational process in the context of digitalization of the educational process is especially relevant with the constantly changing schedule of study groups. An automated scheduling system is being developed at Volgograd State Technical University, which not only allows combining already created schedules for individual groups, but also allocates free academic hours for additional classes. The purpose of this work is to develop a client-server application for integrating and processing schedules received from the official website of the university for new study groups. The article analyzes existing methods of creating and processing schedules, obtains the results of a comparative analysis of five main specialized software products designed to automate the process of creating schedules, which have a user-friendly interface and use various algorithms to optimize the schedule. The process of designing a schedule processing program is considered, the functional requirements for a client-server application are described, an analysis of existing methods of working with Excel in programming languages is provided. The process of forming the architecture of a client-server application and designing the interface of a client-server application are described. The article analyzes the options for using the client-server application and describes its implementation. An overview of additional tools used to develop the client-server application is provided. The algorithms of the function for analyzing free pairs, the functions for forming study groups and for forming a schedule, the stages of developing the database and the interface of the client-server application are described, and the results of testing the client-server application are given. During the study, a client-server application was created that ensured the automation of the process of creating schedules for mass additional education programs.*

**Key words:** digitalization of society, curriculum, optimization methods, data analysis, information technologies, client-server applications, framework, database, interface, system testing

Кравец А.Г.<sup>1,2</sup>, Сальникова Н.А.<sup>3</sup>, Струкова И.В.<sup>1</sup>

## РАЗРАБОТКА КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ ДЛЯ ФОРМИРОВАНИЯ РАСПИСАНИЯ ПРОГРАММ МАССОВОГО ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ

<sup>1</sup>Волгоградский государственный технический университет, Волгоград, Российская Федерация

<sup>2</sup>Государственный университет Дубна, Московская область, Дубна, Российская Федерация

<sup>3</sup>Волгоградский институт управления – филиал Российской академии народного хозяйства и государственной службы при Президенте РФ, Волгоград, Российская Федерация  
ns3112@mail.ru

*Задача автоматизации процесса составления расписаний для учебного процесса в условиях цифровизации образовательного процесса особенно актуальна при постоянно изменяющемся графике работы учебных групп. В Волгоградском государственном техническом университете разрабатывается автоматизированная система составления расписания, которая не только позволяет объединять созданные расписания для отдельных групп, а также выделяет свободные академические часы для проведения дополнительных занятий. Целью настоящей работы является построение клиент-серверного приложения для интеграции и обработки расписаний, получаемых с официального сайта университета, для новых учебных групп. В статье анализируются существующие методы составления и обработки расписаний, получены результаты сравнительного анализа 5 основных специализированных программных продуктов, предназначенных для автоматизации процесса составления расписаний, которые имеют удобный интерфейс и используют различные алгоритмы для оптимизации расписания. Рассматривается процесс проектирования программы обработки расписаний, описываются функциональные требования, предъявляемые к клиент-серверному приложению, приводится анализ существующих методов работы с Excel в языках программирования. Описываются процесс формирования архитектуры клиент-серверного приложения и проектирование интерфейса клиент-серверного приложения. Проводится анализ вариантов использования клиент-серверного приложения и описывается его реализация. Приводится обзор дополнительных средств, использованных для разработки клиент-серверного приложения. Описываются алгоритмы функции для анализа свободных пар, функции для формирования учебных групп и для формирования расписания, этапы разработки базы данных и интерфейса клиент-серверного приложения, приводятся результаты тестирования клиент-серверного приложения. В ходе исследования создано клиент-серверное приложение, которое обеспечило автоматизацию процесса составления расписаний для программ массового дополнительного образования.*

**Ключевые слова:** цифровизация общества, учебное расписание, методы оптимизации, анализ данных, информационные технологии, клиент-серверные приложения, фреймворк, база данных, интерфейс, тестирование системы

Статья поступила в редакцию – 22.04.2025  
Принята к публикации – 29.04.2025

## Введение

Разработка расписаний является одной из ключевых задач в образовательных учреждениях, так как от качества и точности составления расписания зависят эффективность учебного процесса, удобство для студентов и преподавателей, а также рациональное использование учебных помещений. В условиях цифровизации образовательных процессов автоматизация этой задачи становится особенно актуальной. В Волгоградском государственном техническом университете (ВолгГТУ) возникла потребность в системе, которая интегрирует существующие расписания, а также помогала выделять свободные академические часы для проведения дополнительных занятий. Современные технологии позволяют создать клиент-серверное приложение, которое упростит процесс составления и оптимизации расписаний, сделав его более гибким и эффективным [1, 2].

Целью настоящей работы является построение клиент-серверного приложения для интеграции и обработки расписаний ВолгГТУ. Это приложение должно автоматизировать процесс получения расписаний с официального сайта университета, анализировать их для выявления свободных академических часов, достаточных для проведения дополнительных занятий, и формировать новые учебные группы и расписания [3, 4].

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ существующих методов составления и обработки расписаний.
- изучить и выбрать технологии для разработки клиент-серверного приложения.
- сформировать функциональные требования и спроектировать интерфейс приложения.
- разработать серверную часть для получения и обработки данных с сайта ВолгГТУ.
- разработать клиентскую часть для взаимодействия с пользователями.
- провести тестирование приложения и оценить его эффективность.

Объектом исследования является процесс формирования расписаний в ВолгГТУ. Предметом исследования являются методы и средства автоматизации процесса с использованием информационных технологий.

Научная новизна работы заключается в построении алгоритмов и методов для автоматического анализа и оптимизации расписаний, которые ранее не применялись в ВолгГТУ. Практическая значимость состоит в возможности использования разработанного приложения для реального улучшения учебного процесса в университете, что позволит более эффективно использовать учебное время и ресурсы.

## Анализ области обработки информации, существующих методов и решений

Управление расписанием в вузе является важным элементом организации учебного процесса. Традиционно составление расписания, включающее использование бумажных или электронных таблиц, требует значительных усилий и времени, что делает процесс трудоемким и подверженным ошибкам [5, 6]. Администраторам необходимо учитывать множество факторов: предпочтения преподавателей и студентов, доступность аудиторий, особенности учебных программ, а также различные ограничения и правила университета.

Одной из основных проблем ручного составления расписания является высокая трудоемкость. Администраторам приходится вручную анализировать множество факторов, что требует значительных временных и умственных затрат. Кроме того, человеческий фактор может привести к ошибкам в распределении занятий, конфликтам в расписании и неэффективному использованию аудиторий и ресурсов университета [7].

Для эффективного управления расписаниями в ВолгГТУ необходимо провести анализ различных данных. Эти данные включают информацию о занятиях, учебных группах, преподавателях, аудиториях и временных интервалах [8]. Понимание структуры и формата этих данных является важным шагом для разработки эффективных автоматизированных систем управления расписанием [9].

Основные источники данных для составления расписаний включают университетскую базу данных, системы управления обучением и внешние источники, такие как информация о временных изменениях. Анализ этих источников помогает определить доступность и формат данных.

Оценка объема данных, включая количество занятий, учебных групп, преподавателей и аудиторий, а также частота изменения этого объема, позволяет определить требования к производительности и масштабируемости программы. Анализ частоты обновления данных, включая изменения в учебном плане, добавление новых курсов и изменения доступности преподавателей и аудиторий, помогает определить необходимость и методы обновления данных в программе [10, 11].

Основные операции обработки данных, такие как оптимизация распределения ресурсов, учет ограничений и предпочтений, а также выбор методов и алгоритмов для их реализации, являются ключевым шагом для эффективного управления расписаниями [12, 13].

Таким образом, автоматизация процесса составления расписания в учебных заведениях позволяет значительно сократить временные затраты и уменьшить количество ошибок. Это обеспечивает более эффективное использование учебного времени и ресурсов, а также повышает качество образовательного процесса.

## Проектирование программы и клиент-серверного приложения для интеграции и обработки расписаний ВолгГТУ

**Описание процесса.** Формирование групп для Цифровой кафедры (ЦК) ВолгГТУ в настоящее время выполняется вручную и состоит из нескольких этапов. Вначале необходимо скачать файлы с расписанием занятий на локальный компьютер. Затем формируется список студентов, которые обучаются на Цифровой кафедре, основываясь на направлениях их обучения.

После этого начинается анализ направлений и академических групп студентов. Выбирается студент, записавшийся на направление ЦК, и выясняется, в какой академической группе он обучается. Если в той же группе имеются другие студенты, также записавшиеся на это направление, они включаются в одну группу ЦК. Далее проверяется наличие студентов с курса, записавшихся на заданное направление. Важно, чтобы у этих студентов совпадали свободные учебные пары, а именно две учебные пары подряд. Если такие совпадения обнаружены, то студенты объединяются в группу. Если на определенном этапе группа достигает 20 человек, то она считается сформированной, и процесс продолжается для следующих студентов.

Если на текущем курсе факультета не удается сформировать группу, анализ продолжается для следующего курса. После рассмотрения всех курсов одного факультета, проверяются другие факультеты, с предпочтением объединения студентов одного факультета в одну группу.

После анализа расписания ВолгГТУ и формирования групп начинается процесс формирования групп для других вузов. Для студентов из других вузов не проводится анализ совпадений расписания. Группы формируются оптимально, и, если возможно, объединяются со студентами ВолгГТУ. В основном, группы для ЦК, состоящие из студентов других вузов, занимаются в вечернее время.

Сформированные группы распределяются по дням недели таким образом, чтобы все студенты из академических групп, входящих в состав групп ЦК, могли заниматься. Каждая группа должна иметь 4-5 занятий в двухнедельном расписании, аналогичном расписанию ВолгГТУ. Расписание создается в виде Excel-файла, в котором указывается расписание занятий, а также формируется отдельный Excel-файл со списками сформированных групп.

Процесс формирования групп для Цифровой кафедры в ВолгГТУ представляет сложную и длительную задачу, в целом он занимает от 16 до 28 часов.

**Функциональные требования, предъявляемые к клиент-серверному приложению.** Для эффективной автоматизации процесса формирования групп для ЦК в ВолгГТУ необходимо разработать клиент-серверное приложение, соответствующее функциональным требованиям. Основные функции, которые должны быть реализованы в приложении следующие:

1. Загрузка и обработка расписания:
  - приложение должно иметь возможность автоматически загружать расписание с официального сайта ВолгГТУ;

- приложение должно обрабатывать расписание, представленное в форматах .xls и .xlsx;
- приложение должно уметь выявлять свободные академические часы и определять две свободные пары подряд, так как пары на ЦК продолжаются четыре академических часа;
- необходимо обеспечить корректную обработку данных о датах лабораторных и практических занятий для дальнейшего формирования учебных групп и расписаний.

2. Загрузка и обработка списка студентов ЦК:
  - приложение должно позволять загружать список студентов, записавшихся на ЦК, в формате Excel;
  - в списке должны быть указаны данные о вузе, факультете, группе и направлении на Цифровой кафедре;
  - загруженные данные должны обрабатываться корректно.

3. Формирование учебных групп:
  - приложение должно анализировать совпадения свободных академических часов в расписании вуза у загруженного списка студентов;
  - на основе совпадений свободных часов приложение должно автоматически формировать учебные группы;
  - учебные группы должны распределяться оптимально по времени и составу, минимизируя конфликты в расписании.

4. Создание расписания для ЦК:
  - приложение должно автоматически составлять расписание для ЦК на основе сформированных учебных групп и анализа свободных часов;
  - расписание должно быть оптимизировано по времени проведения занятий, чтобы избежать пересечений;
  - приложение должно предусматривать возможность обновления расписания при изменении исходных данных, не нарушая уже сформированных учебных групп.

5. Экспорт данных:
  - приложение должно предоставлять возможность скачивания сформированных учебных групп и расписаний в формате Excel;
  - экспортанные данные должны быть структурированы и легко читаться.

6. Пользовательский интерфейс:
  - интерфейс должен включать кнопки для загрузки расписания с сайта ВолгГТУ и файла с распределением студентов по профилям ЦК;
  - должны быть предусмотрены кнопки для запуска процессов формирования групп и расписаний;
  - интерфейс должен быть интуитивно понятным и удобным для пользователя;
  - все основные функции должны быть доступны с

главного экрана приложения.

Таким образом, клиент-серверное приложение должно охватывать весь процесс автоматизации формирования учебных групп и составления расписания для Цифровой кафедры ВолгГТУ. Это позволит существенно снизить временные затраты, уменьшить количество ошибок и повысить эффективность учебного процесса. На рис. 1 представлена структурная схема системы со связями между блоками.

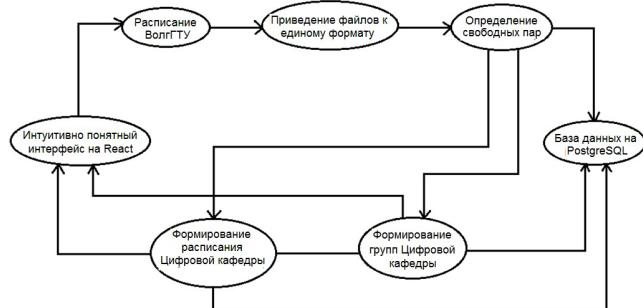


Рис. 1. Структурная схема приложения

**Анализ существующих методов работы с Excel в языках программирования.** Для разработки клиент-серверного приложения, способного обрабатывать расписания ВолгГТУ, важно выбрать оптимальный способ работы с файлами Excel. После анализа основных методов работы с Excel в различных языках программирования: Python, JavaScript, Java и C#, проведен их сравнительный анализ, результаты которого отображены в таблице 1.

Таблица 1. Результаты сравнения функционала языков программирования

Библиотека	Поддержка форматов	Функциональность	Производительность	Требования к ресурсам
pandas	.xlsx	Мощный анализ данных	Средняя	Высокие
openpyxl	.xlsx	Широкие возможности	Средняя	Средние
xlrd/xlwt	.xls	Ограниченнная	Высокая	Низкие
xlsx	.xlsx	Гибкая работа с данными	Низкая	Низкие
exceljs	.xlsx	Расширенные возможности	Средняя	Средние
Apache POI	.xls, .xlsx	Широкие возможности	Средняя	Высокие
JExcelAPI	.xls	Ограниченнная	Высокая	Низкие
EPPlus	.xlsx	Расширенные возможности	Высокая	Средние
NPOI	.xls, .xlsx	Широкие возможности	Средняя	Средние

После анализа различных методов работы с Excel-файлами выбран Python с использованием библиотеки openpyxl. Основные преимущества openpyxl включают поддержку создания и модификации сложных таблиц, что особенно важно для работы с расписаниями, а также широкие возможности для форматирования данных. Библиотека обеспечивает высокую производительность и удобство использования [14, 15].

Таким образом, выбор Python с библиотекой openpyxl предоставляет оптимальный баланс между функциональностью и производительностью для задач по интеграции и обработке расписаний ВолгГТУ.

**Формирование архитектуры клиент-серверного приложения.** Архитектура приложения для интеграции и обработки расписаний ВолгГТУ представлена на рис. 2.

Архитектура приложения построена следующим

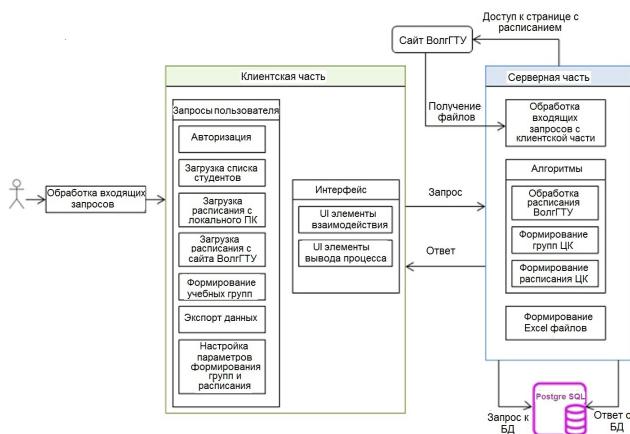


Рис. 2. Архитектура приложения

образом:

- все компоненты приложения разворачиваются в контейнерах Docker, что обеспечивает изолированность окружений и упрощает процесс развертывания [16, 17]. Docker-контейнеры позволяют создавать одинаковые условия для разработки, тестирования и эксплуатации, что снижает вероятность ошибок и упрощает масштабирование приложения;
- пользователь взаимодействует с интерфейсом через веб-приложение, разработанное на React [18, 19]. В интерфейсе находятся кнопки для загрузки расписания, формирования групп и обновления данных [20]. Все команды, введенные пользователем, отправляются на сервер для дальнейшей обработки;
- серверная часть приложения реализована с использованием FastAPI. FastAPI сервер управляет логикой приложения, обрабатывая запросы от клиента, взаимодействуя с базой данных [21, 22]. Сервер принимает запросы от клиентской части, выполняет необходимые вычисления и операции с данными [23]. При загрузке расписания сервер анализирует данные и выявляет свободные академические пары. На основе данных о студентах и расписаниях сервер формирует учебные группы, оптимизируя их по времени и составу;
- в качестве базы данных используется PostgreSQL. Серверная часть отправляет запросы к базе данных для чтения, записи и обновления данных, причем PostgreSQL обеспечивает надежное и эффективное хранение данных, связанных с расписаниями, студентами и сформированными группами, а также быструю их выборку [24, 25];
- приложение позволяет пользователям экспортить сформированные группы и расписания в формате Excel. Это делает работу с данными вне системы удобной и позволяет легко делиться ими с другими пользователями.

## Разработка алгоритмов обработки расписания

**Разработка алгоритма для анализа свободных пар.** Для анализа свободных пар в расписании используется алгоритм, который обрабатывает данные из Excel-файла. Основная задача алгоритма заключается в определении наличия пары, её продолжительности и дат проведения [26]. Для этого применяется концепция "рабочей зоны" – группы ячеек размером 4 на 3, содержащих всю необходимую информацию о конкретной паре.

Вначале выполняется загрузка файла Excel и определение учебной программы и факультета по имени файла. Далее инициализируется рабочая зона и создается словарь для хранения данных о группах.

Для каждой пары в расписании используется структура данных, в которой хранится информация о продолжительности, датах проведения и наличии занятия. В

пределах одного дня структура данных для пар выглядит следующим образом: для каждой пары хранятся её продолжительность, даты проведения и информация о наличии занятия.

Для определения дат занятий используется функция, которая проверяет содержимое ячейки на наличие строк, похожих на даты. Если такие строки обнаруживаются, то они добавляются в массив дат. Определение продолжительности пары осуществляется аналогично: функция проверяет содержимое ячейки на наличие информации о продолжительности пары, выраженной в часах.

Алгоритм обработки данных заключается в следующем: для каждой пары проверяется наличие занятия, определяются даты и продолжительность. Если пара имеет место, то информация о ней добавляется в соответствующую структуру данных. Если у предыдущей пары продолжительность больше одной, то даты предыдущей пары переносятся на текущую пару, что обеспечивает корректное сохранение дат и продолжительности пар. После обработки данных текущей пары рабочая зона перемещается вниз для анализа следующей пары. Алгоритм определения и записи информации о парах приведен на рис. 3.

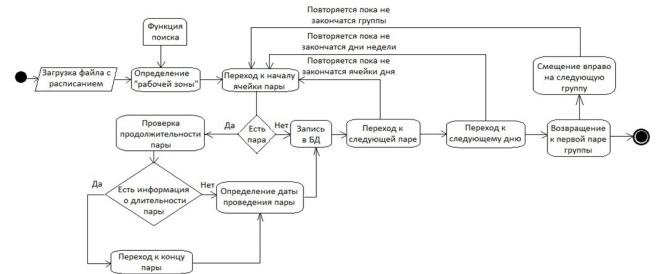


Рис. 3. Алгоритм определения и записи информации о парах

Этот процесс продолжается до конца дня. Затем информация отправляется в базу данных и алгоритм переходит к анализу следующего дня. После завершения анализа всех дней недели для одной группы рабочая зона сдвигается вправо и поднимается наверх для анализа следующей группы. Процесс повторяется до тех пор, пока не будут обработаны все группы.

Разработанный алгоритм позволяет эффективно обрабатывать расписание, анализируя каждую пару и собирая необходимую информацию. Автоматизация этого процесса значительно сокращает временные затраты и уменьшает вероятность ошибок, связанных с ручной обработкой данных. Внедрение такого алгоритма обеспечивает высокую точность и надежность формирования расписаний для Цифровой кафедры ВолгГТУ.

**Разработка алгоритма формирования учебных групп.** Для формирования учебных групп и примерного расписания Цифровой кафедры ВолгГТУ используется комплексный алгоритм. Первоначально происходит подключение к базе данных и извлечение необходимых данных, включая списки студентов, направления ЦК и их учебные группы, а также списки свободных дат и временных интервалов пар ЦК, которые соответствуют двум парам ВолгГТУ. Затем создается словарь, в котором ключами являются программы ЦК, а значениями – списки студентов, записанных на эти программы.

На следующем этапе осуществляется формирование групп для каждой программы ЦК. Студенты объединяются по их исходным академическим группам ВолгГТУ, чтобы студенты из одной академической группы ВолгГТУ были в одной группе ЦК. Одновременно проверяются свободные пары студентов для совпадения и распределения мест в будущем расписании. Даты и временные интервалы возможных занятий для групп сохраняются в таблицу примерного расписания.

При формировании групп учитывается заданное ограничение на максимальное количество студентов в группе. Если количество студентов в текущей группе достигает этого предела, то формируется новая группа, и оставшиеся студенты перераспределяются. Таким образом, минимизируется количество малых групп, примерно меньше 5 человек.

Для выявления оптимального ограничения на количество студентов в группе алгоритм циклически проходит по разным значениям в диапазоне от 20 до 25. Оптимальной является компоновка, при которой остается минимальное количество малых групп. Это позволяет эффективно распределить студентов по группам.

Одновременно с формированием групп происходит предварительное распределение занятий по дням недели и времени. Для каждой группы определяется количество пар, которые они могут посещать одновременно. Эти данные сохраняются в базе данных, что позволяет на следующем этапе создать финальное расписание для ЦК. Алгоритм формирования учебных групп ЦК приведен на рис. 4.

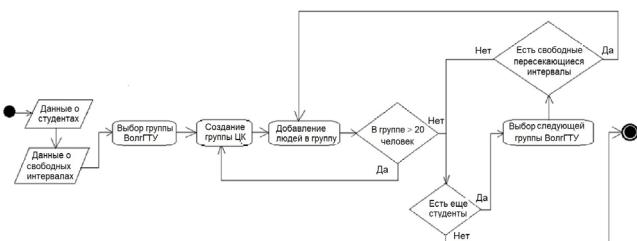


Рис. 4. Алгоритм формирования групп ЦК

Разработанный алгоритм обеспечивает точное распределение студентов по группам с учетом их свободных интервалов, минимизируя временные затраты и снижая вероятность ошибок, связанных с ручной обработкой данных. Это способствует созданию оптимального расписания, соответствующего потребностям студентов, и повышает эффективность учебного процесса в целом.

**Разработка алгоритма формирования расписания.** Для создания точного и удобного расписания учебных групп необходимо разработать функцию, которая распределяет занятия по доступным временным интервалам, учитывая ограничения и избегая конфликтов.

Вначале функция подключается к базе данных и подготавливает необходимые структуры данных и переменные. Важным шагом на этом этапе является очистка таблицы, предназначенной для хранения данных о новых интервалах занятий, чтобы подготовить её для обновления и избежать дублирования данных.

Затем происходит загрузка из базы данных. На этом этапе извлекаются все свободные временные интервалы и сведения обо всех сформированных учебных группах. Эти данные критически важны для дальнейшего анализа и распределения занятий по временным интервалам. Свободные временные интервалы определяют, когда возможны занятия.

После загрузки данных начинается этап формирования расписания. Для каждой группы инициализируются счетчики и структуры данных, такие как списки занятых интервалов и занятых дней. Затем проводится проверка каждого свободного интервала на превышение лимита занятий для текущей группы. Если лимит не превышен, то выполняется проверка на пересечения интервалов с занятыми на эту дату. Также проверяется занятость данной группы. Если интервал свободен и день не занят, то интервал добавляется в расписание, и соответствующие счетчики обновляются. Это позволяет эффективно использовать доступное время и минимизировать конфликты в расписании. Алгоритм формирования расписания ЦК приведен на рис. 5.

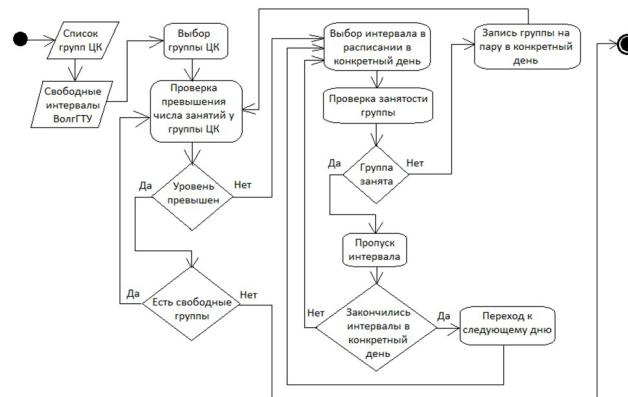


Рис. 5. Алгоритм формирования расписания ЦК

После того, как расписание сформировано для всех групп, результаты сохраняются в таблице базы данных. Это обеспечивает доступность данных для дальнейшего использования и анализа.

Основные этапы функции формирования расписания:

1. подключение к базе данных и подготовка данных:
  - подключение к базе данных;
  - подготовка необходимых структур данных и переменных;
  - очистка таблицы для хранения данных о новых интервалах занятий;
2. загрузка данных:
  - загрузка всех свободных временных интервалов из базы данных;
  - загрузка сведений обо всех сформированных учебных группах;
3. формирование расписания:
  - инициализация счетчиков и структур данных для каждой группы;
  - проверка каждого свободного интервала на предмет превышения лимита занятий;
  - проверка на пересечения интервалов с занятыми на определенную дату;
  - проверка, занят день для данной группы;
  - добавление интервала в расписание и обновление соответствующих счетчиков;
4. сохранение итогового расписания:
  - сохранение результатов в таблице базы данных.
  - Функция для формирования расписания автоматизирует процесс распределения занятий по доступным временным интервалам, что позволяет избежать конфликтов и пересечений. Данные о сформированном расписании сохраняются в базе данных. После этого они могут быть собраны в Excel-файл для дальнейшего использования и анализа. Такой подход значительно упрощает процесс составления расписания и повышает его точность и эффективность.

## Реализация клиент-серверного приложения

**Описание функционала разработанного приложения.** Разработанное клиент-серверное приложение для формирования учебных групп и расписаний на Цифровой кафедре ВолгГТУ обеспечивает пользователям удобный и интуитивно понятный интерфейс, позволяя автоматизировать процесс создания учебных групп и расписаний. Приложение включает функциональные возможности, как со стороны клиента, так и со стороны сервера.

Приложение состоит из нескольких основных страниц: главная страница загрузки данных, страница обработки данных и страница скачивания данных.

При входе в систему пользователь проходит процедуру авторизации. Авторизация выполняется с исполь-

зыванием логина и пароля, затем пользователь получает доступ к основным функциям приложения.

На главной странице загрузки данных (рис. 6) пользователю предоставляется возможность загрузить расписание двумя способами: загрузив расписание непосредственно с сайта ВолгГТУ или конкретный файл расписания с компьютера пользователя. После успешной загрузки данных на странице отображается дата и время последнего действия, что позволяет пользователю отслеживать актуальность загруженной информации.

Рис. 6. Экранная форма страницы загрузок

После загрузки данных пользователь может перейти на страницу обработки данных (рис. 7). Здесь находится кнопки "Сформировать", которые запускают формирования групп. Приложение анализирует расписание студентов и объединяет их в группы с учетом их учебных графиков. Процесс формирования групп отображается на экране.

Рис. 7. Экранная форма страницы обработки

На этой странице пользователь может воспользоваться кнопкой "Сформировать", которая позволяет автоматически генерировать расписание занятий для сформированных групп. Приложение использует данные о свободных временных интервалах и датах, чтобы создать оптимальное расписание, избегая конфликтов и пересечений занятий. Также пользователю предоставляется возможность выбора числа занятий у групп из диапазона от 4 до 8. По умолчанию в системе установлено 5 пар.

На странице скачивания данных пользователю предоставляются кнопки для скачивания сформированного списка групп и расписания (рис. 8). Первая кнопка

"Скачать" позволяет загрузить сформированные группы в формате Excel, а вторая предоставляет возможность скачать составленное расписание занятий также в формате Excel. Это обеспечивает удобство использования данных вне системы и возможность их распространения.

Рис. 8. Экранная форма страницы загрузок

Серверная часть приложения, реализованная на базе FastAPI и PostgreSQL, обеспечивает надежную и быструю обработку данных. Когда пользователь загружает расписание или список студентов, сервер принимает эти файлы и извлекает из них необходимые данные.

Сервер анализирует загруженные данные, используя алгоритмы для определения свободных временных интервалов и составления оптимальных учебных групп. На основе сформированных групп и доступных временных интервалов сервер генерирует расписание занятий. В процессе используются данные о свободных временных промежутках, загруженные из расписания.

Сервер также генерирует файлы Excel со списками групп и расписанием, которые могут быть загружены пользователями. Это обеспечивает удобный способ получения и использования данных вне системы.

**Входные и выходные данные.** Входными данными для клиент-серверного приложения являются:

- расписание занятий ВолгГТУ в формате .xls или .xlsx;
- список студентов.

Выходными данными, для разработанного клиент-серверного приложения являются:

- база данных с информацией о свободных парах в группах ВолгГТУ, начиная со 2 курса;
- сформированные учебные группы;
- расписание занятий для Цифровой кафедры;
- Excel файл с группами ЦК;
- Excel файл с расписанием ЦК.

Таким образом, входные данные включают расписания, списки студентов и параметры формирования расписания, а выходные данные представлены в виде сформированных групп, расписания и отчетов об обработке данных.

**Тестирование клиент-серверного приложения.**

Тестирование проводится на удаленном сервере.

Характеристики сервера:

- процессор 2 ГГц;
- 2 ядра;
- память 4 ГБ ОЗУ;
- SSD 30 Гб.

Тестирование клиент-серверного приложения направлено на проверку корректности выполнения ключевых функций, таких как скачивание расписания, конвертация расписания, парсер расписания, обработка файла со списком студентов, формирование групп, формирование расписания. Эти тесты помогают убедиться, что клиент-серверное приложение работает стабильно и надежно, обеспечивая точность и эффективность обработки данных. Результаты тестирования приведены в таблице 2.

Таблица 2. Тесты функций, развернутых на сервере

Описание теста	Входные данные	Ожидаемый результат	Фактический результат	Время выполнения (с)
Получение файлов с сайта ВолгГТУ	Запрос к сайту	Файлы успешно загружены	Файлы загружены	17.97
Конвертация файлов в формат .xlsx	Загруженные файлы .xls и .xlsx	Определены файлы в формате .xls и преобразованы в .xlsx	Необходимые файлы конвертированы	289.01
Парсинг расписания	Файлы с расписанием	Выявлены свободные часы у групп и занесены в БД	Часы выявлены и занесены	1301.42
Обработка файла студентов	Файл со списком студентов	Выявлены направления ЦК и занесены в БД: список студентов, ВУЗ, факультет, группа	Направления выявлены и все данные занесены в БД	3.54
Формирование групп	Данные с БД о студентах и расписании	Сформированы оптимальные группы, все люди из которых могут заниматься в общий день	Группы сформированы	11.74
Формирование расписания	Данные с БД о группах и временных интервалах	Расписание сформировано без ошибок и пересечений с расписанием ВолгГТУ	Расписание сформировано	0.59

## Заключение

В ходе исследования разработана программа для интеграции и обработки расписаний ВолгГТУ. Основной задачей проекта являлось создание клиент-серверного приложения, которое автоматизирует процесс получения расписаний с официального сайта университета, их анализа для выявления свободных академических часов и формирования учебных групп и расписания для Цифровой кафедры.

Результатом работы стала система, значительно сокращающая временные затраты на формирование расписаний, уменьшающая количество ошибок, связанных с ручной обработкой данных, и повышающая эффективность учебного процесса. Программа обеспечивает автоматическую загрузку расписаний, анализирует их для выявления свободных пар, формирует учебные группы на основе совпадений свободных часов и создаёт расписания для ЦК.

Перспективы дальнейшего развития проекта включают расширение функционала, добавление модуля для автоматического составления расписания экзаменов и консультаций, внедрение алгоритмов машинного обучения для оптимизации расписания.

## Литература

1. Исаев А.В., Кравец А.Г., Шахламджян А.Ш. Дистанционное образование: анализ информационных ресурсов. // Известия Волгоградского государственного технического университета. 2010. № 6 (66). С. 100-103.
2. Лопухов Н.В., Сальникова Н.А. Моделирование профессиональной деятельности в образовательном процессе. // Научный вестник Волгоградского филиала РАНХиГС. Серия: Экономика. 2011. Т. 2. № 6. С.84-89.
3. Кравец А.Г., Асеева С.Д. Методы и программные средства оценки эффективности

деятельности преподавателей опорных вузов. // Прикаспийский журнал: управление и высокие технологии. 2016. № 1 (33). С. 90-102.

4. Сальникова Н.А. Адаптивное тестирование как инструмент повышения качества учебного процесса. // Известия Волгоградского государственного технического университета. Серия: Новые образовательные системы и технологии обучения в вузе. 2011. Т. 8. № 10 (83). С.126-129.

5. Бирюков И.С., Ковалев И.В., Левченко Д.Р., Петров В.С., Плотников И.П., Матохина А.В., Щербакова Н.Л., Соколов А.А. Обзор CRM систем по работе с клиентами и сотрудниками центров дополнительного образования. // Наукосфера. 2021. № 10-1. С. 104-107.

6. Самсонова Н.В., Симонов А.Б. Составление расписания в высшем учебном заведении: математические методы и программные продукты. // Инstrumentальные и математические методы в процессах управления. 2018. Т. 1. № 1. С. 60-69.

7. Чан В.Фу, Щербаков М.В., Нгуен Т.А., Скоробогатченко Д.А. Метод сбора и слияния разнотипных данных в проактивных системах интеллектуальной поддержки принятия решений. // Нейрокомпьютеры: разработка, применение. 2016. № 11. С. 40-44.

8. Панкратьев Е. В., Чеповский А. М., Черепанов Е. А., Чернышев С. В. Алгоритмы и методы решения задач составления расписаний и других экстремальных задач на графах больших размерностей. // Фундаментальная и прикладная математика. 2003. Т. 9. № 1. С. 235—251.

9. Нестеров Б.В., Кизим А.В., Матохина А.В. Программа автоматизации наполнения онтологий. // Свидетельство о регистрации программы для ЭВМ RU 2017611759, 09.02.2017. Заявка № 2016663680 от 13.12.2016.

10. Чан В.Ф., Щербаков М.В., Нгуен Т.А. Обзор архитектур систем поддержки принятия решений, использующих аналитику данных в режиме реального времени. // Известия Волгоградского государственного технического университета. 2016. № 3 (182). С. 95-100.

11. Лазарев А.А., Гафаров Е.Р. Теория расписаний: задачи и алгоритмы. Москва. 2019. 222 с.

12. Масляев Д. А. Современное состояние задачи автоматизации составления оптимального учебного расписания в вузе. // Вестник Сыктывкарского университета. Сер. 1: Математика. Механика. Информатика. 2022. Вып. 1 (42). С. 23-40.

13. Бэнкс А., Порсельло Е. Современные шаблоны для разработки приложений. Санкт-Петербург. Питер. 2022. 320 с.

14. Использование Python и Excel для обработки и анализа данных. URL: <https://habr.com/ru/companies/otus/articles/331998/> (Дата обращения: 20.04.2025).

15. openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files. URL: <https://openpyxl.readthedocs.io/en/stable/> (Дата обращения: 20.04.2025).

16. DockerCompose (2017) Environment variables in compose // Docker Documentation. URL: <https://docs.docker.com/compose/environment-variables/> (Дата обращения: 20.04.2025).

17. Чиганов, Д. Р. Docker ключ к контейнеризации и масштабируемости. // Вестник науки. 2023. Т. 4 (64). №7. С. 270-272.

18. React: библиотека для создания пользовательских интерфейсов. URL: <https://ru.reactjs.org/> (Дата обращения: 20.04.2025).

19. Quick Start – React. // Documentation. URL: <https://react.dev/learn> (Дата обращения: 25.03.2025).

20. Дакетт, Д. HTML и CSS. Разработка и создание веб-сайтов. Москва. Эксмо. 2020. 480 с.

21. Пишем FastAPI с нуля. URL: <https://habr.com/ru/>

articles/708678/ (Дата обращения: 20.04.2025).

22. FastAPI documentation – DevDocs // FastAPI Documentation. URL: <https://devdocs.io/fastapi/> (Дата обращения: 20.04.2025).

23. Косс Е. Н., Белеевский К.А., Васецкий А.М. Создание серверной части для автоматизации формирования расписания университета. // Успехи в химии и химической технологии. 2021. Т. 35. №10. С. 83-85.

PostgreSQL: The World's Most Advanced Open Source Relational Database. // Documentation. URL: <https://www.postgresql.org/> (Дата обращения: 11.01.2025).

PostgreSQL // Documentation. URL: <https://www.postgresql.org/docs/> (Дата обращения: 11.01.2025).

26. FET Free Timetabling Software // FET Documentation. – URL: <https://lalescu.ro/liviu/fet/download.html> (Дата обращения: 02.03.2025).

## References

1. Isaev A.V., Kravets A.G., Shahlamdzhan A.Sh. Distancionnoe obrazovanie: analiz informacionnyh resursov. // Izvestiya Volgogradskogo gosudarstvennogo tekhnicheskogo universiteta. 2010. № 6 (66). S. 100-103.

2. Lopuhov N.V., Sal'nikova N.A. Modelirovaniye professional'noj deyatel'nosti v obrazovatel'nom processe. // Nauchnyj vestnik Volgogradskogo filiala RANHiGS. Seriya: Ekonomika. 2011. Т. 2. № 6. S.84-89.

3. Kravets A.G., Aseeva S.D. Metody i programmnye sredstva ocenki effektivnosti deyatel'nosti prepodavatelej opornyh vuzov. // Prikaspijskij zhurnal: upravlenie i vysokie tekhnologii. 2016. № 1 (33). S. 90-102.

4. Sal'nikova N.A. Adaptivnoe testirovanie kak instrument povysheniya kachestva uchebnogo processa. // Izvestiya Volgogradskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Novye obrazovatel'nye sistemy i tekhnologii obucheniya v vuze. 2011. Т. 8. № 10 (83). S.126-129.

5. Biryukov I.S., Kovalev I.V., Levchenko D.R., Petrov V.S., Plotnikov I.P., Matohina A.V., Shcherbakova N.L., Sokolov A.A. Obzor CRM sistem po rabote s klientami i sotrudnikami centrov dopolnitel'nogo obrazovaniya. // Naukosfera. 2021. № 10-1. S. 104-107.

6. Samsonova N.V., Simonov A.B. Sostavlenie raspisaniya v vysshem uchebnom zavedenii: matematicheskie metody i programmnye produkty. // Instrumental'nye i matematicheskie metody v processah upravleniya. 2018. Т. 1. № 1. S. 60-69.

7. Chan V.Fu, Shcherbakov M.V., Nguen T.A., Skorobogatchenko D.A. Metod sbora i sliyaniya raznotipnyh dannyh v proaktivnyh sistemah intellektual'noj podderzhki prinyatiya reshenij. // Nejrokomp'yutery: razrabotka, primenie. 2016. № 11. S. 40-44.

8. Pankrat'ev E. V., Chepovskij A. M., Cherepanov E. A., Chernyshev S. V. Algoritmy i metody resheniya zadach sostavleniya raspisanij i drugih ekstremal'nyh zadach na grafe bol'shih razmernostej. // Fundamental'naya i prikladnaya matematika. 2003. Т. 9. №

1. S. 235—251.

9. Nesterov B.V., Kizim A.V., Matohina A.V. Programma avtomatizacii napolneniya ontologij. // Svidetel'stvo o registracii programmy dlya EVM RU 2017611759, 09.02.2017. Zayavka № 2016663680 ot 13.12.2016.

10. Chan V.F., Shcherbakov M.V., Nguen T.A. Obzor arhitektur sistem podderzhki prinyatiya reshenij, ispol'zuyushchih analitiku dannyh v rezhime real'nogo vremeni. // Izvestiya Volgogradskogo gosudarstvennogo tekhnicheskogo universiteta. 2016. № 3 (182). S. 95-100.

11. Lazarev A.A., Gafarov E.R. Teoriya raspisanij: zadachi i algoritmy. Moskva. 2019. 222 s.

12. Maslyaev D. A. Sovremennoe sostoyanie zadachi avtomatizacii sostavleniya optimal'nogo uchebnogo raspisaniya v vuze. // Vestnik Syktyvkarskogo universiteta. Ser. 1: Matematika. Mekhanika. Informatika. 2022. Vyp. 1 (42). C. 23-40.

13. Benks A., Porseillo E. Sovremennye shablony dlya razrabotki prilozhenij. Sankt-Peterburg. Peter. 2022. 320 s.

Ispol'zovanie Python i Excel dlya obrabotki i analiza dannyh. URL: <https://habr.com/ru/companies/otus/articles/331998/> (Data obrashcheniya: 20.04.2025).

15. openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files. URL: <https://openpyxl.readthedocs.io/en/stable/> (Data obrashcheniya: 20.04.2025).

16. DockerCompose (2017) Environment variables in compose // Docker Documentation. URL: <https://docs.docker.com/compose/environment-variables/> (Data obrashcheniya: 20.04.2025).

17. Chiganov, D.R. Docker klyuch k kontejnerizacii i masshtabiruemosti. // Vestnik nauki. 2023. Т. 4 (64). №7. С. 270-272.

18. React: biblioteka dlya sozdaniya pol'zovatel'skih interfejsov. URL: <https://ru.legacy.reactjs.org/> (Data obrashcheniya: 20.04.2025).

19. Quick Start – React. // Documentation. URL: <https://react.dev/learn> (Data obrashcheniya: 25.03.2025).

20. Dakett, D. HTML i CSS. Razrabotka i sozdanie veb-sajtov. Moskva. Eksmo. 2020. 480 s.

21. Pishem FastAPI s nulya. URL: <https://habr.com/ru/articles/708678/> (Data obrashcheniya: 20.04.2025).

22. FastAPI documentation – DevDocs // FastAPI Documentation. URL: <https://devdocs.io/fastapi/> (Data obrashcheniya: 20.04.2025).

23. Koss E. N., Beleevskij K. A., Vaseckij A.M. Sozdanie servernoj chasti dlya avtomatizacii formirovaniya raspisaniya universiteta. // Uspuki v himii i himicheskoy tekhnologii. 2021. Т. 35. №10. С. 83-85.

24. PostgreSQL: The World's Most Advanced Open Source Relational Database. // Documentation. URL: <https://www.postgresql.org/> (Data obrashcheniya: 11.01.2025).

25. PostgreSQL // Documentation. URL: <https://www.postgresql.org/docs/> (Data obrashcheniya: 11.01.2025).

26. FET Free Timetabling Software // FET Documentation. – URL: <https://lalescu.ro/liviu/fet/download.html> (Data obrashcheniya: 02.03.2025).

## Информация об авторах

Алла Григорьевна Кравец, доктор техн. наук, профессор, профессор кафедры систем автоматизированного проектирования и поискового конструирования Волгоградского государственного технического университета, Российской Федерации, Волгоград, пр. Ленина, 28, профессор кафедры системного анализа и управления Государственного университета Дубна, Российской Федерации, Московская область, Дубна, ул. Университетская, 19; Alla G. Kravets, Doctor of Technical Sciences, Professor, Professor of the Department of Computer-Aided Design and Search Design, Volgograd State Technical University, Russian Federation, Volgograd, pr. Lenina, 28, Professor of the Department of Systems Analysis and Management, Dubna State University, Russian Federation, Moscow region, Dubna, Universitetskaya str., 19, allagkravets@yandex.ru

Наталья Анатольевна Сальникова, канд. техн. наук, доцент, доцент кафедры информационных систем и математического моделирования, Волгоградский институт управления – филиал Российской академии народного хозяйства и государственной службы при Президенте РФ, Российской Федерации, Волгоград, ул. Гагарина, 8; Natalia A. Salnikova, Ph. D., Associate Professor, Associate Professor of the Department of Information Systems and Mathematical Modeling, Volgograd Institute of Management - branch of the Russian Academy of National Economy and Public Administration under the President of the Russian Federation, Russian Federation, Volgograd, st. Gagarina, 8, ns3112@mail.ru

Инна Владимировна Струкова, главный специалист кафедры систем автоматизированного проектирования и поискового конструирования, Волгоградский государственный технический университет, Российской Федерации, Волгоград, пр. им. Ленина, 28; Inna V. Strukova, Chief Specialist of the Department of Computer-Aided Design and Search Design, Volgograd State Technical University, Russian Federation, Volgograd, pr. im. Lenina, 28, inna-strukova@yandex.ru