

Реферат

Структура работы представлена следующими разделами:

Введение. В разделе формулируется цель работы, поставленные задачи и необходимость данной работы.

Раздел 1. Анализ предметной области автоматизации образовательных и организационных процессов Университета. Изучение и сравнительный анализ различных подходов в хранении и обработки информации с целью выбора подходящего набора для данной задачи. Изучение и анализ взаимосвязей внутри университета для их последующей автоматизации. Анализ поведения пользователей различных ролей применительно к задачам получения студентом высшего образования. Анализ возможностей инструментального комплекса, для определения возможностей масштабирования.

Раздел 2. Моделирование модели функционирующих зависимостей в университете и разработка на основе этих зависимостей модель базы данных, обеспечивающая автоматизацию большинства процессов в университете. Раздел включает: граф взаимосвязей внутри отделов университета, Концептуальную модель предметной области и Концептуальную модель базы данных.

Раздел 3. Реализация автоматизированной web-системы LMS(Learning Management System). Анализ и обработка инцидентов связанных с введением в эксплуатацию LMS. Налаживание автономной службы технической поддержки. Создание базы данных и запросов к ней, исходя из требований университета.

Объем работы составляет 30 страниц.

В ходе проведение исследования было изучено и проанализировано 17 источников литературы.

Содержание

Реферат.....	2
Содержание.....	3
Введение	4
Раздел 1. Анализ предметной области автоматизации образовательных и организационных процессов Университета	5
1.1. Изучение и анализ взаимосвязей внутри университета для их последующей автоматизации....	6
1.2. Изучение и сравнительный анализ различных подходов в хранении и обработки информации с целью выбора подходящего набора для данной задачи	9
1.3. Организация дистанционного образования.....	Ошибка! Закладка не определена.
Раздел 2. Моделирование модели функционирующих зависимостей в университете	12
2.1. Детальный разбор сущностей необходимых для функционирования системы.	12
2.2. Основные запросы к базе данных..	15
2.3. Разработка на основе этих сущностей модели базы данных, обеспечивающую автоматизацию большинства процессов в университете..	Ошибка! Закладка не определена.
Раздел 3. Реализация автоматизированной web-системы LMS(Learning Management System).	19
3.1. Анализ и обработка инцидентов связанных с введением в эксплуатацию LMS.....	19
3.2. Налаживание автономной службы технической поддержки.....	20
3.3. Создание базы данных и запросов к ней, исходя из требований университета..	20
3.4. Заполнение таблиц. ETL-процессы загрузки базы данных..	22
3.5. Оценка размеров Базы Данных и каждого из файлов..	26
Заключение	28
Список литературы	29

Введение

Автоматизированные информационные системы (АИС) в сфере образования на сегодняшний день становятся незаменимыми частями университетов по всему миру. Огромное количество разнородной информации и необходимость в перенесении всех процессов в дистанционный формат вынуждает каждую образовательную организацию озаботиться об автоматизации потоков данных.

На данный момент существует множество уже готовых АИС специализированных на образовательных процессах каждая из которых обладает своими преимуществами и недостатками. Но если рассматривать сложную узкоспециализированную систему, то для нее пользоваться готовым решением зачастую является непригодным вариантом, потому что такие решения часто не покрывают все специфические особенности образовательного процесса.

В данной работе будут рассмотрены процессы, сущности и взаимодействия между ними в рамках образовательной организации. В качестве решения проблемы автоматизации выбрано создание единой централизованной информационной системы LMS. Этапы моделирования, проектирования, создания и внедрения описаны в рамках данной работы.

Цель исследования заключается в следующем:

Разработать автоматизированную образовательную web-систему LMS, для автоматизации образовательных и организационных процессов внутри университета.

Для достижения поставленной цели сформулированы следующие задачи:

1. Изучить все внутренние взаимодействия внутри университета, проблемы которые существуют при отсутствии автоматизации.
2. Сформировать требования к проектируемой web-системе
3. Спроектировать и реализовать систему для университета.
4. Тестирование системы в эксплуатации на пользователях, сбор обратной связи, исправление ошибок, расширение функциональности по просьбам пользователей
5. Внедрение системы среди студентов, преподавателей и университетских отделов.

Раздел 1. Анализ предметной области автоматизации образовательных и организационных процессов Университета

Университет – высшее учебное заведение, которое: реализует образовательные программы высшего и послевузовского профессионального образования по широкому спектру направлений подготовки (специальностей); осуществляет подготовку, переподготовку и (или) повышение квалификации работников высшей квалификации, научных и научно-педагогических работников; выполняет фундаментальные и прикладные научные исследования по широкому спектру наук; является ведущим научным и методическим центром в области своей деятельности[1]

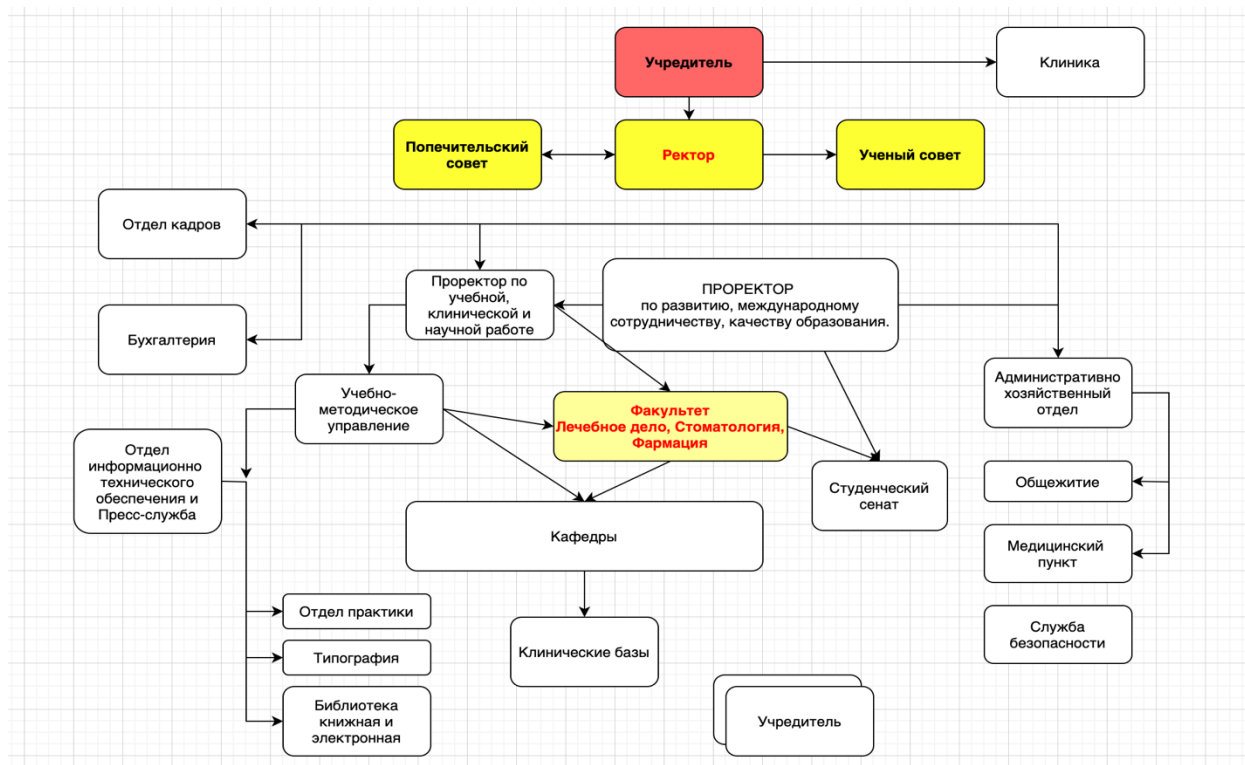
Как ориентир для конечного результата можно рассмотреть системы: Chamilo [2], Moodle [3] и экосистему НИЯУ МИФИ

Система LMS разрабатывается для медицинского университета, который имеет отличительные черты от университетов других специализаций. Во-первых на весь университет существует всего три специальности (Лечебное дело, Стоматология, Фармация) с огромным перевесом студентов в первой из них. Во-вторых при составлении расписания желательно выделять полностью один день в неделю для студентов 2 и старше курсов для прохождения студентами клинической практики.

Без использования автоматизированных информационных систем(АИС) все процессы внутри образовательной организации значительно усложняются. А некоторые из них, как например дистанционное образование в связи с COVID-19, становятся невозможны[4].

1.1. Изучение и анализ взаимосвязей внутри университета для их последующей автоматизации.

Граф зависимостей внутри университета представлен ниже.



Самые важные отделы для построения АИС:

1. Учебно-методическое управление (Учебный отдел).
2. Отдел кадров (ОК).
3. Деканат.
4. Отдел информационно-технического обеспечения (ИТ-отдел).
5. Бухгалтерия.
6. Ректорат.

Ниже приведено их описание и роль в АИС:

1. Учебный отдел. Данный департамент занимается вопросами касающимися учебной деятельности. В их обязанности входит:

- Составление расписания.
- Распределение нагрузки на преподавателей.
- Инструктаж преподавателей.

- Контроль качества занятий.

2. Отдел кадров. Данный департамент внутри себя разделяется на:

2.1 Студенческий отдел кадров (СОК). В их ведении и контроле информация о студентах. В СОК хранятся личные дела студентов. До внедрения LMS единственная информация о студентах была только в личных делах, в распечатанном формате. В обязанности Студенческого отдела кадров входит:

- Выдача справок студентам.
- Хранение, анализ и выдача по запросу данных о студенте.
- Прием и оформление личных дел полученных из Приемной комиссии.

2.1 Отдел кадров сотрудников(ОКС)В их ведении и контроле информация о сотрудниках. В ОКС хранятся личные дела сотрудников. В обязанности Отдела кадров сотрудников входит:

- Направляют в бухгалтерию отчеты об отработанных часах сотрудников, для последующей оплаты.
- Хранение, анализ и выдача по запросу данных о сотруднике.
- Наем, прием, перевод и увольнение сотрудников.

3. Деканат. Отдел занимающийся прямой работой со студентами. В обязанности Деканата входит:

- Руководство учебной, воспитательной и научной работой.
- Решение большинства проблем студентов.
- Распределение студентов по группам.
- Контроль за оплатой студентами обучения.

4. Отдел информационно-технического обеспечения (ИТ-отдел). Отдел занимающийся техническим оборудованием университета и решения мелких технических проблем. В обязанности ИТ-отдела входит:

- Дизайн, разработка и сопровождение сайта визитки.

- Прокладка интернета по всей территории университета.
- Инвентаризация оборудования и аудиторий.

5. Бухгалтерия. Отдел занимающийся всеми финансовыми вопросами в университете. В обязанности бухгалтерии входит:

- Прием платежей за обучение от студентов, с внесением платежей в систему.
- Выплата зарплаты сотрудникам.
- Бухгалтерский учет университета.
- Своевременная выплата налогов.

6. Ректорат. Самый высокий отдел, занимается контролем всех остальных отделов. В обязанности ректората входит:

- Контроль за прохождением университетом аккредитации.
- Формирование отделов и иерархии внутри них.

Рассмотрим как студент связан с остальными отделами. При поступлении в университет студент подает документы в Приемную комиссию. Далее Деканатом составляются списки на зачисление, на этом же этапе Деканат определяет группу студента. Далее документы студента передаются в СОК, где на студента составляется личное дело, создается аккаунт в АИС и вся информация о студенте цифровизируется в его аккаунт (путем заполнения полей и сканирования документов). После этого Учебный отдел составляет расписание для группы, в которой учится студент и контролирует сессию и модульную неделю.

1.2. Изучение и сравнительный анализ различных подходов в хранении и обработки информации с целью выбора подходящего набора для данной задачи.

В данном проекте огромное количество сущностей и связей между ними. Эти сущности задаются однажды и не меняются на всем времени эксплуатации системы. В связи с этим было принято решение о выборе реляционной модели базы данных. Для полей которые могут изменяться в результате деятельности пользователя (документы пользователя, ответы к вопросу теста, кредиты в учебном плане и т.д.) было решено использовать json атрибут соответствующей таблицы.

В качестве системы управления базы данных была выбрана система управления базами данных (СУБД) MySQL

Основные причины выбора СУБД MySQL:

- многопоточность, поддержка нескольких одновременных запросов;
- оптимизация связей с присоединением многих данных за один проход;
- записи фиксированной и переменной длины;
- гибкая поддержка форматов чисел, строк переменной длины и меток времени;
- Наличие удобного Eloquent ORM (PHP Laravel);
- быстрая работа, масштабируемость;
- бесплатна;
- Удобное и быстрое разворачивание MySQL в docker-compose контейнерах;

Подробнее о данной СУБД можно узнать в [5]

Помимо данных в БД есть необходимость хранить файлы пользователей (заявления, аттестаты, паспорта студентов, Учебные материалы, Домашние задания и т.д.).

Рассматривались следующие варианты хранения:

1. Локально (На том же сервере, что и система в одной из директорий).

Преимущества: Независимость от работы интернета, скорость выше чем на облаке, безопасность (файлы не покидают сервер для хранения)

Недостатки: Большая цена на дисковое пространство сервера, дополнительная нагрузка на сервер, вероятность потерять данные при выходе сервера из строя.

2. На облачных серверах (Yandex.cloud)

Преимущества: Независимость от работы сервера, Низкая цена, Убирание нагрузки с сервера

Недостатки: Невозможность работы системы локально (без доступа к сети интернет)

Больше о сравнении этих подходов можно прочитать в [6]. Был выбран подход с облаками, а конкретней Yandex.cloud. Данный сервис нивелирует все недостатки локального подхода и обеспечивает все потребности в хранении данных [9]. Использование облачных технологий в образовательной среде имеет ряд особенностей отраженных в [10].

В качестве средства обработки информации используется язык применяемый для разработки веб-приложений. Рассматривались 2 варианта PHP и Python. Производительность PHP выше [7] чем у Python и на языке PHP написано большое количество удобных веб-фреймворков[8] что и послужило основной причиной выбора PHP как основного языка обработки данных системы.

В качестве фреймворка был выбран Laravel, у него не самая большая производительность среди PHP фреймворков [8], но при этом он изначально рассчитан на расширение функциональности, практически все уязвимости безопасности PHP web-приложений в нем устранены изначально.

1.3. Организация дистанционного образования.

Пандемия COVID-19, повлекла за собой появление острой необходимости в переводе образования в дистанционный формат [11].

Необходимый минимум условий для запуска дистанционного образования это:

1. Выбор стриминговой платформы в которой будут проходить занятия.
2. Обучение преподавателей/студентов пользованию стриминговой платформой.
3. Менеджмент ссылок/id/приглашений от преподавателей студентам

При выполнении всех трех условий университет может начать проводить все занятия дистанционно.

1. Для семинарских занятий была рассмотрена платформа Zoom (с бесплатными аккаунтами 45 минут семинара 5 минут перерыв 45 минут семинара).

Как было написано ранее, в медицинском университете подавляющее количество студентов учатся на одной специальности «Лечебное дело». В связи с этим было принято решение на лекции разбивать потоки по 12-15 групп. В одной группе в среднем содержится 22 студента. Таким образом на одной лекции может максимально находиться 330 студентов. Аудитории в самом университете оборудованы достаточно, чтобы позволять такому количеству студентов очно посещать лекции, не испытывая дискомфорта, но в формате Zoom такие занятия проводить не целесообразно.

Для лекций был выбран формат YouTube трансляций с доступом по ссылке.

2. С обучением студентов пользованию YouTube и Zoom проблем не возникло, студенты сами в краткие сроки обучились, мы же им предоставили инструкцию на двух страницах и у студентов ни разу не возникало проблем с данными сервисами.

С преподавателями было гораздо сложнее. Zoom спустя 1-2 недели освоить смогли все. Изначально обучением преподавателей занимался it-отдел, со временем (когда большинство технических проблем было решено) проблемы преподавателей перешли в Учебный отдел. Большая часть лекций записывается телеканалом, но по нагрузке не все преподаватели могут записывать там. Самостоятельную трансляцию лекций на YouTube большинство преподавателей освоило.

3. Постоянное решение этого пункта – создание и использование автоматизированной динамически изменяемой платформы (LMS).

Раздел 2. Моделирование модели функционирующих зависимостей в университете.

2.1. Детальный разбор сущностей необходимых для функционирования системы.

В университете был выделен следующий набор сущностей:

1. Сущности связанные с курсом и повседневным учебным процессом
 - 1.1. Предмет
 - 1.2. Курс преподавателя по предмету
 - 1.3. Расписание
 - 1.4. Учебный материал
 - 1.5. Домашнее задание
 - 1.6. Результат домашнего задания
 - 1.7. Кафедры
 - 1.8. Завершенное занятие(столбец электронного журнала)
2. Тестовый модуль:
 - 2.1. Шаблон теста
 - 2.2. Вопрос к шаблону теста
 - 2.3. Назначенный тест
 - 2.4. Результаты студента по тесту
3. Пользователи и их объединения:
 - 3.1. Пользователь
 - 3.2. Группа
 - 3.3. Студенческий поток
 - 3.4. Специальности
4. Сущности которыми обладают пользователи:
 - 4.1. Уведомления
 - 4.2. Приказы
 - 4.3. Платежи
 - 4.4. Права доступа

5. Снаряжение и инвентаризация университета:

4.1. Аудитория

4.2. Оборудование

6. Семестры

7. Учебный план

Рассмотрим каждую сущность подробнее:

1. Сущности связанные с курсом и повседневным учебным процессом

1.1. Предмет

Данная сущность представляет собой название курса (пример Анатомия).

Реализация: таблица «course_names»

Связи: Кафедра (предмет закреплен за кафедрой)

Зависят: Курс, Шаблон теста, Учебный план

Атрибуты:

- Название курса (name) varchar(255)
- Id Кафедры (department_id) int foreign key
- Hex код цвета заливки в расписании (color) varchar(8)

1.2. Курс преподавателю по предмету(Курс)

Данная сущность представляет собой назначенный курс по предмету с преподавателем, по конкретному семестру. По сути Курс является сводной таблицей между Пользователями как преподавателями и Предметами, обеспечивающий между ними связь многие-ко-многим.

Реализация: таблица «courses»

Связи: Название курса, Пользователь(Преподаватель курса)

Зависят: Расписание, Учебные материалы, Домашние задания,

Атрибуты:

- Id Предмета (name) int foreign key
- Id Преподавателя (department_id) int foreign key
- Семестр (color) varchar(8) – ограничивает потоки доступные преподавателю на этом курсе, внутри групп, у которых по учебному плану в этом семестре есть данный Предмет.

1.3. Расписание

Данная сущность представляет собой ячейку в таблице расписания у Потока по Курсу в день недели. Составляется и редактируется Учебным отделом.

Реализация: таблица «time_tables»

Связи: Курс, Поток, Семестр, Аудитория

Зависят: Завершенное занятие

Атрибуты:

- Id Курса (course_id) int foreign key
- День недели проведения занятия(week_day) int – Вс = 0, Пн = 1, ..., Сб = 6
- Id Аудитории (classroom_id) int foreign key
- Id Потока (stream_id) int foreign key
- Тип занятия (lesson_type) int 0-Лекция, 1-Семинар
- Порядковый номер занятия (order) int
- Ссылка на проводимое дистанционное занятие (link) int
- Четность (half) int 0-Занятия каждую неделю, 1-по нечетным, 2- по четным
- Id Семестра (semester_id) int foreign key

1.4. Учебный материал

Какой-либо файл или ссылка оставленные преподавателем по курсу, чтобы студенты этого курса могли получить доступ к ним

Связи: Курс

Атрибуты:

- Id Курса по которому загружен материал (course_id) int foreign key
- Название материала (name) varchar
- Тип ресурса (resource_type) int 0-ссылка, 1-файл
- Ресурс (resource) varchar при resource_type==0 ссылка оставленная преподавателем при resource_type==1 название файла на облаке в директории /edu_materials/

2.2. Основные запросы к базе данных.

Для каждой сущности в том или ином виде реализованы все операции CRUD (от английских слов create, read, update, delete) [12].

Помимо них реализованы следующие запросы:

Для пользователя:

- Группа
- Специальность
- Курсы которые преподаются в данном семестре
- Результаты тестов
- Загрузить файл профиля (изменение бд + загрузка файла на облако)
- Ссылки на все файлы профиля
- Проверка доступа по (group->permissions)
- Расписание пользователя (2 разных запроса для студента и преподавателя)
- Вывод электронного журнала
- Вывод уведомлений
- Расчет рейтинга студента

Для группы:

- Специальность
- Пользователи
- Учебные потоки в которых состоит группа
- Курсы которые ведутся у группы
- Тесты которые проводились/проводятся у группы
- Расписание группы
- Журнал группы

Для курса:

- Потоки доступные для этого семестра по учебному плану специальности потока
- Группы назначенные на курс
- Задания по курсу
- Учебные материалы по курсу
- Преподаватель курса

Для шаблона теста:

- Банк вопросов
- Автор шаблона
- Курс по которому проходит тест
- Количество вариантов теста
- Назначенные тесты по этому шаблону

Для вопроса теста:

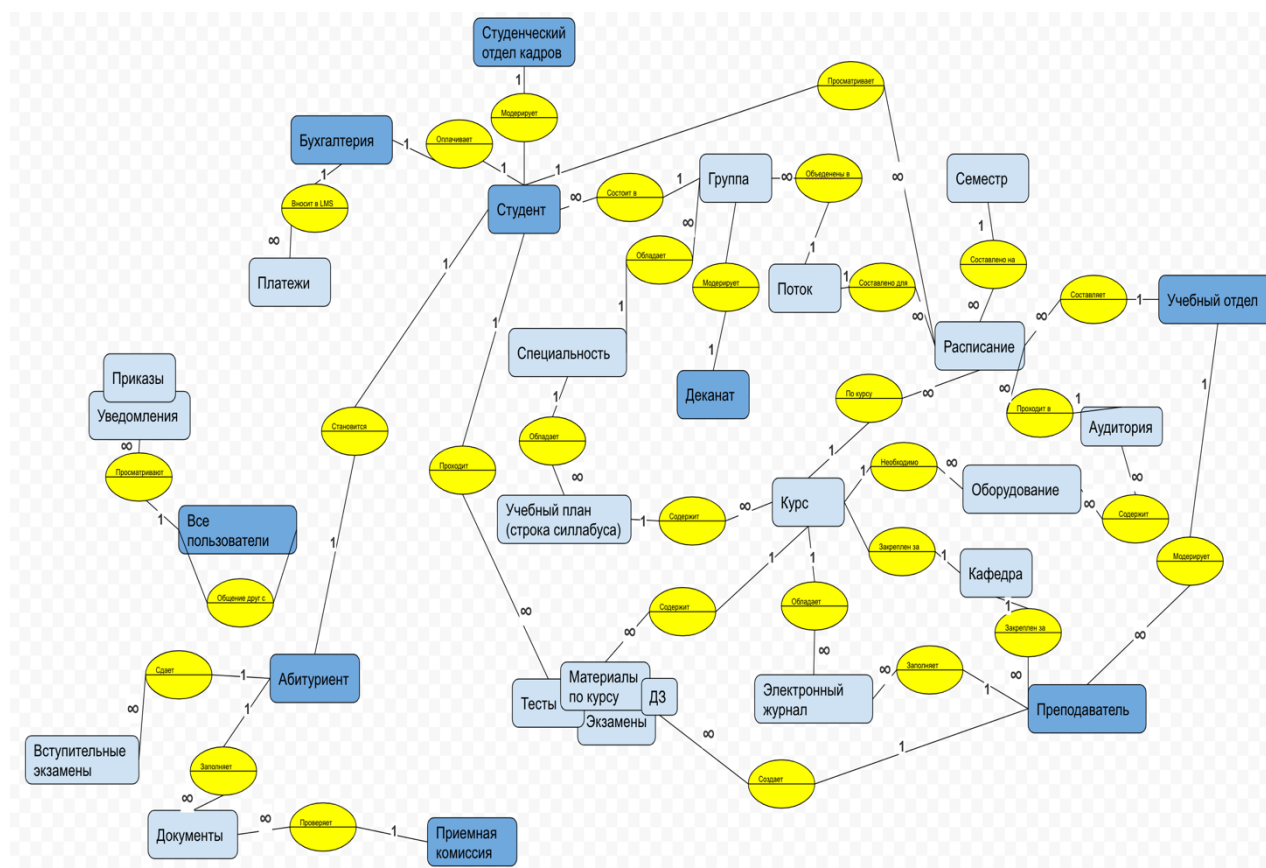
- Загрузить файл приложения к вопросу
- Получить ссылку на файлы вложения
- Шаблон к которому относится вопрос

Для назначенного теста:

- Шаблон, который назначен на этот тест
- Группа, которая проходит этот тест
- Можно ли сейчас начать выполнение этого теста
- Может ли студент пройти этот тест(есть ли в `test_results` у этого студента по этому тесту записей меньше чем попыток теста и подходит ли время)
- Студенты написавшие этот тест
- Результаты этого теста

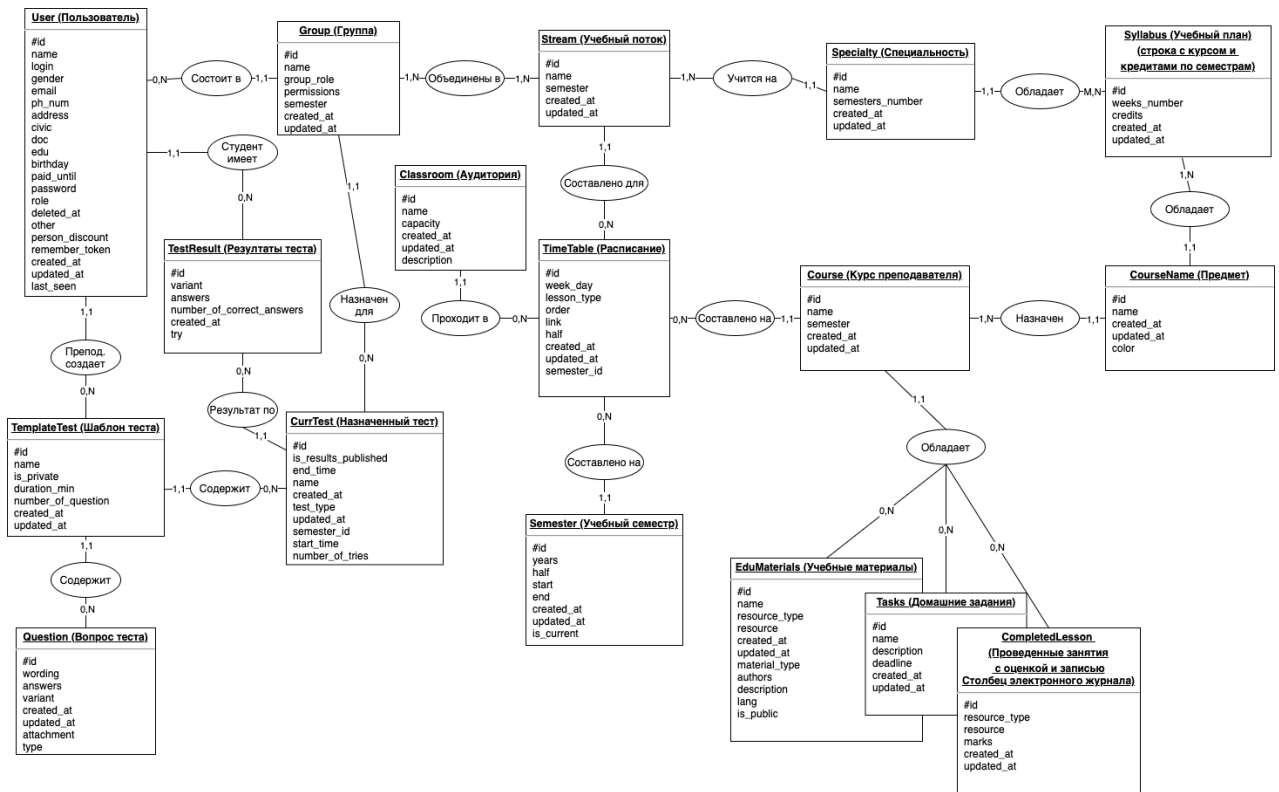
2.2. Разработка на основе сущностей и связей модели базы данных, обеспечивающую выделенные запросы.

На основании выделенных сущностей была построена Концептуальная модель предметной области(КМПО) указанная ниже.

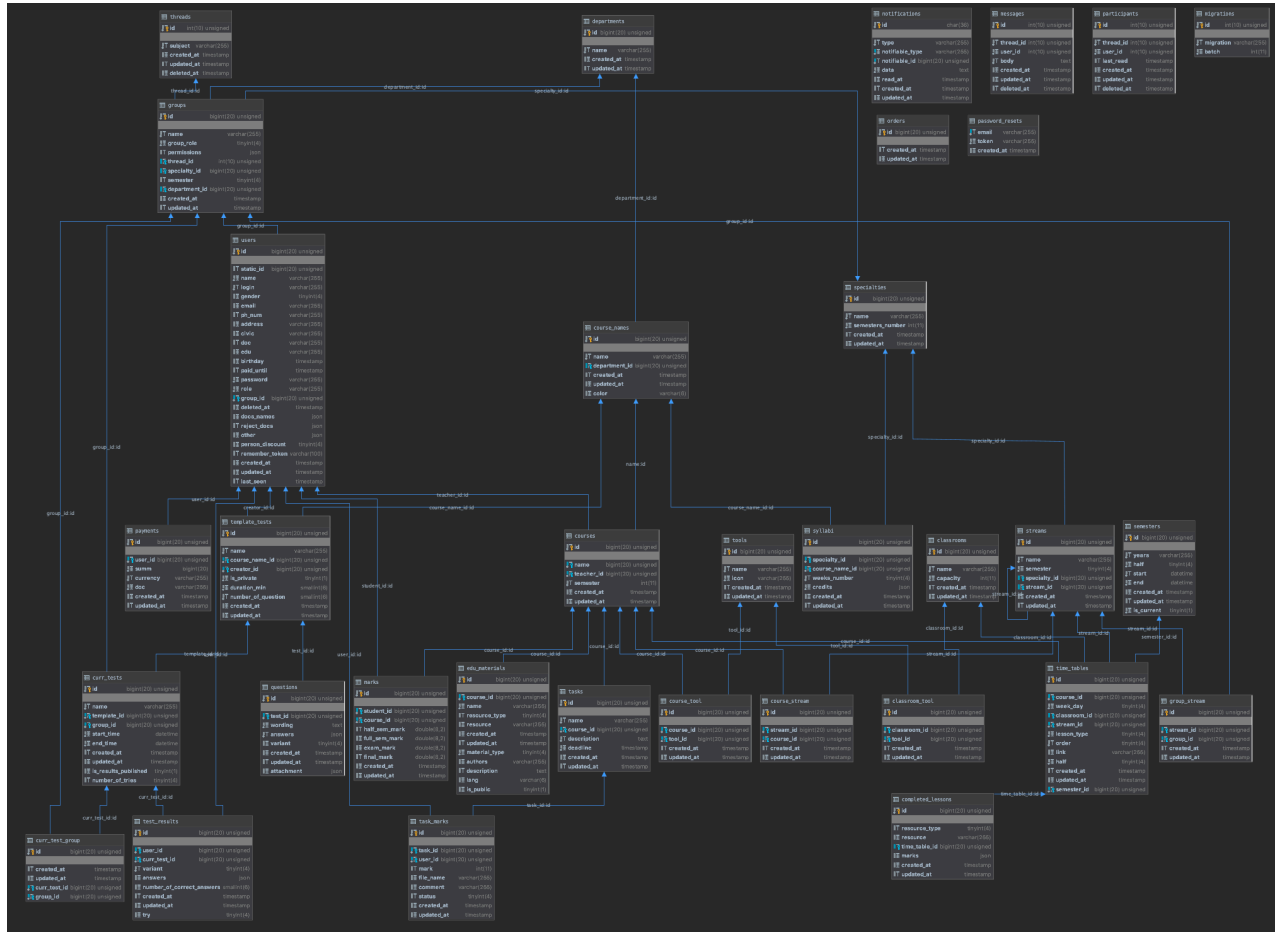


Все субъектные сущности в базе данных были объединены в одну: Пользователь(users). С разграничением доступа, ролей и обязанностей через Группу.

На основании КМПО была спроектирована Концептуальная модель базы данных(КМБД) указанная ниже.



На основании КМБД была построена физическая модель базы данных



Раздел 3. Реализация автоматизированной web-системы LMS(Learning Management System).

3.1. Анализ и обработка инцидентов связанных с введением в эксплуатацию LMS.

Первый ввод в эксплуатацию системы был в июне 2020, тогда первый выпуск университета сдавал государственные экзамены. На тот момент многие концепции были не реализованы и системы была очень недоработана. Тогда был реализован тестовый модуль в его изначальном виде.

После сдачи государственных экзаменов университет перешел в режим приемной комиссии, это и стало основным вектором работы. К концу июня приемная комиссия была реализована, а сотрудники обучены ей пользоваться.

К началу октября система была практически готова к вводу в эксплуатацию, была проблема в том, что расписания на этот университет не было. Последнее чего не было в системе для старта дистанционного образования, так это инструментов мониторинга и составления расписания. Так как расписание составлял я, понимая что мне было бы удобно при составлении, был реализован механизм хранения в базе данных расписания, пользовательский интерфейс для ввода и множество инструментов для просмотра расписания в разных плоскостях и нахождения наложений занятий. В этом же самом процессе было создано расписание на тот семестр. После этого были загружены (группы, студенты, преподаватели), студентам и преподавателям были выданы данные, написаны инструкции и учебный процесс через систему начался.

В декабре в преддверии зимней сессии система была развернута еще на двух серверах для еще одного, в 4 раза больше первого, медицинского университета и медицинского колледжа. На основании обратной связи со студентами и преподавателями в системе произошли изменения (При создании вопроса теста, можно создавать формулы, для пересдачи не нужно назначать новый тест, а просто увеличить количество попыток и перенести дату и т.д.)

3.2. Налаживание автономной службы технической поддержки.

Выдача логинов и паролей студентам осуществлялась через старост группы, а преподавателям через учебный отдел. Неизбежно с появлением пользователей появились и ошибки, как системы, так и пользователей. Было решено организовать линию технической поддержки.

В системе появилось модальное окно с номером телефона и почты технической поддержки.

Благодаря техподдержке было выявлено и устранено огромное количество ошибок в системе. В среднем время от нахождения ошибки до устранения ее на всех серверах организаций занимало не более 3-х часов.

Около 95% обращений приходилось на WhatsApp аккаунт привязанный к аккаунту номера техподдержки, около 1% обращений приходилось на почту и около 4% обращений приходилось на мой персональный номер, который деканат за моей спиной отдал студентам.

Большая часть обращений студентов приходится на время контрольных мероприятий (сессия, первый модель, второй модуль). И в подавляющем большинстве причиной является то, что студент забыл пароль.

3.3. Создание базы данных и запросов к ней, исходя из требований университета.

База данных создается автоматизировано контейнеризатором docker-compose. Такое решение было принято исходя из того, что на тот момент нужно было регулярно разворачивать систему на серверах трех организаций, а автоматизированная контейнеризация в пару команд значительно упрощает этот процесс.

В файле docker-compose.yml указаны следующие параметры для создания БД:

```
mysql:
  image: mysql:5.7
  ports:
    - 3306:3306
  environment:
    - MYSQL_ROOT_PASSWORD=#####
    - MYSQL_DATABASE=lms_imu
```

Таблицы в базе данных создавались с помощью миграций ORM Eloquent [13] (Расположены в порядке исполнения миграций):

- create_password_resets_table.php
- create_threads_table.php
- create_messages_table.php

- `create_participants_table.php`
- `create_semesters_table.php`
- `create_departments_table.php`
- `create_specialties_table.php`
- `create_groups_table.php`
- `create_users_table.php`
- `create_course_names_table.php`
- `create_courses_table.php`
- `create_streams_table.php`
- `create_tasks_table.php`
- `create_group_stream_table.php`
- `create_task_marks_table.php`
- `create_edu_materials_table.php`
- `create_orders_table.php`
- `create_classrooms_table.php`
- `create_tools_table.php`
- `create_classroom_tool_table.php`
- `create_time_tables_table.php`
- `create_course_tool_table.php`
- `create_syllabi_table.php`
- `create_template_tests_table.php`
- `create_questions_table.php`
- `create_curr_tests_table.php`
- `create_curr_test_group_table.php`
- `create_test_results_table.php`
- `create_payments_table.php`
- `create_course_stream_table.php`
- `create_completed_lessons_table.php`
- `create_notifications_table.php`
- `create_logs_table.php`

Пример миграции для таблицы users модели User:

```
1  <?php
2
3  use Illuminate\Support\Facades\Schema;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Database\Migrations\Migration;
6
7  class CreateUsersTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create( table: 'users', function (Blueprint $table) {
17             $table->bigIncrements( column: 'id');
18             $table->unsignedBigInteger( column: 'static_id')->nullable();
19             $table->string( column: 'name');
20             $table->string( column: 'login');
21             $table->tinyInteger( column: 'gender')->nullable();
22             $table->string( column: 'email')->nullable();
23             $table->string( column: 'ph_num')->nullable();
24             $table->string( column: 'address')->nullable();
25             $table->string( column: 'civic')->nullable();
26             $table->string( column: 'doc')->nullable();
27             $table->string( column: 'edu')->nullable();
28             $table->timestamp( column: 'birthday')->nullable();
29             $table->timestamp( column: 'paid_until')->nullable();
30             $table->string( column: 'password');
31             $table->string( column: 'role')->default( value: 'student');
32             $table->unsignedBigInteger( column: 'group_id');
33             $table->foreign( columns: 'group_id')->references( columns: 'id')->on( table: 'groups')
34                 ->onDelete( action: 'cascade');
35             $table->timestamp( column: 'deleted_at')->nullable();
36             $table->timestamp( column: 'last_seen')->nullable();
37             $table->json( column: 'docs_names')->nullable();
38             $table->json( column: 'reject_docs')->nullable();
39             $table->json( column: 'other')->nullable();
40             $table->tinyInteger( column: 'person_discount')->nullable();
41             $table->rememberToken();
42             $table->timestamps();
43         });
44     }
45 }
```

3.4. Заполнение таблиц. ETL-процессы загрузки базы данных.

В системе LMS на разных этапах развития использовались разные инструменты для обеспечения ETL процессов, ниже приведены основные этапы ETL процессов и то каким образом они реализованы в системе (этапы взяты из [14]):

- **Извлечение данных из источника:**

В системе LMS реализованы механизмы извлечения данных из следующих источников:

1. Веб-страница — получение данных с веб страницы реализованы с помощью штатных инструментов фреймворка Laravel с помощью html форм и последующей обработкой информации с них.
2. Any File — файлы любого типа (.pdf для аттестационных работ, .jpg/.png/.bmp для изображений студентов, .word/.pdf/.rtf для документов и т.д.) загрузка файла осуществляется через веб-страницу как file-input элемент формы и передается дальше в контроллер.
3. Json — первая необходимость в массовом импорте и экспорте при создании тестового модуля, а именно импорт и экспорт банка вопросов из шаблона.

Так выглядели экспорт и импорт вопросов в формат json:

```
public function export($template_id)
{
    $questions = TemplateTest::find($template_id)->questions;
    $filename = "questions_bank.json";
    $handle = fopen($filename, 'w+');
    fputs($handle, $questions);
    fclose($handle);
    $headers = $headers = array(
        'Content-Type: application/json;charset=utf-8'
    );

    return response()->download($filename, "questions_bank.json", $headers);
}

public function import($template_id, Request $request)
{
    $json_str = $request->file('import');
    $handle = fopen($json_str, 'r');
    $questions = json_decode(fgets($handle));
    fclose($handle);
    foreach ($questions as $question){
        $import = Question::create([
            'test_id' => $template_id,
            'wording' => $question->wording,
            'answers' => $question->answers,
        ]);
    }

    return redirect()
        ->back();
}
```

Но в скором времени от этого формата пришлось отказаться в виду следующих причин:

- Огромная сложность в ручном редактировании файлов (в особенности людям не связанным с IT)
- Сложность поиска ошибок и промежуточной проверки при тестировании
- Каждый раз вся логика импорта должна быть с нуля прописана в контроллере с нуля
- Частое возникновение плоховоспроизводимых ошибок связанных с кодировкой
- Некоторые файлообменники/соц. сети запрещают отправку друг другу файлов .json, заподозривая во вредоносном коде

В связи с этим было решено выбрать как основной тип файлов импорта-экспорта электронные таблицы.

4. Excel – для использования электронных таблиц использовался пакет Maatwebsite Laravel-Excel [15] с очень мощным инструментальным набором.

Впервые он был использован для вывода результатов теста студентов в читаемом и понимаемом формате, который можно по нажатию кнопки скачать и отправить в министерство образования (первым испытанием тестового модуля были гос. Экзамены).

5. sql – файлы формата .sql используются для хранения дампов базы данных тем самым образуя бэкапы. Не извлекается. Генерируется запросами к контейнеру с mysql

• **Очистка данных** (приведение разнородных данных к единому формату, удаление лишнего, устранение недочетов и пр);

Для веб-страниц и файлов загружаемых в контроллер реализована валидация средствами Laravel

При импорте каких либо данных через Excel создается класс импорта, рассмотрим на примере метода model класса UserImport. Данный метод получает на вход ассоциативный массив с содержанием исходной Excel Таблицы

```

public function model(array $row)
{
    if (!$row['login']) {
        $row['login'] = AdmissionController::get_login($row['name']);
    }
    if (!$row['password']) {
        $row['password'] = bcrypt(mb_strtolower(str_random( length: 6)));
    }
    if ($row['paid_until']) {
        $row['paid_until'] = Date::excelToDateTimeObject($row['paid_until']);
    }

    return User::create($row);
}

```

На примере ‘paid_until’ можно увидеть, как из формата даты/времени excel приводим к формату Carbon, хорошо интегрированным с PHP

- **Обогащение** (применение алгоритмов или внешних источников для получения новых данных, связанных с обрабатываемыми данными);

На примере выше можно рассмотреть параметры ‘login’ и ‘password’. При отсутствии этих данных в файле импорта (например первичная загрузка студентов, когда у них еще нет логина и пароля)

Login генерируется по схожей с МИФИ системой в формате abc001

Где abc – инициалы пользователя, а 001 это порядковый номер этих инициалов в БД.

А password при отсутствии это хеш от случайной строки из 6 символов

- **Трансформирование;**

Трансформирование можно рассмотреть на примере импорта вопросов. На вход при импорте подается таблица следующего вида:

	A	B	C	D	E
1	1. Mr. loepky is not conscious; he's	["c", "a", "a", "b", "a", "a", "b", "a", "a", "a", "a", "b", "b", "			
2	a. Lost				
3	b. tired				
4	c. unconscious				
5	d. evacuate				
6	2. Darlene is sighted, but Darren is				
7	a. blind				
8	b. sight				
9	c. able				
10	d. complaint				
11	3. Erin is mobile, but Eric is				
12	a. Immobile				
13	b. auto				

Которая в конечном итоге превращается в следующую структуру:

id	test_id	wording	answers
201	201	5 Mr. Loepky is not conscious; he's	{ "a. Lost": false, "b. tired": false, "d. evacuate": false, "c. unconscious": true }
202	202	5 Darlene is sighted, but Darren is	{ "c. able": false, "a. blind": true, "b. sight": false, "d. complaint": false }
203	203	5 Erin is mobile, but Eric is	{ "b. auto": false, "d. firm": false, "c. relaxed": false, "a. Immobile": true }
204	204	5 Frances is well, Frank is	{ "a. ill": false, "b. sick": true, "c. dull": false, "d. typical": false }
205	205	5 Gary has a limp handshake, while Greg's is	{ "a. firm": true, "b. rigid": false, "c. sick": false, "d. unwell": false }
206	206	5 Henri's bandages are tight, but Hank's are	{ "b. big": false, "d. new": false, "c. old": false, "a. Loose": true }
207	207	5 Ilona clenches her jaw when she is nervous, but Ian's...	{ "a. Dull": false, "c. fixed": false, "d. begged": false, "b. relaxed": true }
208	208	5 Jake has a muscle contraction in his right hand from ...	{ "muscle": false, "b . broken": false, "c. fracture": false, "d .sneezing": false, "a. Relaxation": t...
209	209	5 Kevin says his kids bring him joy most of the time, b...	{ "b. ill": false, "c. weak": false, "d. happy": false, "a. grief": true }
210	210	5 Dr. Levine begins the surgery by opening the abdomina...	{ "b. open": false, "c. broken": false, "d. change": false, "a. Closing": true }
211	211	5 Nerve pain is	{ "d. inertia": false, "c. headache": false, "a. neuralgia": true, "b. neurology": false }
212	212	5 loss of muscle strength is	{ "a. myalgia": false, "d. muscular": false, "c. myocardia": false, "b. myasthenia": true }
213	213	5 condition of sluggishness or lack of activity such as...	{ "b. inertia": true, "c. sarcoma": false, "a. botulism": false, "d. tonsillitis": false }
214	214	5 inflammation of the tonsils is	{ "b. throat": false, "c. gullet": false, "d. mindalis": false, "a. tonsillitis": true }
215	215	5 disease of disorder of the nerves or neural pathways ...	{ "b. sarcoma": false, "d .nervous": false, "c. neuralgia": false, "a. neuropathy": true }

- **Загрузка** (интеграция в единую целевую модель).

Интеграция в единую целевую модель происходит немедленно, как только получена нужная структура (массив либо коллекция элементов из которых создается модель). Местами используется прогон строк в цикле для создания модели. При создании модели, она по умолчанию сохраняется в соответствующей ей таблице.

3.5. Оценка размеров Базы Данных и каждого из файлов

Данной системой сейчас пользуются 3 организации, ниже представлены количественные данные их баз данных:

1. Медицинский университет:

Пользователей: 926

Групп: 92

Ячеек расписания: 288

Занятий отведено онлайн(с записями и журналом оценок): 857

Шаблонов тестов создано: 61

Вопросов создано: 2898

Тестов назначено на группу: 1561

Тестов пройдено студентами: 11530

Учебных материалов по курсу загружено: 302

Домашних работ создано преподавателями по курсу: 45

Домашних работ отправлено студентами: 2082

Размер дампа базы данных: 23 Мб

2. Медицинский университет:

Пользователей: 2234

Групп: 134

Ячеек расписания: 1286

Занятий отведено онлайн(с записями и журналом оценок): 5418

Шаблонов тестов создано: 321

Вопросов создано: 20805

Тестов назначено на группу: 3736

Тестов пройдено студентами: 37285

Учебных материалов по курсу загружено: 4384

Домашних работ создано преподавателями по курсу: 167

Домашних работ отправлено студентами: 5157

Размер дампа базы данных: 126 Мб

3. Медицинский колледж(на данный момент LMS используется только для тестов):

Пользователей: 511

Групп: 35

Ячеек расписания: 0

Занятий отведено онлайн(с записями и журналом оценок): 0

Шаблонов тестов создано: 93

Вопросов создано: 3460

Тестов назначено на группу: 442

Тестов пройдено студентами: 4985

Учебных материалов по курсу загружено: 2

Домашних работ создано преподавателями по курсу: 0

Домашних работ отправлено студентами: 0

Размер дампа базы данных: 7.5 Мб

Общий размер всех файлов из этих систем на облаке: 282.47 Гб

Заключение

В данном проекте проанализированы: информационные потребности университета, методы внедрения дистанционного образования, способы создания и поддержки АИС, опыт уже созданных и внедренных систем LMS.

В ходе выполнения данной работы были изучены: Реляционные базы данных и методы их моделирования, язык создания web-приложений PHP и фреймворк Laravel, способы контейнеризации docker-compose, поддержка web-сервера Nginx, администрирование UNIX систем,

На основании полученных знаний была разработана система LMS, удовлетворяющая информационным запросам всех участников процесса получения студентами высшего образования.

Данная система была протестирована и внедрена в эксплуатацию в трех образовательных организациях. В данный момент система модернизируется и дополняется в соответствии с обратной связью с конечными пользователями

Список литературы

1. ФЗ "О высшем и послевузовском профессиональном образовании" от 19 июля 1996 г.
2. Р.Н. Яценко, О.В. Полевич /Адаптивное Управление коммуникацией в системе дистанционного обучения Chamilo // БИ. –2012. –7. –С.231-234.
3. Lms Moodle в сетевом обучении / И.Брезгунова, С.Максимов // Наука и инновации. –2013. –128. –С.50-52.
4. Л.Н. Сорокина / Дистанционное обучение: настоящее и будущее // Столыпинский вестник. –2020. –2. –С.412-420.
5. MySQL: использование и администрирование. Викрам Васвани / В. Васвани — Санкт-Петербург.: Издательский центр «Питер», 2011. — 368 с.
6. О.З. Корабошев / СПОСОБЫ ЭФФЕКТИВНОГО ИСПОЛЬЗОВАНИЯ «ОБЛАЧНЫХ ТЕХНОЛОГИЙ» В СИСТЕМЕ ОБРАЗОВАНИЯ // Вестник науки и образования. –2020. –22-3. –С.41-43.
7. Н.А. Борсук, О.О. Козеева / Сравнительный анализ языков программирования Python и php // Символ науки. –2017. –4. –С.33-36.
8. Е.С. Шевченко / Сравнительное тестирование PHP-фреймворков // Вестник науки и образования. –2019. –10-2 (64). –С.40-45.
9. <https://cloud.yandex.ru/docs>
10. Т.В. Милушенко / СПЕЦИФИКА КОНСТРУИРОВАНИЯ СИСТЕМЫ УПРАВЛЕНИЯ ОБУЧЕНИЕМ LMS СРЕДСТВАМИ ОБЛАЧНЫХ СЕРВИСОВ И ТЕХНОЛОГИЙ // Вестник СИБИТа. –2020. –3. –С.24-29.
11. Амлаев К.Р., Кошель В.И., Ходжаян А.Б., Койчуева С.М., Агранович Н.В., Ветрова И.Н., Знаменская С.В. / МЕДИЦИНСКИЙ ВУЗ В УСЛОВИЯХ ПАНДЕМИИ COVID-19: НОВЫЕ ВЫЗОВЫ и выученные уроки // Медицинское образование и профессиональное развитие. –2020. –3 (39). –С.176-185.
12. Гридин В.Н., Анисимов В.И., Васильев С.А. / МЕТОДЫ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ СОВРЕМЕННЫХ ВЕБ-ПРИЛОЖЕНИЙ // Известия ЮФУ. Технические науки. –2020. –2. –С.193-200.
13. Брусов А.С., Тарасов С.О. / Использование фреймворка Laravel 5. 0 для разработки web-приложений // Современные материалы, техника и технологии. –2015. –3 (3). –С.48-52.
14. ETL Tools / David Haertzen. // The Analytical Puzzle: Profitable Data Warehousing, Business Intelligence and Analytics. — Technics Publications, 2012. — 346 с.
15. <https://docs.laravel-excel.com/>

16. Алгоритм составления расписания занятий / Ю.В. Береговых, Б.А. Васильев, Н.А. Володин // «Искусственный интеллект». –2009. –2. –С.50-56.

17. Управление данными: учебник для студ. высших учеб. заведений / А. В. Кузовкин, А. А. Цыганов, Б. А. Щукин. — М.: Издательский центр «Академия», 2010. — 256 с.