



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____

КАФЕДРА _____

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
НА ТЕМУ:
«Информационная система поддержки
профессионально образования»

Студент _____
(Группа)

(Подпись, дата)

(И.О.Фамилия)

Руководитель ВКР

(Подпись, дата)

(И.О.Фамилия)

Консультант

(Подпись, дата)

(И.О.Фамилия)

Консультант

(Подпись, дата)

(И.О.Фамилия)

Нормоконтролер

(Подпись, дата)

(И.О.Фамилия)

2019 г.

РЕФЕРАТ

Квалификационная работа «Информационная система сопровождения профессионального образования» посвящена разработке веб-приложения, упрощающего процесс организацию образовательного процесса и коммуникации между преподавателями и студентами. Система позволяет назначать занятия, оставлять сопровождающие материалы и ставить студентам оценки полученные ими на занятии.

Система представляет собой клиент-серверное приложение, где клиентом выступает браузер пользователя. Система предоставляет возможность одновременного подключения большого количества пользователей.

СОДЕРЖАНИЕ

РЕФЕРАТ	2
СОДЕРЖАНИЕ.....	3
ВВЕДЕНИЕ	5
1 КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ.....	6
1.1 Общетеchnическое обоснование разработки.....	6
1.1.1 Постановка задачи проектирования	6
1.1.2 Описание предметной области.....	6
1.1.3 Перечень процессов подлежащих автоматизации	7
1.1.4 Функциональные задачи системы.....	7
1.1.5 Выбор и обоснование критериев качества.....	8
1.1.6 Анализ аналогов.....	10
1.2 Разработка программного изделия	15
1.2.1 Выбор СУБД	15
1.2.2 Выбор HTTP сервера.....	16
1.2.3 Выбор средств разработки клиентской части системы	17
1.2.4 Выбор средств разработки серверной части системы	18
1.2.5 Архитектура системы.....	19
1.2.6 Проектирование базы данных	20
1.2.6.1 Инфологическая модель.....	20
1.2.6.2 Дatalogическая модель	28
1.2.7 Выбор программных средств	33
1.2.8 Выбор аппаратных средств.....	34
1.3 Разработка интерфейса взаимодействия с пользователем	35
1.3.1 Интерфейс взаимодействия со студентом.....	35
1.3.2 Интерфейс взаимодействия с преподавателем	35
1.3.3 Интерфейс взаимодействия с модератором	35
2 ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ. ВЫБОР HTTP-СЕРВЕРА.....	36
2.1 Метод максимума взвешенной суммы	38

2.2 Метод близости к идеалу	39
2.3 Метод ранжирования Борда	39
2.4 Метод гарантированного результата	39
ЗАКЛЮЧЕНИЕ.....	40
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	41
ПРИЛОЖЕНИЕ А	42

ВВЕДЕНИЕ

Квалификационная работа на тему «Информационная система сопровождения профессионального образования» посвящена разработке системы, позволяющей организовать и значительно упростить коммуникацию между преподавателями и студентами в рамках образовательного процесса.

По состоянию на 2016 год в России функционирует около 3,3 тыс. образовательных организаций среднего профессионального образования и около 950 образовательных организаций высшего профессионального образования.

В большей части образовательных организациях не существует централизованной системы коммуникации между студентами и преподавателями. Часто общение между ними осуществляется через электронную почту или средства обмена сообщениями.

Необходимость такой системы обусловлена тем, что объём информационного потока, проходящий через человека, увеличивается с каждым годом. Также, с ростом информационного потока увеличивается объёмы образовательных материалов, которые необходимо усваивать студентам.

Информационная система сопровождения профессионального образования «SkillArea» решает задачи создания единого информационного образовательного пространства, упрощения организации образовательного процесса и централизации обмена информацией между его участниками.

1 Конструкторско-технологическая часть

1.1 Общетеchnическое обоснование разработки

1.1.1 Постановка задачи проектирования

Целью данной работы является проектирование и реализация информационной системы, состоящей из серверной и клиентской части. Предусмотреть возможность работы большого количества пользователей и отказоустойчивость системы.

Информационная система сопровождения профессионального образования должна предоставлять возможность преподавателям размещать организационную информацию, образовательные материалы и расписание занятий, а также вести журнал оценок.

Таким образом, задачи проектирования могут быть сформулированы следующим образом:

- исследование предметной области;
- разработка инфологической и даталогической моделей данных.
- разработка структуры системы.
- разработка алгоритмов основных функций системы.
- разработка интерфейса системы.
- программная реализация.
- тестирование системы.
- разработка документации к системе.

1.1.2 Описание предметной области

Предметной областью данной информационной системы является совокупность сущностей, объединенных общим образовательным процессом.

Основными действующими лицами в этом процессе являются студенты, преподаватели и администрация образовательного учреждения. Исходя из этого

можно выделить три основные группы пользователей, имеющих различные потребности и функциональные возможности:

- 1) Студент – пользователь системы, который может просматривать материалы предметов, расписание занятий и текущую успеваемость
- 2) Преподаватель – пользователь системы, который может назначать занятия, размещать материалы, связанные с занятием и ставить оценки студентам за активность на занятиях или выпаленные задания.
- 3) Модератор – пользователь системы, имеющий доступ ко всей информации, размещённой в ней. Кроме того, он способен редактировать данные групп и отдельных студентов

1.1.3 Перечень процессов подлежащих автоматизации

При создании информационной системы автоматизации подлежат следующие процессы:

- добавление/редактирование информации о студентах;
- добавление/редактирование информации о группах студентов;
- добавление/редактирование информации о преподаваемых предметах;
- добавление/редактирование информации о проводимых занятиях;
- добавление/редактирование общих новостей;
- добавление/редактирование баллов за занятия;
- добавление/редактирование образовательных материалов;
- формирование расписания;
- формирование статистики о успеваемости.

1.1.4 Функциональные задачи системы

В информационной системе все пользователи должны иметь возможность и право:

- регистрироваться в системе;
- вносить и редактировать данные о себе;

- просматривать общую организационную информацию.

Помимо общих возможностей пользователь со статусом «Студент» должен иметь возможность и право:

- просматривать информацию и учебные материалы, связанные с занятиями по тем предметам, которым он обучается;
- просматривать текущее расписание занятий;
- просматривать свою текущую успеваемость.

Помимо общих возможностей пользователь со статусом «Преподаватель» должен иметь возможность и право:

- назначать занятия и оставлять сопровождающие материалы;
- размещать общую организационную информацию;
- ставить и редактировать оценки студентов, полученные ими на занятии.

Помимо общих возможностей пользователь со статусом «Модератор» должен иметь возможность и право:

- добавлять, редактировать и удалять группы студентов;
- редактировать данные студентов;
- добавлять и удалять предметы;
- размещать, редактировать и удалять общую организационную информацию;
- назначать и отменять занятия;
- добавлять, редактировать и удалять материалы занятий;
- ставить, редактировать и удалять оценки студентов.

1.1.5 Выбор и обоснование критериев качества

Качество — это совокупность характеристик объекта, имеющая отношение к его способности удовлетворить установленные и предполагаемые требования потребителя. Под объектом качества может пониматься как собственно продукция (товары или услуги, информационная технология),

процесс ее производства, так и производитель (организация, организационная структура или даже отдельный работник).

Качество информационной системы — обобщенная положительная характеристика системы, выражающая степень ее полезности пользователю.

Для проектируемого программного комплекса приоритетными являются следующие критерии качества:

- функциональность;
- надежность;
- практичность;
- эффективность;
- сопровождаемость;
- мобильность.

Функциональность характеризует способность информационной системы обеспечивать функции, соответствующие предметной области и удовлетворяющие установленным потребностям пользователей в заданных условиях.

Надежность характеризует устойчивость системы к большим нагрузкам, а также к обработке заведомо неверной или поддельной информации. Данное требование гарантирует наличие в системе только корректных данных. При возникновении всякого рода ошибок система должна оповещать пользователя о них и принимать возможные попытки по их устранению.

Практичность характеризует простоту и понятность работы с системой. Обеспечение удобства интерфейса позволяет минимизировать время, затрачиваемое при работе с системой и получать максимальную эффективность работы.

Эффективность включает требования к временным характеристикам. Можно определить как отношение получаемых с помощью информационной системы результатов к затрачиваемым на это ресурсам всех типов.

Сопровождаемость характеризует удобство и простоту администрирования, возможность расширения функционала и документированность системы.

Мобильность характеризует способность информационной системы к масштабированию или переносу на новое аппаратное обеспечение. Также важной характеристикой системы является её устойчивость к изменению конфигурации или архитектуры аппаратного обеспечения.

1.1.6 Анализ аналогов

Первые системы управления обучением создавались под конкретный проект и, чаще всего, той же организацией (либо её дочерней структурой), в которой затем использовались.

С появлением специализированных компаний-поставщиков системы управления обучением получили развитие универсальные решения, использовавшиеся в различных отраслях. Следующим этапом развития рынка стала специализация системы управления обучением по направлениям использования, с реализацией функционала под соответствующие требования.

Наиболее распространёнными аналогами являются:

- система управления курсами «Moodle»;
- система управления обучением «Canvas».

Критерии сравнения приведены в таблице 1.

Таблица 1 – Критерии сравнения

Критерий	Обозначение	Описание
Функциональность	K1	Способность удовлетворить установленные потребностям пользователей в заданных условиях
Надежность	K2	Способность информационной системы поддерживать определенную работоспособность в заданных условиях

Практичность	K3	Свойство информационной системы, характеризующееся сложностью ее понимания, изучения и использования, а также привлекательность для пользователя при применении в указанных условиях
Эффективность	K4	Свойство системы обеспечивать требуемую производительность с учетом количества используемых вычислительных ресурсов в установленных условиях
Сопровождаемость	K5	Приспособленность информационной системы к модификации и изменению конфигурации, а также наличие документации
Мобильность	K6	Приспособленность информационной системы к внедрению системы в новой аппаратно-операционной среде.

Варианты систем приведены в таблице 2.

Таблица 2 – Варианты сравнения

Вариант	Обозначение	Описание
Система управления курсами «Moodle»	B1	Свободное (распространяющееся по лицензии GNU GPL) веб-приложение, предоставляющее возможность создавать сайты для онлайн-обучения. Система модернизируется и поддерживается с 2001 года по настоящий момент.
Система управления обучением «Canvas»	B2	Коммерческое (распространяющееся по лицензии GNU AGPLv3) веб-приложение с открытым исходным кодом, разработанная и поддерживаемая компанией «Instructure Inc».

		Система модернизируется и поддерживается с 2010 года по настоящий момент.
АИС «SkillArea»	ВЗ	Веб-приложение, предоставляющее единое информационное пространство для организации образовательного процесса и коммуникации между его участниками

В таблице 3 приведён перевод качественных характеристик в количественные. Эффективность, приведённая в таблице была рассчитана на производительность эквивалентную одному ядру 64х битного процессора архитектуры x86 с частотой 2,0 ГГц и оперативной памяти объёмом 2 Гб и частотой 1,6 ГГц.

Таблица 3 – Перевод качественных характеристик в количественные

Значение	Функциональность	Надёжность	Практичность
1	Полное соответствие требованиям	Работоспособность при максимально возможных нагрузках	Интуитивно понятный интерфейс
0,75	Соответствие требованиям выше 90%	Работоспособность при нагрузках значительно выше среднего	Время освоения основного функционала менее 20 минут
0,5	Соответствие требованиям выше 80%	Работоспособность при средних нагрузках	Время освоения основного функционала менее 1 часа
0,25	Соответствие требованиям выше 50%	Работоспособность при минимальных нагрузках	Не требуются специальные знания для освоения

0	Несоответствие требованиям	Отсутствие работоспособности при минимальных нагрузках	Требуются специальные знания для освоения
---	----------------------------	--	---

Значение	Эффективность	Сопровождаемость	Мобильность
1	Способность обработать более 50 запросов в секунду	Использованы актуальные средства разработки, документация на русском или английском языке	Автоматизированное внедрение системы занимает менее 20 минут
0,75	Способность обработать более 25 запросов в секунду	Использованы не актуальные средства разработки, документация на русском или английском языке	Автоматизированное внедрение системы занимает менее 2х часов
0,5	Способность обработать более 10 запросов в секунду	Использованы не актуальные средства разработки, присутствует документация	Автоматизированное внедрение системы занимает более 2х часов
0,25	Способность обработать более 5 запросов в секунду	Использованы устаревшие средства разработки, присутствует документация	Ручное внедрение занимает менее суток

0	Не способность обработать более 5 запросов в секунду	Прочее	Ручное внедрение занимает более суток
---	--	--------	---

Таблица 4. – Сравнение аналогов и прототипов с учетом весовых коэффициентов

Критерий	Коэффициент	B1	B2	B3
K1	0,2	1	1	1
K2	0,15	0,75	1	1
K3	0,2	0,5	0,75	1
K4	0,2	1	0,5	0,75
K5	0,15	0,75	1	1
K6	0,1	0,25	1	1
Итог		0,75	0,85	0,95

АИС «SkillArea» набрал наибольший коэффициент, а значит выполняет поставленные задачи лучше остальных, поэтому разработка системы является целесообразной.

1.2 Разработка программного изделия

Для успешного создания и функционирования системы необходимы следующие компоненты:

- База данных и СУБД. Необходимы для хранения всех данных системы и её пользователей;
- Сервер приложения. Необходим для исполнения бизнес-логики программного изделия;
- HTTP сервер. Необходим для получения запросов от клиентов и отдачи статических файлов;
- Клиентское приложение. Необходима для взаимодействия с пользователем

1.2.1 Выбор СУБД

Для выбора СУБД, используемой в системе, был проведен сравнительный анализ наиболее популярных продуктов по следующим критериям:

- Стоимость использования
- Поддержка технологии клиент-сервер
- Наличие средств администрирования
- Широкое распространение и поддержка

Результаты сравнения вариантов СУБД представлены в таблице 6.

Таблица 5 – Сравнительный анализ СУБД

	Microsoft SQL server	PostgreSQL	MySQL	SQLite
Стоимость использования	От 60 тыс. рублей	Бесплатно	Бесплатно	Бесплатно

Поддержка технологий клиент-сервер	да	да	да	нет
Наличие средств администрирования	да	да	да	да
Широкое распространение и поддержка	да	да	да	да

Несмотря на то, что при сравнительном анализе СБУД «MySQL» и «PostgreSQL» показали себя равными, мной была выбрана база данных PostgreSQL как наиболее знакомая в использовании

1.2.2 Выбор HTTP сервера

В качестве HTTP-сервера был выбран nginx поскольку является свободно распространяемым продуктом и использует асинхронную систему ввода-вывода информации, что на практике означает экономию ресурсов системы и выгодно сказывается в случае больших нагрузок, обладает встроенной устойчивостью к многим видам DoS и DDos атак. Сравнительный анализ nginx и некоторых других HTTP-серверов приводится в исследовательской части.

Описание nginx.

Nginx — это HTTP-сервер и обратный прокси-сервер, почтовый прокси-сервер, а также TCP/UDP прокси-сервер общего назначения.

В nginx рабочие процессы обслуживают одновременно множество соединений. Рабочие процессы выполняют цикл обработки событий от дескрипторов. Полученные от клиента данные разбираются с помощью конечного автомата. Разобранный запрос последовательно обрабатывается цепочкой модулей, задаваемой конфигурацией. Ответ клиенту формируется в буферах, которые хранят данные либо в памяти, либо указывают на отрезок

файла. Буферы объединяются в цепочки, определяющие последовательность, в которой данные будут переданы клиенту. Для чтения и отправки файлов клиенту используются отдельные потоки или системные вызовы (в зависимости от операционной системы).

Конфигурация HTTP-сервера nginx разделяется на виртуальные серверы (директива `server`). Виртуальные серверы разделяются на локации (`location`).

Для эффективного управления памятью nginx использует пулы. Пул — это последовательность предварительно выделенных блоков динамической памяти. Длина блока варьируется от 1 до 16 килобайт.

Nginx содержит модуль географической классификации клиентов по IP-адресу. В его основу входит база данных соответствия IP-адресов географическому региону.

Nginx является одним из немногих серверов, который решил проблему 10 тысяч одновременных соединений.

1.2.3 Выбор средств разработки клиентской части системы

В качестве языка программирования клиентской части системы был выбран «JavaScript» как наиболее гибкий и распространённый. Для разработки пользовательского интерфейса была выбрана библиотека «React» поскольку является наиболее популярным и гибким инструментом для данной решения данной задачи на языке JavaScript.

Описание JavaScript.

JavaScript (часто сокращенно JS) - динамический, слабо типизированный, прототипный и мультипарадигмальный язык программирования. JavaScript наряду с HTML и CSS является одной из трех основных технологий World Wide Web. JavaScript позволяет создавать интерактивные веб-страницы и, таким образом, является неотъемлемой частью веб-приложений. Для исполнения программного обеспечения написанного на JavaScript используется среда Node.js или web-браузер.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

JavaScript имеет компактный, но очень гибкий синтаксис. Сообществом разработчиков написано большое количество инструментов, которые предоставляют доступ к большому количеству дополнительных функций.

Описание React.

React — JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. React может использоваться для разработки одностраничных и мобильных приложений. Его цель — предоставить высокую скорость, простоту и масштабируемость.

React использует виртуальный DOM (англ. virtual DOM), он создает кэш структуру в памяти, что позволяет вычислять разницу между предыдущим и текущим состояниями интерфейса для оптимального обновления DOM браузера. Таким образом программист может работать со страницей, считая, что она обновляется вся, но библиотека самостоятельно решает, какие компоненты страницы необходимо обновить.

1.2.4 Выбор средств разработки серверной части системы

В качестве языка программирования серверной части был выбран Python. Для разработки серверного приложения был выбран фреймворк Django, поскольку он является наиболее функциональным и распространенным инструментом для решения данной задачи на языке Python.

Описание Python

Python — это интерпретируемый язык программирования общего назначения. Python имеет философию разработки, которая нацелена на хорошую читаемость кода, особенно используя значительные отступы. Он предоставляет конструкции, которые обеспечивают четкое программирование как в малых, так и в больших масштабах. Python имеет динамическую типизацию и автоматическое управление памятью. Он поддерживает несколько парадигм

программирования, в том числе объектно-ориентированную, императивную, функциональную и процедурную, и имеет большую и всеобъемлющую стандартную библиотеку.

Описание Django

Django – свободный фреймворк для веб-приложений на языке Python, реализующий шаблон проектирования MVC.

В составе Django присутствует собственный веб-сервер для разработки. Сервер автоматически определяет изменения в файлах исходного кода проекта и перезапускается, что ускоряет процесс разработки на Python, но пригоден только для процесса разработки и отладки приложения.

Некоторые возможности Django:

- ORM с поддержкой транзакций для доступа к БД;
- Встроенный интерфейс администратора, с уже имеющимися переводами на многие языки;
- Диспетчер URL на основе регулярных выражений;
- Система кеширования;
- Встроенная автоматическая документация по тегам шаблонов и моделям данных, доступная через административное приложение.

1.2.5 Архитектура системы

В качестве архитектуры была выбрана трёхуровневая архитектура клиент-серверная архитектура. Она включает себя клиентскую часть, серверную часть и базу данных.

Клиентом системы выступает браузер пользователя, на котором запускается приложение. Клиентское приложение выполняет роль пользовательского интерфейса системы и осуществляет взаимодействие с серверной частью системы.

В серверную часть приложения входит HTTP-сервер и серверное приложение, которые могут располагаться как на одной серверной машине, так

и на разных. Кроме того, HTTP-сервер также может также выступать в роле балансировщика нагрузки.

Хранение данных осуществляется в кластере баз данных. В кластере баз данных обязательно должна присутствовать только одна мастер-реплика, в которую будет осуществляться запись данных. Кроме неё в кластере могут находиться слейв-реплики, с которых будет производиться чтение данных. Количество слейв-реплик ограничено максимальным количеством соединений, которое поддерживается мастер-репликой. Однако стоит учитывать, что каждое соединение замедляет работу сервера мастер-реплики. В случае, когда слейв-реплики отсутствуют чтение данных должно происходить с мастер-реплики.

Архитектура системы показана на рисунке 1.

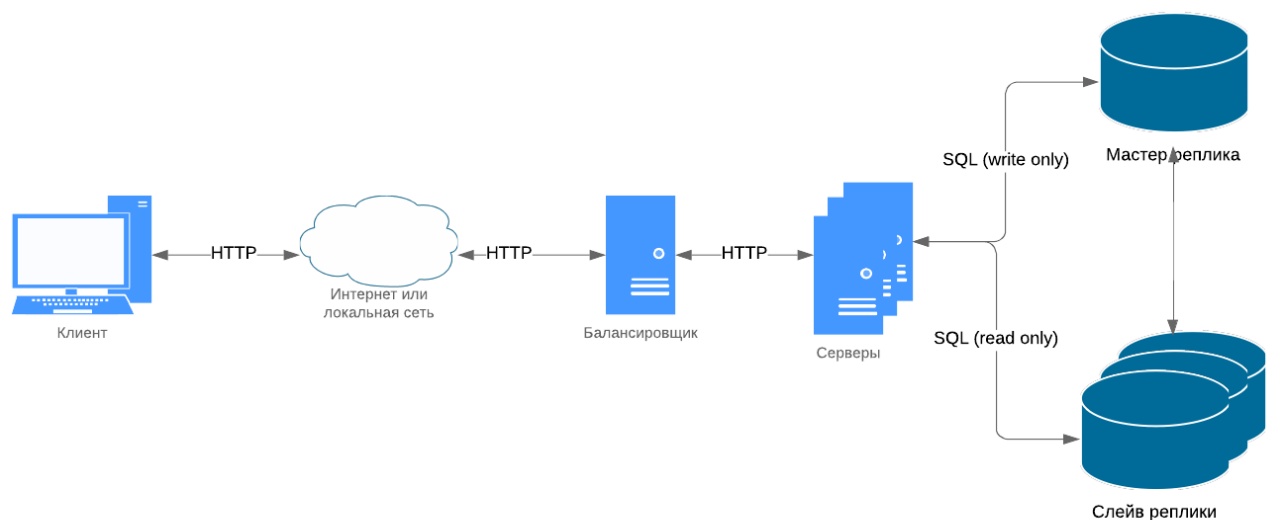


Рисунок 1 — Архитектура информационной системы

1.2.6 Проектирование базы данных

1.2.6.1 Инфологическая модель

Теоретическая часть

На этапе анализа предметной области были выделены основные объекты предметной области, то есть была проведена предварительная структуризация объектов предметной области. Следующим шагом является решение вопроса, какая информация об объектах должна храниться в базе данных и как ее представить с помощью данных. Сущностью концептуального этапа проектирования базы данных является установление соответствия между состоянием предметной области, его восприятием и представлением в базе данных.

Для описания инфологической схемы используются диаграммы «сущность – связь» (ER-диаграмма). Эта схема позволяет моделировать объекты предметной области и отношения между ними. Для ER-диаграмм существуют формальные механизмы преобразования их понятий в понятия реляционной модели данных, поддерживаемой большинством современных СУБД.

Для построения модели типа «сущность–связь» используются три основных конструктивных элемента – сущность, атрибут и связь.

Сущность – поименованная конструкция для моделирования объектов предметной области. Это собирательное понятие, некоторая абстракция реально существующего объекта, процесса или явления, о котором необходимо хранить информацию в базе данных. Каждый экземпляр сущности должен быть уникален. Для этого вводят первичные и вторичные ключи.

Первичный ключ (Primary Key) – это атрибут (или группа атрибутов), однозначно идентифицирующий экземпляр сущности. При этом значение идентифицирующего атрибута находится во взаимнооднозначном соответствии с экземплярами сущности. Первичный ключ должен удовлетворять ряду требований:

Уникальность. Два экземпляра сущности не должны иметь одинаковых значений ключа.

Минимум атрибутов в ключе. Составной первичный ключ (группа атрибутов) не должен содержать ни одного атрибута, удаление которого не приводило бы к утрате уникальности.

Вторичный (или внешний) ключ (Foreign Key) – он появляется в процессе установления связей между сущностями. Это описательный атрибут, который назначается в дополнение к идентифицирующему атрибуту и служит не для идентификации этого уникального экземпляра, а для выделения из набора тех объектов, которые обладают интересующими нас свойствами. Этот ключ часто называют "ключом поиска". Ограничения на состав вторичных ключей определяются логикой запроса

Атрибут – поименованная конструкция для моделирования свойств объекта предметной области (сущности). Основное назначение атрибута – описание свойства сущности, а также идентификация экземпляра сущностей.

Связь – поименованная конструкция для моделирования отношений между объектами (сущностями). При анализе связей между сущностями могут встречаться бинарные (между двумя сущностями), тернарные (между тремя сущностями) и, в общем случае n-арные связи. Может также встречаться унарные (рекурсивные) связи, когда экземпляр определенного типа сущности связан с другим экземпляром той же самой сущности. Наиболее часто встречаются бинарные связи. При проведении классификации видов связей обычно выделяют следующие виды связей:

отображение типа 1:1 (связь "один-к-одному")

отображение типа 1:M (связь "один-ко-многим")

отображение типа M:1 (связь "многие-к-одному")

отображение типа M:M (связь "многие-ко-многим")

Различают зависимые и независимые сущности. Тип сущности определяется связью между сущностями.

Независимая сущность – обычная сущность, которая не зависит при идентификации от других объектов в модели, или просто независимая, если каждый экземпляр сущности может быть однозначно идентифицирован без определения его отношений с другими сущностями.

Зависимая сущность – это дочерняя сущность, уникальность которой зависит от атрибута внешнего ключа, или просто сущность, у которой

однозначная идентификация экземпляра сущности зависит от его отношения к другой сущности.

Зависимые сущности далее классифицируются на сущности, которые не могут существовать без родительской сущности, и сущности, которые не могут быть идентифицированы без использования ключа родителя (сущности, зависящие от идентификации).

Идентифицирующая связь – это связь, которая характеризуется глагольной формой со сказуемым "должен", где подлежащим является имя дочерней таблицы. Она используется, когда требуется, чтобы внешний ключ передавался в дочернюю сущность (и, в результате, создавал зависимую сущность). Идентифицирующая связь устанавливается между независимой (родительской – начало связи) и зависимой (дочерней – конец связи) сущностями.

Очень важно обратить внимание на то, что при установлении идентифицирующей связи атрибуты первичного ключа родительской сущности автоматически переносятся в состав первичного ключа дочерней сущности. Происходит миграция атрибутов. В дочерней сущности новые атрибуты помечаются как внешний ключ (foreign key).

Неидентифицирующая связь – это связь, которая характеризуется глагольной формой со сказуемым "может", где подлежащим является имя дочерней таблицы. Она используется для отображения другого типа передачи атрибутов внешних ключей – передача в область данных дочерней сущности. Очень важно обратить внимание на то, что при установлении неидентифицирующей связи происходит миграция атрибутов первичного ключа родительской сущности не в состав первичного ключа дочерней сущности, а в состав неключевых атрибутов дочерней сущности.

Существенно, что при идентифицирующей связи внешний ключ в дочерней сущности не может принимать значение NULL. Если внешний

ключ должен существовать, то это означает, что запись в дочерней сущности может существовать только при наличии ассоциированной с ним родительской записи.

Сущности и их атрибуты

Таблица 6 – Сущность «Новость»

Логическое имя атрибута	Тип данных	Ключ
Идентификатор	Числовой	Первичный
Заголовок	Текстовый	
Содержание	Текстовый	
Дата добавления	Дата	
Идентификатор автора	Числовой	Внешний

Таблица 7 – Сущность «Группа студентов»

Логическое имя атрибута	Тип данных	Ключ
Идентификатор	Числовой	Первичный
Название	Текстовый	
Дата начала обучения	Дата	

Таблица 8 – Сущность «Дисциплина»

Логическое имя атрибута	Тип данных	Ключ
Идентификатор	Числовой	Первичный
Название	Текстовый	
Сокращённое название	Текстовый	
Идентификатор группы	Числовой	Внешний
Идентификатор преподавателя	Числовой	Внешний

Таблица 9 – Сущность «Занятие»

Логическое имя атрибута	Тип данных	Ключ
Идентификатор	Числовой	Первичный
Название	Текстовый	
Описание	Текстовый	
Дата проведения	Дата	
Место проведения	Текстовый	

Таблица 10 – Сущность «Оценка»

Логическое имя атрибута	Тип данных	Ключ
Идентификатор	Числовой	Первичный
Идентификатор студента	Числовой	Внешний
Идентификатор занятия	Числовой	Внешний
Значение	Числовой	
Дата добавления	Дата	

Таблица 11 – Сущность «Студент»

Логическое имя атрибута	Тип данных	Ключ
Идентификатор	Числовой	Первичный
Идентификатор группы	Числовой	Внешний
Идентификатор пользователя	Числовой	Внешний

Таблица 12 – Сущность «Преподаватель»

Логическое имя атрибута	Тип данных	Ключ
Идентификатор	Числовой	Первичный

Должность	Текстовый	
Идентификатор пользователя	Числовой	Внешний

Таблица 13 – Сущность «Пользователь»

Логическое имя атрибута	Тип данных	Ключ
Идентификатор	Числовой	Первичный
Фамилия	Текстовый	
Имя	Текстовый	
Отчество	Текстовый	
Электронная почта	Текстовый	
Пароль	Текстовый	

Таблица 14 – Сущность «Сессия»

Логическое имя атрибута	Тип данных	Ключ
Идентификатор	Числовой	Первичный
Ключ доступа	Текстовый	
Идентификатор пользователя	Числовой	Внешний
Дата создания	Дата	

Связи между сущностями.

- 1) «Публикует»: связь между сущностями «Пользователь» и «Новость» с типом связи 1:М (связь «один-ко-многим»);
- 2) «Преподаётся»: связь между сущностями «Группа» и «Дисциплина» с типом связи 1:М (связь «один-ко-многим»);

- 3) «Соответствует»: связь между сущностями «Дисциплина» и «Занятие» с типом связи 1:М (связь «один-ко-многим»);
- 4) «Преподаёт»: связь между сущностями «Преподаватель» и «Дисциплина» с типом связи 1:М (связь «один-ко-многим»);
- 5) «Получает»: связь между сущностями «Студент» и «Оценка» с типом связи 1:М (связь «один-ко-многим»);
- 6) «Относится»: связь между сущностями «Занятие» и «Оценка» с типом связи 1:М (связь «один-ко-многим»);
- 7) «Входит»: связь между сущностями «Студент» и «Группа» с типом связи 1:М (связь «один-ко-многим»);
- 8) «Является»: связь между сущностями «Студент» и «Пользователь» с типом связи 1:1 (связь «один-ко-многим»);
- 9) «Является»: связь между сущностями «Преподаватель» и «Пользователь» с типом связи 1:1 (связь «один-ко-многим»);
- 10) «Относится»: связь между сущностями «Сессия» и «Пользователь» с типом связи 1:1 (связь «один-ко-многим»).

Инфологическая модель изображена на рисунке 3.

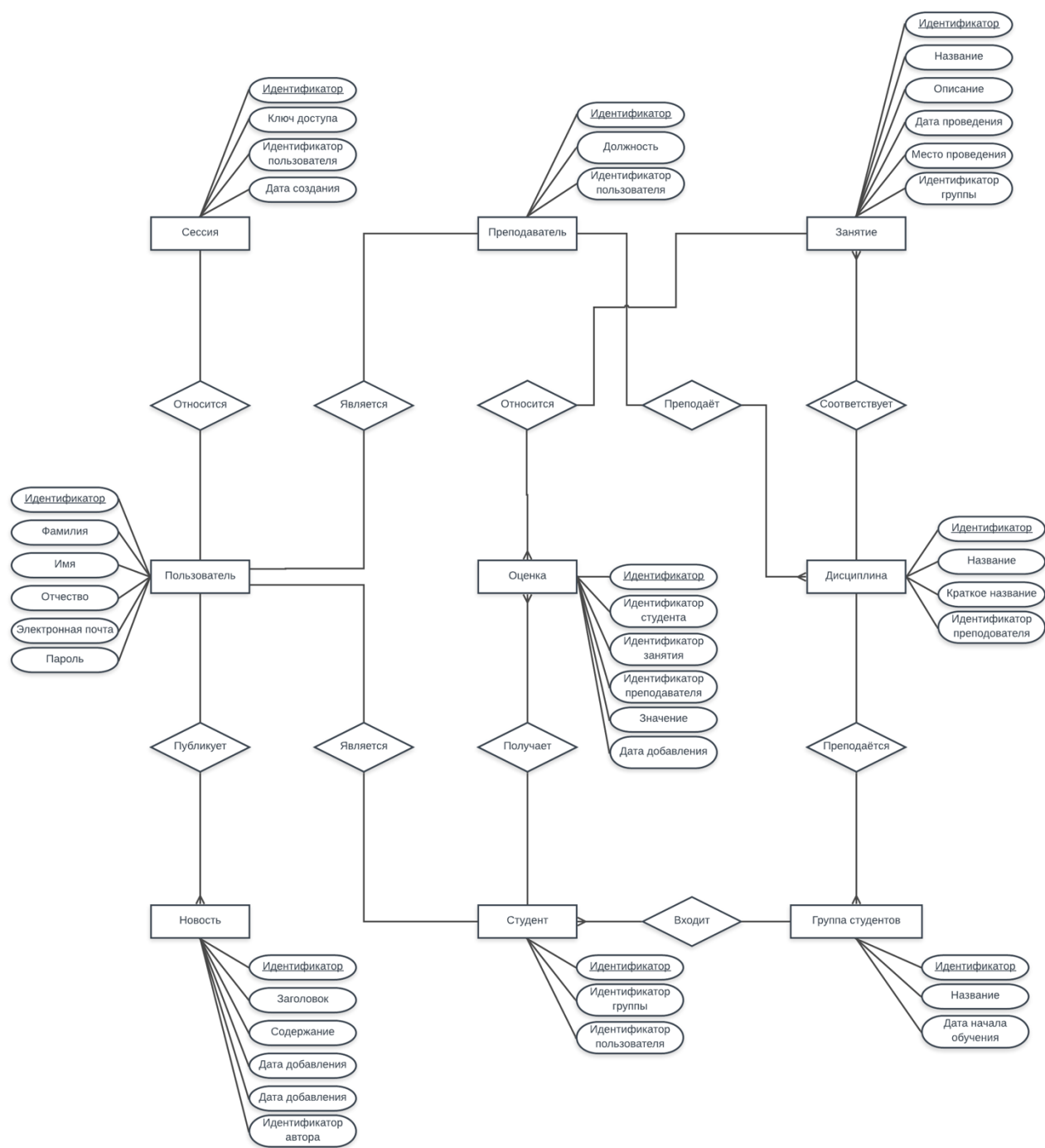


Рисунок 3 — Инфологическая модель

1.2.6.2 Даталогическая модель

Построение даталогической модели основано на разработанной ранее инфологической модели.

Для обеспечения целостности данных в соответствующие сущности инфологической модели добавляются ключевые атрибуты, которые

функционально определены и зависят только от первичного ключа, и не имеют связи между собой. При отображении инфологической модели в даталогическую, сущности соответствуют таблицами, атрибуты — полям таблиц. В даталогическую модель вводятся связи, аналогичные связям инфологической модели.

Таблица 15 – Сущность «news»

Логическое имя атрибута	Поле	Тип данных	Null/Not null	Ключ
Идентификатор	id	int	Not null	Primary Key
Заголовок	title	text	Not null	
Содержание	content	text	Not null	
Дата добавления	created_at	datetime	Not null	
Идентификатор автора	author_id	int	Not null	Foreign Key

Таблица 16 – Сущность «group»

Логическое имя атрибута	Поле	Тип данных	Null/Not null	Ключ
Идентификатор	id	int	Not null	Primary Key
Название	name	text	Not null	
Дата начала обучения	start_at	date	Not null	

Таблица 17 – Сущность «discipline»

Логическое имя атрибута	Поле	Тип данных	Null/Not null	Ключ
Идентификатор	id	int	Not null	Primary Key
Название	name	text	Not null	

Сокращённое название	short_name	text	Null	
Идентификатор группы	group_id	int	Not null	Foreign Key
Идентификатор преподавателя	teacher_id	int	Not null	Foreign Key

Таблица 18 – Сущность «activity»

Логическое имя атрибута	Поле	Тип данных	Null/Not null	Ключ
Идентификатор	id	int	Not null	Primary Key
Название	name	text	Not null	
Описание	description	text	Not null	
Дата проведения	date	datetime	Not null	
Место проведения	place	text	Not null	
Идентификатор дисциплины	disciplin_id	int	Not null	Foreign Key

Таблица 19 – Сущность «mark»

Логическое имя атрибута	Поле	Тип данных	Null/Not null	Ключ
Идентификатор	id	int	Not null	Primary Key
Идентификатор студента	student_id	int	Not null	Foreign Key
Идентификатор занятия	activity_id	int	Not null	Foreign Key
Значение	value	int	Not null	
Дата добавления	date	datetime	Not null	

Таблица 20 – Сущность «student»

Логическое имя атрибута	Поле	Тип данных	Null/Not null	Ключ
Идентификатор	id	int	Not null	Primary Key
Идентификатор группы	group_id	int	Null	Foreign Key
Идентификатор пользователя	user_id	int	Not null	Foreign Key

Таблица 21 – Сущность «teacher»

Логическое имя атрибута	Поле	Тип данных	Null/Not null	Ключ
Идентификатор	id	int	Not null	Primary Key
Должность	position	text	Not null	
Идентификатор пользователя	user_id	int	Not null	Foreign Key

Таблица 22 – Сущность «user»

Логическое имя атрибута	Поле	Тип данных	Null/Not null	Ключ
Идентификатор	id	id	Not null	Primary Key
Фамилия	last_name	text	Not null	
Имя	first_name	text	Not null	
Отчество	patronymic	text	Not null	
Электронная почта	email	text	Not null	
Пароль	password	text	Not null	

Таблица 23 – Сущность «session»

Логическое имя атрибута	Поле	Тип данных	Null/Not null	Ключ
Идентификатор	id	int	Not null	Primary Key
Ключ доступа	access_key	uuid	Not null	
Идентификатор пользователя	user_id	id	Not null	Foreign Key
Дата создания	create_at	datetime	Not null	

Даталогическая модель изображена на рисунке 4.

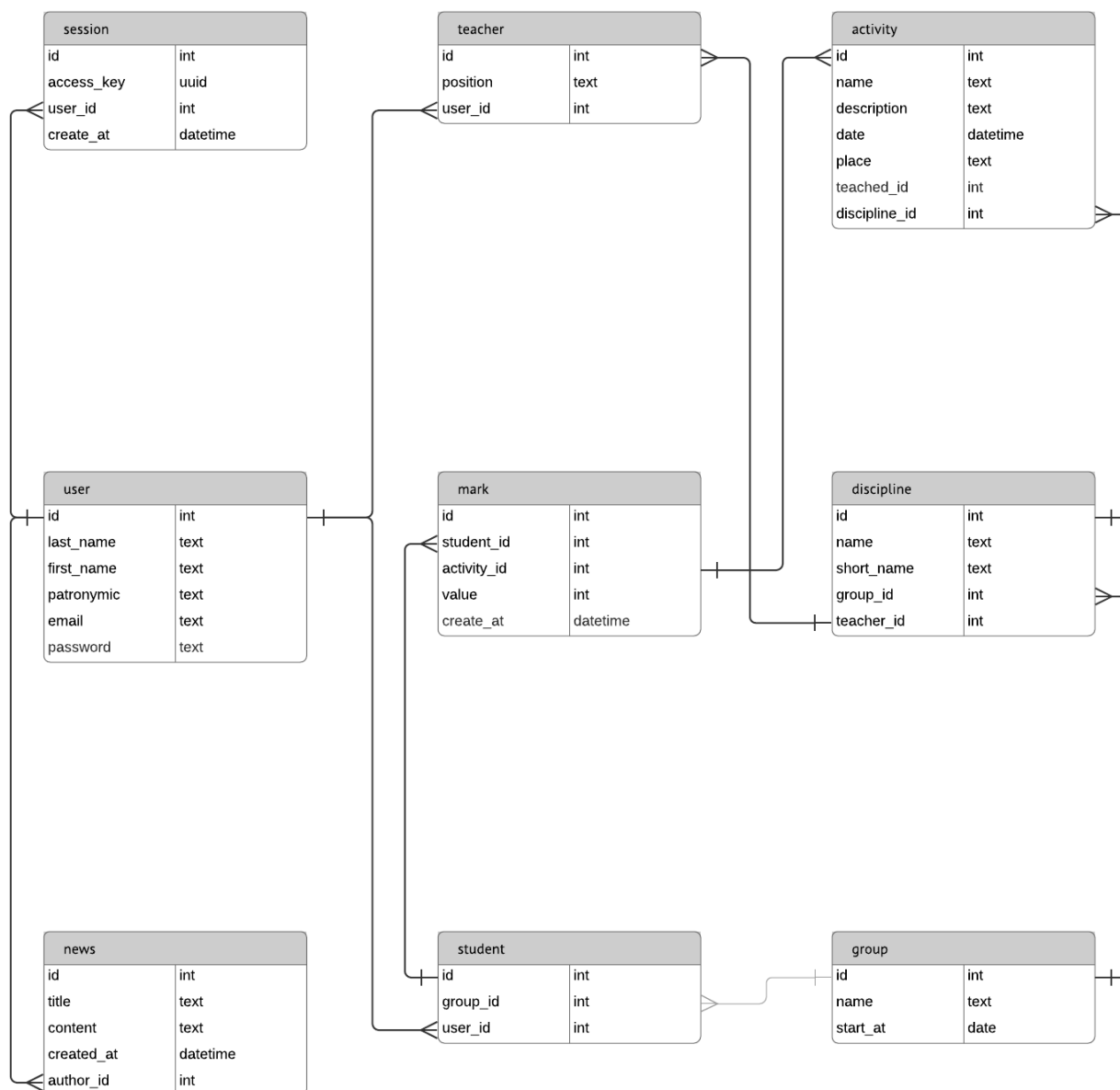


Рисунок 4. Дatalogическая модель

1.2.7 Выбор программных средств

Для корректной работы серверной части данного программного продукта требуется следующее программное обеспечение:

- СУБД PostgreSQL 10.6;
- Веб-сервер NginX;
- Средство разработки Python 3.7;
- Фреймворк для web-сервера Django 2.1;
- Операционная система Ubuntu 16.04;

В качестве операционной системы была выбрана Ubuntu поскольку она имеет следующие преимущества:

- Свободно распространяемая операционная система
- Срок поддержки 5 лет
- Легка в настройке

Для корректной работы системы на компьютере или ноутбуке пользователя требуется следующее программное обеспечение:

- ОС (не регламентируется);
- Веб-браузер Opera 58.0;

1.2.8 Выбор аппаратных средств

Для корректной работы серверной части данного программного продукта требуется следующее минимальное аппаратное обеспечение:

- Процессор Intel Core 2 Quad (с тактовой частотой 2.5 ГГц);
- Оперативную память объемом 8 Гб;
- Жёсткий диск объемом 500 Гб, интерфейс: SATA;
- Сетевая карта с выходом в сеть Интернет.

Для корректной работы системы на компьютере или ноутбуке пользователя требуется следующее минимальное аппаратное обеспечение:

- Процессор Intel Pentium 4 / Athlon 64;
- Оперативную память объемом 512 Мб;
- Свободное место на диске 350 Мб;
- Сетевая карта с выходом в сеть;
- Монитор с разрешением 1365×768;
- Наличие средств ввода вывода.

1.3 Разработка интерфейса взаимодействия с пользователем

1.3.1 Интерфейс взаимодействия со студентом

1.3.2 Интерфейс взаимодействия с преподавателем

1.3.3 Интерфейс взаимодействия с модератором

2 Исследовательская часть. Выбор HTTP-сервера

Одним из наиболее важных элементов информационной системы является HTTP-сервер. Необходимо провести исследование и сравнить наиболее популярные варианты. Участвовавшие в исследовании сервера приведены в таблице N. В тестировании принимали

Таблица N – варианты HTTP-серверов.

Обозначение	Название	Описание
B1	Nginx	Свободно распространяемый HTTP-сервер и обратный прокси-сервер, почтовый прокси-сервер, а также TCP/UDP прокси-сервер общего назначения
B2	Apache	Свободно распространяемый кроссплатформенным веб-сервер. Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т. д. Поддерживает IPv6.
B3	HAProxy	Серверное программное обеспечение для обеспечения высокой доступности и балансировки нагрузки для TCP и HTTP-приложений, посредством распределения входящих запросов на несколько обслуживающих серверов.
B4	Lighttpd	Веб-сервер, разрабатываемый с расчётом на скорость и защищённость, а также соответствие стандартам. Это свободное программное обеспечение, распространяемое по лицензии BSD. lighttpd работает в Linux и других Unix-подобных операционных системах, а также в Microsoft Windows.

Замеры метрик серверов проводилось на тестовом стенде со следующими характеристиками:

- Количество ядер процессора: 2
- Частота процессора: 2,2 ГГц
- Объём оперативной памяти: 4 Гбайт
- Частота оперативной памяти: 1,6 ГГц

Тестирование проводилось на операционной системе Ubuntu. Версии и даты выхода серверов, принимавших участие в тестировании:

- Nginx 1.17.0 (21 мая 2019)
- Apache HTTP Server 2.4.39 (27 марта 2019)
- HAProxy 1.8.10 (22 июня 2018)
- Lighttpd 1.4.53 (27 января 2019)

Замеры использования процессора и оперативной памяти производились при нагрузке в 1000 запросов в секунду и 10 одновременных соединениях.

Тесты состояли из 2х типов запросов:

- 1) Запрос на отдачу статического файла весом 10 Кбайт.
- 2) Запрос, проектирующийся на сервер приложения, вес тела ответа на запрос составлял 1 Кбайт

Критерии для сравнения вариантов приведены в таблице N.

Таблица N – критерии для выбора HTTP-сервера.

Обозначение	Название	Описание
K1	Среднее использование процессора, %	Средний процент использования процессора в тестах
K2	Максимальное использование процессора, %	Процент максимального использования процессора в тестах
K3	Среднее использование памяти, Мб	Средний объём используемой памяти в тестах

K4	Максимальное использование памяти, Мб	Максимальный объем используемой памяти в тестах
K5	Количество запросов в секунду	Среднее количество запросов в секунду, которое может обработать сервер
K6	Поддержка HTTP/2	Поддержка протокола HTTP/2. Протокол HTTP/2 является бинарным. По сравнению с предыдущим стандартом изменены способы разбиения данных на фрагменты и транспортирования их между сервером и клиентом.

Таблица N. Значения вариантов по заданным критериям

	B1	B2	B3	B4
K1	21,73	77,76	41,02	21,98
K2	29,66	91,49	44,23	25,78
K3	2,10	14,52	2,83	1,96
K4	2,11	14,64	2,85	2,09
K5	4213	2164	3547	4256
K6	+	+	+	-

Для выбора лучшего варианта воспользуемся методами:

- Метод анализа иерархии
- Метод максимума взвешенной суммы локальных критериев
- Метод ранжирования Борда
- Метод близости сравниваемых критериев к идеальному
- Метод гарантированного результата

2.1 Метод максимума взвешенной суммы

По методу нахождения максимума взвешенной суммы нужно нормировать значения и рассчитать взвешенные значения для каждого варианта. Рассчитанные значения приведены в таблице N.

Таблица N – Нормированные значения, коэффициенты и взвешенная сумма

	B1	B2	B3	B4	α
K1	1	0,279	0,529	0,989	0,2
K2	0,869	0,282	0,583	1	0,15
K3	0,933	0,134	0,629	1	0,15
K4	0,990	0,143	0,733	1	0,1
K5	0,989	0,508	0,833	1	0,25
K6	1	1	1	0	0,15
	0,966	0,409	0,719	0,848	

Предпочтительность вариантов по таблице N:

$B1 > B4 > B3 > B2$

По методу максимума взвешенной суммы лучшим является **вариант 1**.

2.2 Метод близости к идеалу

2.3 Метод ранжирования Борда

2.4 Метод гарантированного результата

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЕ А