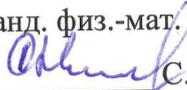


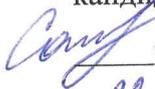
Министерство науки и высшего образования
Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет инновационных технологий
Кафедра инновационного обеспечения инновационной деятельности

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП, зав. каф. ИОИД,
канд. физ.-мат. наук, ст. н. с.

С. Л. Миньков
« 22 » 06 2020 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

МОДИФИКАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ АВТОМАТИЗАЦИИ
ОТЧЕТНЫХ ДОКУМЕНТОВ ПО УЧЕБНЫМ ДОСТИЖЕНИЯМ УЧАЩИХСЯ И
ПРОФЕССИОНАЛЬНЫМ ДОСТИЖЕНИЯМ ПЕДАГОГОВ В МАОУ ГИМНАЗИЯ
№13 Г. ТОМСКА

по основной образовательной программе подготовки бакалавров
направление подготовки 09.03.03 – Прикладная информатика

Руководитель ВКР
Доцент кафедры ИОИД,
кандидат физ.-мат. наук

С. И. Самохина
« 22 » 06 2020 г.

Автор работы
студент группы №18511

А. С. Кравченко

Министерство образования и науки Российской Федерации

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

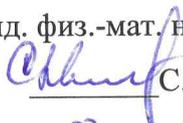
Факультет инновационных технологий

Кафедра информационного обеспечения инновационной деятельности

УТВЕРЖДАЮ

Руководитель ООП, зав. каф. ИОИД,

канд. физ.-мат. наук, ст. н. с.

 С.Л. Миньков

« 02 » 02 2020 г.

ЗАДАНИЕ

по подготовке выпускной квалификационной работы студенту группы № 18511

Кравченко Андрею Сергеевичу

1. Тема ВКР: «Модификация информационной системы автоматизации отчетных документов по учебным достижениям учащихся и профессиональным достижениям педагогов в МАОУ гимназия №13 г. Томска» (утверждена приказом по факультету от «20» мая 2020 г. №252-20).

2. Срок сдачи студентом законченной ВКР:

а) на кафедре 19.06.2020 г.

б) в ГЭК 23.06.2020 г.

3. Исходные данные к ВКР: отчет по производственной практике, научно-исследовательской работе, отчет по преддипломной практике, документация к средствам разработки.

4. Краткое содержание работы: анализ предметной области, анализ существующей информационной системы, выявление и устранение критических недостатков информационной системы.

5. Перечень графического материала: описание аналогов, постановка цели работы, описание логической и физической модели базы данных, описание интерфейса информационной системы.

6. Дата выдачи задания: 2 февраля 2020 г.

Руководитель ВКР,

Доцент кафедры ИОИД

 С. И. Самохина

Задание принял к исполнению

 А. С. Кравченко

РЕФЕРАТ

Бакалаврская работа содержит: 53 с., 17 рис., 45 табл., 16 литературных источников.

Список ключевых слов: АВТОМАТИЗАЦИЯ ОТЧЕТНЫХ ДОКУМЕНТОВ, ИНФОРМАЦИОННАЯ СИСТЕМА, БАЗА ДАННЫХ, ANGULAR, BOOTSTRAP, ВЕБ-ПРИЛОЖЕНИЕ, C#, WEB API, JSON, AJAX, JAVASCRIPT, TYPESCRIPT, FRONT-END, BACK-END.

Объектом исследования данной бакалаврской работы является существующая информационная система отчетных документов МАОУ гимназии № 13 города Томска, разработанная студентами выпускниками факультета инновационных технологий в 2018 году.

Целью бакалаврской работы является модификация информационной системы для приведения ее в рабочее состояние для дальнейшей передачи информационной системы в эксплуатацию работникам гимназии №13 города Томска.

Результатом данной выпускной квалификационной работы является модифицированный программный продукт, позволяющий реализовывать поставленные цели.

Текст ВКР выполнен в текстовом редакторе MS Word 2016. Также использована программа MS Paint. В качестве графического материала представлена презентация, выполненная в MS Power Point 2016.

ABSTRACT

Bachelor's work contains: 53 pages, 17 figures, 45 tablets, 16 literary sources.

Key words list: AUTOMATION OF REPORTING DOCUMENTS, INFORMATION SYSTEM, DATABASE, ANGULAR, BOOTSTRAP, WEB APPLICATION, C #, WEB API, JSON, AJAX, JAVASCRIPT, TYPESCRIPT, FRONT-END, BACK.

The object of the study of this bachelor's work is the existing information system of reporting documents of the MAOU gymnasium No. 13 of the city of Tomsk, developed by students graduates of the faculty of innovative technologies in 2018.

The aim of the bachelor's work is to modification the information system to bring it into working condition for the further transfer of the information system to the operation of the employees of gymnasium No. 13 of the city of Tomsk.

The result of this final qualification work is a modernized software product that allows you to achieve your goals.

The WRC text was made in the text editor MS Word 2016. The program MS Paint was also used. A presentation made at MS Power Point 2016 is presented as graphic material.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	6
1 Обзор информационных систем	7
2 Сравнительный анализ информационных систем	13
3 Архитектура построения информационной системы	14
3.1 BACKEND технологии	14
3.1.1 Язык программирования С#	14
3.1.2 ASP.NET WEB API 2	14
3.1.3 Entity Framework	16
3.1.4 Microsoft SQL Server	17
3.1.5 IDE Visual Studio	19
3.2 FRONTEND технологии	20
3.2.1 Angular 5	20
3.2.2 TypeScript	21
3.2.3 HTML 5	23
3.2.4 CSS3	24
3.2.5 Архитектура взаимодействия frontend и backend	24
4 Анализ информационной системы	26
4.1 Модель вариантов использования	26
4.2 Логическая модель информационной системы	26
4.3 Физическая модель	28
4.3.1 Общие сущности	28
4.3.2 Сущности, относящиеся к модулю начальных классов	31
4.3.3 Сущности модуля учителя	35
4.3.4 Сущности модуля мероприятий	39
5 Основные недостатки информационной системы	42
6 Устранение замечаний	44
ЗАКЛЮЧЕНИЕ	51
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	52

ВВЕДЕНИЕ

Целью выпускной квалификационной работы является модификация существующей информационной системы с учетом предложений и замечаний эксплуатирующей организации.

Предметом исследования является информационная система для образовательного учреждения. Использование информационных систем изменяет организационную структуру образовательного учреждения и функции всех участников образовательного процесса, улучшает и упрощает процесс получения и анализа информации о состоянии управляемого объекта. Современная информационная система для образовательного учреждения представляет собой комплекс различных программ для автоматизации основных процессов (оформление документов, ведение личных дел преподавателей, сотрудников и учащихся, составление расписаний и т.д.). Основная цель и задача такой информационной системы – обеспечить необходимую интеграцию и создание единого образовательного информационного пространства. Решение этой проблемы начинается с обеспечения уникальности и полной эквивалентности, а также интерпретации описания предметной области в информационной комнате для различных операционных систем и приложений. В то же время, ИС не должна отклоняться от указанных требований в течение всего жизненного цикла системы.

Полное функционирование современного образовательного учреждения уже невозможно без таких систем, поскольку человек без помощи подходящих программ не сможет самостоятельно обрабатывать большое количество различной информации. Цели создания программного продукта на рынке, который можно использовать для автоматизации процессов в образовательном учреждении:

- 1) Облегчить и упростить деятельность администраторов и преподавателей, с тем чтобы сократить время, необходимое для выполнения рутинных операций, и увеличить время, непосредственно выделяемое на процесс обучения;
- 2) Обеспечить сбор, обработку и анализ информации о ходе образовательного, воспитательного или иного процесса, и принятие обоснованных решений;
- 3) Сформировать единую информационную инфраструктуру учреждения;
- 4) Организовать информационное взаимодействие между всеми сотрудниками учебного заведения.

Для выполнения цели выпускной квалификационной работы необходимо:

- 1) Произвести анализ действующей информационной системы;
- 2) Произвести устранение критичных недостатков информационной системы;

1 Обзор информационных систем

ИАС «АВЕРС». Фирма «Аверс» предлагает комплекс программ, который автоматизирует широкий спектр бизнес-процессов учебного заведения [1]:

1) ИАС «АВЕРС: Управление образовательным учреждением» (КРМ «Директор»). Инструментальная среда информационной поддержки управленческих решений в деятельности администрации общеобразовательных учреждений (рисунок 1).

Предназначена для управления следующими процессам:

- планирование учебно-образовательной деятельности;
- мониторинг учебно-образовательной деятельности;
- унификация внутришкольного и кадрового делопроизводства.

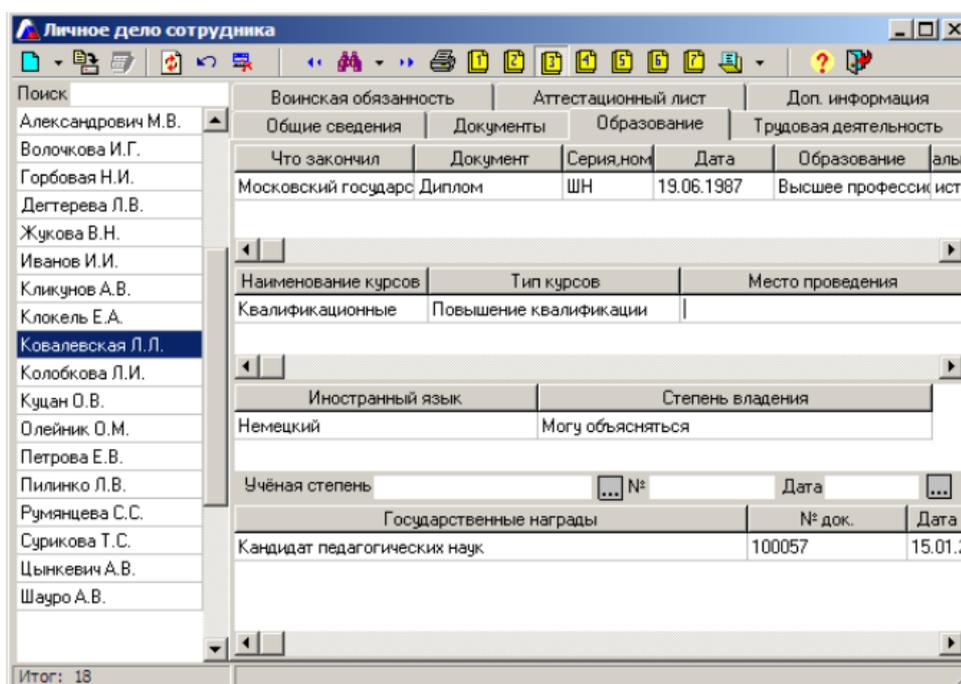


Рисунок 1 – Раздел «Личное дело сотрудника» КРМ «Директор»

2) ИАС «АВЕРС: Электронный Классный Журнал». Инструментальная среда информационной поддержки системы государственно-общественного управления образовательным учреждением. Предназначена для автоматизации процессов учета и контроля оценивания успеваемости в учебном заведении.

3) ИАС «АВЕРС: Расписание». Программа состоит из четырех разделов: «Списки», «Нагрузки», «Расписание» и «Замены». Раздел «Списки» используется для ввода, редактирования и печати списков классов, учителей, предметов, классных комнат и учащихся. Раздел «Нагрузки» используется для создания, редактирования и печати учебных программ и нагрузок для учителей, а также для их оценки. Занятия, учителя,

классные комнаты и школа в целом запланированы в разделе «Расписание». В разделе «Замены» учителей можно заменить. Расписание может быть создано в разных режимах: автоматический, ручной или комбинированный; переход из одного режима в другой может происходить в любое время при создании расписания. При планировании в автоматическом режиме программа учитывает все сформулированные требования к графику, в том числе требования СанПиН. При планировании в ручном режиме программа предлагает возможные варианты организации уроков выбранного учителя, возможные варианты заполнения пустых ячеек в расписании и контроля количества мест в классах.

4) И другие.

«Школьный офис». Программный комплекс «Школьный офис» — простой и удобный инструмент, разработанный в тесном сотрудничестве со специалистами школ и органов управления, для автоматизации управления современным образовательным учреждением. [2].

Модули ПО:

1) «Паспорт образовательного учреждения (общие сведения)» — характеристика образовательного учреждения как части региональной сети учебных заведений: учредители, лицензии, реквизиты, контактные телефоны и др.

2) «Контингент» — предназначен для хранения широкого спектра личных данных учащихся образовательного учреждения (рисунок 2). Позволяет производить генерацию произвольных отчетных документов. Поддерживает функцию передачи электронных личных дел учащихся при смене учебного заведения.

3) «Кадры» — предназначен для хранения информации о сотрудниках учебного заведения. Ведет личные записи, ведет учет и планирует своевременную передачу обучения и переаттестацию, рассчитывает стаж работы сотрудников, учитывает движение сотрудников и работает с архивом. Позволяет генерировать любые отчетные документы. Он поддерживает передачу электронных личных дел учителей при смене школы.

4) «Тарификация» — полный цикл планирования и подготовки рейтинга учебного заведения к новому учебному году, от создания учебного плана до распределения нагрузки; от расчета нормативной таблицы персонала до позиционирования сотрудников с учетом работы групп дополнительного образования и долгосрочных групп. Подробная система проверки соответствия стандартам и требованиям законодательства.

5) «Нормативные документы» — площадка для формирования банка нормативно-методических документов, регламентирующих деятельность образовательного учреждения; банк электронных копий правоустанавливающих документов; Регламент научно-методических разработок учебного заведения; награды для

учебных заведений и работников. Быстрый доступ к тематическим документам в каждом разделе программы - необходимые правила всегда доступны.

б) «Администрирование системы» — предназначен для осуществления сервисных функций. Поддерживает:

- организацию многопользовательской работы;
- создание и разграничение прав доступа пользователей;
- резервное копирование и восстановление данных;
- настройку параметров обновления программного обеспечения через Internet;
- импорт данных для упрощения первоначального наполнения базы данных.

Комплекс интегрирован с системой интернет-дневников.

The screenshot shows a web-based form titled "Сведения об учащихся..." (Student Information) for a student named "Баранова Антонина Ярославовна (4-6)". The form is organized into several sections with blue headers:

- Личные данные**: Includes fields for "Класс" (Class) set to "4-6", "Профильный класс" (Profile class) with a checkbox, and "Форма обучения" (Form of education) set to "Очная" (Full-time).
- Иностранный язык**: Includes "Основной" (Main) language set to "Английский" (English) and "Дополнительный" (Additional) language set to "Немецкий" (German).
- Прибытие**: Includes "Дата прибытия" (Arrival date) set to "23.04.2002", "Откуда прибыл" (From where arrived) set to "Из дошкольного образовательного учреждения" (From a preschool educational institution), and "Откуда (уточнение)" (From where (clarification)) set to "Детский сад" (Nursery).
- Повторное обучение**: Includes a checkbox for "Оставлен на повторное обучение" (Left for repeat education) and a "Привена" (Admission) field with a right arrow button.
- Выбытие**: Includes "Дата выбытия" (Date of departure), "№ приказа о выбытии" (Order number for departure), "Привена выбытия" (Admission of departure), "Сведения о выпускнике" (Information about the graduate), and a "Выдано личное дело" (Personal file issued) checkbox.
- Примечание**: A text area for additional notes.

At the bottom right, there are three buttons: "OK", "Отмена" (Cancel), and "Применить" (Apply).

Рисунок 2 – Раздел «Сведения об учащихся» модуля «Контингент»

«GS-Ведомости». Система комплексной автоматизации образовательного учреждения профессионального образования «GS-Ведомости» представляет собой современный программный продукт, с помощью которого можно автоматизировать процессы, связанные с администрированием учреждений среднего профессионального и высшего образования. Система автоматизации GS-Ведомости имеет модульную структуру. Таким образом, отдельные функции системы сгруппированы в отдельные модули, состав которых пользователь выбирает самостоятельно. [3].

Модули:

1) «Абитуриент» – автоматизирует бизнес-процессы, осуществляемые приемной комиссией. Осуществляет хранение личных данных абитуриентов, их оценочные результаты вступительных экзаменов. Позволяет формировать формы и отчеты для печати;

2) «Контингент учащихся» – предназначен для хранения личных данных учеников, информацию об их успеваемости. Позволяет осуществлять формирование и печать большого количества разнообразных отчетных документов (рисунок 3);

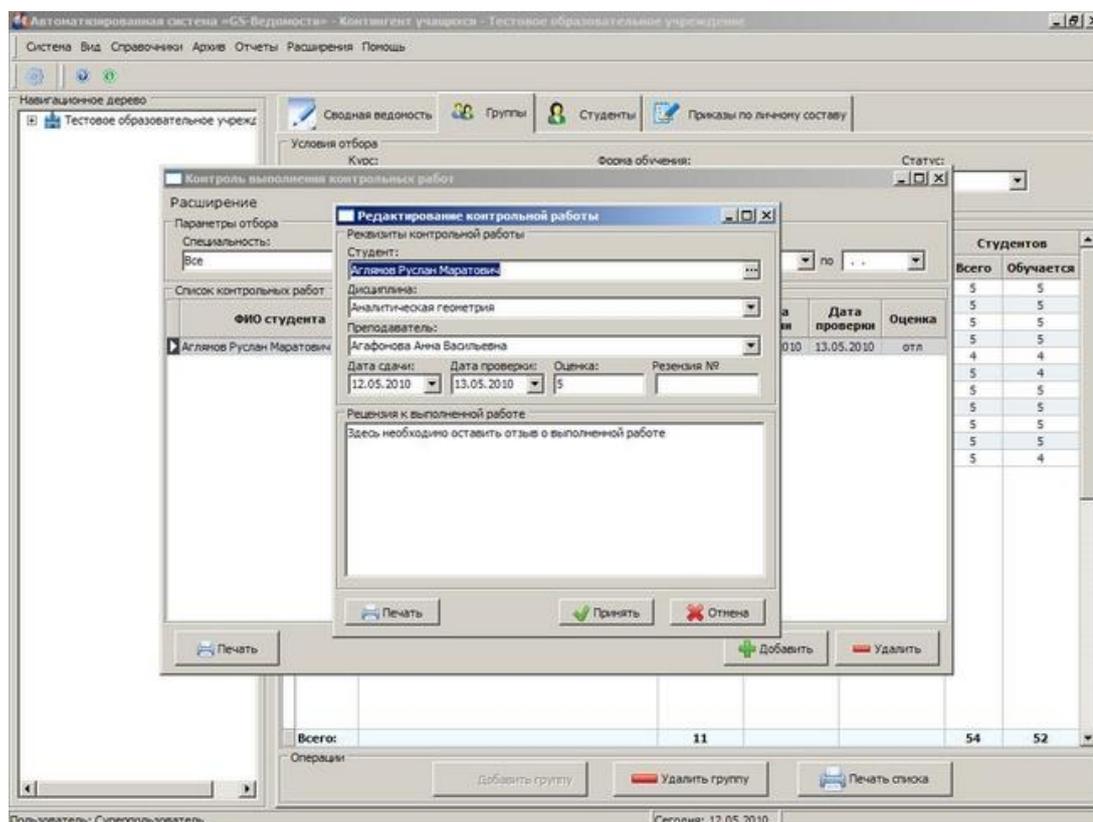


Рисунок 3 – Модуль «Контингент учащихся»

3) «Кадровый учёт» – предназначен для автоматизации бизнес-процессов, осуществляемых отделом кадров, хранит личные данные преподавательского состава учебного заведения. Позволяет формировать разнообразные отчетные документы;

4) «Учебные планы» – предназначен для генерирования, редактирования учебных планов учебного заведения. Позволяет формировать и редактировать на основе учебного плана учебные графики;

5) «Кафедра / ПЦК» – предназначен для автоматизации работы кафедры / предметно-цикловой комиссии, распределения и редактирования информации об учебной нагрузке преподавателей кафедры / ПЦК;

6) «Конструктор расписаний» – позволяет генерировать и редактировать расписание занятий в различных режимах: автоматическом, полуавтоматическом или ручном;

7) «Тестирование учащихся» – обеспечивает возможность организации проведения централизованного тестирования учащихся. Состоит из двух модулей: приложение генератора тестов, приложения прохождения тестов;

8) «Управление контрагентами» – предназначен для учета контрагентов учебного заведения, данных о договорах приема, учебных курсах, стажировках и т.д.;

9) «Документооборот» – позволяет организовать процесс электронного документооборота в системе;

10) «Администратор» – позволяет производить администрирование базы данных, осуществлять разграничение прав доступа для пользователей к отдельным функциональным блокам системы, настраивать и корректировать документооборот;

11) «Директ» – производит учет рабочего времени работников образовательного заведения, позволяет осуществлять планирование деятельности учебного заведения;

12) «Хранилище документов» – позволяет организовать единое сетевое хранилище данных в любом формате (текстовые, графические, мультимедийные файлы), хранилище файлов, сопровождающих учебный процесс, и организацию электронного архива документов с функцией поиска.;

13) «Библиотека» – автоматизирует внутренний библиотечный учет;

14) «Общежитие» – автоматизирует деятельность общежитий учебного заведения;

15) «АРМ руководителя» – предназначен для обеспечения ответственных руководителей образовательного учреждения (председатель приемной комиссии, декан, ректор, проректор, директор и т.д.) быстрым доступом к важнейшим аналитическим функциям системы «GS-Ведомости»;

16) «Импорт/экспорт» – позволяет производить импортирование данных в систему из документов в формате MS Excel и со сканера, экспортировать данные базы данных в файлы различных форматов (Excel, XML и др.);

17) «Студия» – позволяет формировать отчеты любой структуры и сложности на базе генератора отчетов FastReport.

Также комплекс имеет дополнительно приобретаемый модуль «GS-Ведомости: Online». «GS-Ведомости: Online» – это web-интерфейс системы «GS-Ведомости», который позволяет производить манипуляции на данными, хранящимися в базе данных, установленной на удаленном сервере через сеть Интернет. Модуль работает на HTTP-сервере Apache.

2 Сравнительный анализ информационных систем

Для простоты анализа сформирована таблица с характеристиками информационных систем, в которой метка «Галочка» обозначает, что данному продукту соответствует характеристика, приведенная в левой колонке.

Таблица 1 – Сравнительная характеристика ИС

Характеристика	ИАС «АВЕРС»	«Школьный офис»	«GS-Ведомости»
Расписание	✓		✓
Успеваемость, посещаемость	✓		✓
Электронный журнал	✓		✓
Методическая работа	✓	✓	✓
Рабочие программы	✓	✓	✓
Календарный график	✓	✓	✓
Общежитие			✓
Кадровый учет	✓	✓	✓
Формирование отчетов	✓	✓	✓

Из таблицы видно, что все рассмотренные программы выполняют основные функции такие как кадровый учет, календарный график, формирование рабочих программ, формирования отчетов для образовательных учреждений. Информационные системы ИАС «АВЕРС» и «GS-Ведомости» по характеристикам хорошо подходят для образовательных учреждений в сфере школьного образования.

3 Архитектура построения информационной системы

Информационная система является web-приложением, созданным с использованием современных web-технологий. Разработана на основе архитектуры «backend-frontend». Серверная часть информационной системы (backend) разработана в среде разработки «Visual Studio» с помощью фреймворка ASP.NET WEB API 2 и обеспечивает взаимосвязь между клиентским приложением и базой данных под управлением СУБД Microsoft SQL Server. Клиентская часть информационной системы разработана с использованием фреймворка Angular 5.

3.1 BACKEND технологии

Backend разработка – это получение информации, её запись в базу и возврат данных на сайт, где они будут представлены пользователю средствами Frontend [4].

3.1.1 Язык программирования C#

C# – объектно-ориентированный язык программирования. Разработан в 1998 – 2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников – языков C++, Delphi, Модула, Smalltalk и, в особенности, Java – C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественное наследование интерфейсов) [5].

3.1.2 ASP.NET WEB API 2

ASP.NET Web API – это инфраструктура, упрощающая создание служб HTTP, которые охватывают широкий круг клиентов, включая браузеры и мобильные

устройства. ASP.NET Web API является идеальной платформой для создания приложений RESTful в .NET Framework.

Средство Web API основано на добавлении в приложение ASP.NET MVC Framework контроллера специального вида. Эта разновидность контроллеров, которая называется контроллером API, обладает двумя характеристиками:

- Методы действий возвращают объекты моделей, а не объекты типа ActionResult.
- Методы действий выбираются на основе HTTP-метода, используемого в запросе.

Объекты моделей, возвращаемые методом действия контроллера API, кодируются в формате JSON и отправляются клиенту. Контроллеры API предназначены для доставки веб-служб данных, поэтому они не поддерживают представления, компоновки или любые другие средства, которые применялись для генерации HTML-разметки в примере созданного приложения.

Отсутствие возможности у контроллера API генерировать HTML-разметку из представлений является причиной, по которой в одностраничных приложениях комбинируются стандартные приемы ASP.NET MVC Framework с Web API. Инфраструктура ASP.NET MVC Framework выполняет шаги, требуемые для доставки HTML-содержимого пользователю (включая аутентификацию, авторизацию, выбор и визуализацию представления). После того, как HTML-содержимое доставлено в браузер, запросы Ajax, генерируемые содержащимся внутри кодом JavaScript, будут обрабатываться контроллером Web API.

В обычных контроллерах можно создавать методы действий, которые возвращают данные JSON для поддержки Ajax, но контроллер API предлагает альтернативный подход. Этот подход предусматривает отделение действий, относящихся к данным, от действий, связанных с представлением, и делает создание универсальных приложений Web API быстрым и простым.

Контроллеры Web API применяют стиль REST (Representation State Transfer или "передача состояния представления").

Также надо отметить, что платформа Web API 2 не является частью фреймворка ASP.NET MVC и может быть задействована как в связке с MVC, так и в соединении с Web Forms. Поэтому в Web API имеется своя система версий. Так, первая версия появилась с .NET 4.5. А вместе с .NET 4.5.1 и MVC 5 вышла Web API 2.0 [6].

3.1.3 Entity Framework

Entity Framework – это объектно-ориентированная технология, основанная на фреймворке .NET для работы с данными. Хотя вы можете использовать стандартные инструменты ADO.NET для создания соединений, команд и других объектов для взаимодействия с базами данных, Entity Framework предлагает более высокий уровень абстракции, с которым вы можете работать не только с таблицами базы данных, но и непосредственно с самими данными, независимо от типа хранилища данных. Когда операции над таблицами, индексами и ключами выполняются на физическом уровне, Entity Framework работает непосредственно с объектами на концептуальном уровне.

Первая версия Entity Framework - 1.0 была выпущена в 2008 году и предлагала только очень ограниченные функции, базовую поддержку ORM (object-relational mapping – отображение данных на реальные объекты) и уникальный подход к взаимодействию с базой данных – Database First. Много изменилось с выпуском версии 4.0 в 2010 году – с тех пор Entity Framework была рекомендованной технологией для доступа к данным, и в платформу были введены новые функции для взаимодействия с базой данных – Model First и Code First

Дальнейшие функциональные улучшения последовали с выпуском версии 5.0 в 2012 году. Наконец, Entity Framework 6.0 был выпущен в 2013 году с возможностью асинхронного доступа к данным.

Основополагающая концепция Entity Framework – понятие сущности (entity). Сущность – это запись, связанная с конкретным объектом. Следовательно, Entity Framework подразумевает работу не с таблицами, а именно с объектами и их коллекциями.

Каждая сущность, как и любой объект реального мира, имеет список определенных свойств. Свойства не обязательно представляют простые данные типа int, но могут также представлять более сложные структуры данных. И каждая сущность может иметь одно или несколько свойств, которые отличают эту сущность от других и однозначно идентифицируют эту сущность. Подобные свойства называются ключами.

В то же время, сущности могут быть связаны через отношения «один ко многим», «один к одному» и «многие ко многим», подобно тому, как в реальной базе данных происходит связь через внешние ключи.

Отличительной чертой Entity Framework является использование запросов LINQ для выборки данных из БД. С помощью LINQ можно не только извлекать определенные строки, хранящие объекты, из БД, но и получать объекты, связанные различными ассоциативными связями.

Другим ключевым понятием является Entity Data Model. Эта модель сопоставляет классы сущностей с реальными таблицами в БД.

Entity Data Model состоит из трех уровней: концептуального, уровень хранилища и уровень сопоставления (маппинга).

На концептуальном уровне происходит определение классов сущностей, используемых в приложении.

Уровень хранилища определяет таблицы, столбцы, отношения между таблицами и типы данных, с которыми сопоставляется используемая база данных.

Уровень сопоставления (маппинга) служит посредником между предыдущими двумя, определяя сопоставление между свойствами класса сущности и столбцами таблиц.

Таким образом, взаимодействие с таблицами базы данных осуществляется через классы, определенные в приложении.

Способы взаимодействия с БД:

Entity Framework предполагает три возможных способа взаимодействия с базой данных [7]:

- 1) Database first: Entity Framework создает набор классов, которые отражают модель определенной базы данных;
- 2) Model first: сначала программист создает модель базы данных, по которой Entity Framework в свою очередь создает реальную базу данных на сервере;
- 3) Code first: разработчик создает класс модели данных, которые будут храниться в базе данных, а Entity Framework по этой модели генерирует базу данных и ее таблицы.

3.1.4 Microsoft SQL Server

Microsoft SQL Server – это система управления реляционными базами данных или RDBMS, которая поддерживает широкий спектр приложений обработки транзакций, бизнес-аналитики и аналитики в корпоративных ИТ-средах. Это одна из трех ведущих в отрасли технологий баз данных, а также Oracle Database и IBM DB2.

Как и другое программное обеспечение RDBMS, Microsoft SQL Server построен поверх SQL, стандартизованного языка программирования, который администраторы баз данных и другие ИТ-специалисты используют для управления базами данных и запроса данных, которые они содержат. SQL Server привязан к Transact-SQL (T-SQL), реализация SQL от Microsoft, которая добавляет набор проприетарных расширений программирования на стандартный язык.

Microsoft SQL Server 2016, который стал общедоступным в июне 2016 года, был разработан в рамках «первой технологии мобильных технологий, облачных технологий», принятой Microsoft двумя годами ранее. Среди прочего, SQL Server 2016 добавил новые функции для настройки производительности, оперативной аналитики в реальном времени, визуализации данных и отчетности на мобильных устройствах, а также поддержку гибридных облаков, которая позволяет администраторам баз данных запускать базы данных на основе комбинации локальных и общедоступных облачных сервисов для снижения затрат на ИТ. Например, технология SQL Server Stretch Database перемещает редко получаемые данные с локальных устройств хранения в облако Microsoft Azure, сохраняя при этом данные для запросов, если это необходимо.

Как и другие технологии СУБД, SQL Server в основном основан на строковой структуре таблицы, которая связывает связанные элементы данных в разных таблицах, поэтому избыточные данные не нужно хранить в нескольких местах в базе данных. Реляционная модель также обеспечивает ссылочную целостность и другие ограничения целостности для поддержания точности данных. Эти обзоры являются частью более широкого соответствия принципам атомарности, согласованности, изоляции и долговечности, которые в совокупности называются свойствами ACID, для обеспечения надежной обработки транзакций базы данных.

Основным компонентом Microsoft SQL Server является ядро базы данных SQL Server, которое контролирует хранение, обработку и безопасность данных. Он включает в себя реляционный механизм, который обрабатывает команды и запросы, и механизм хранения, который управляет файлами базы данных, таблицами, страницами, индексами, буферами данных и транзакциями. Хранимые процедуры, триггеры, представления и другие объекты базы данных также создаются и выполняются ядром базы данных.

Операционная система SQL Server или SQLOS находится под капотом базы данных. Он обрабатывает подчиненные функции, такие как управление памятью и вводом-выводом, планирование и блокировка данных, чтобы избежать конфликтующих обновлений. Уровень сетевого интерфейса расположен над ядром базы данных и использует протокол потока табличных данных Microsoft для облегчения взаимодействия запросов и ответов с серверами базы данных. На уровне пользователя администраторы и разработчики баз данных SQL Server пишут, среди прочего, операторы T-SQL для создания и изменения структур баз данных, управления данными, обеспечения безопасности и защиты баз данных.

В зависимости от задачи команды T-SQL разделяют на несколько типов:

DDL («Data Definition Language» – Язык определения данных). Этот тип команд предназначен для создания объектов базы данных: сама база данных, ее таблицы, индексы, представления, процедуры и прочее.

К этому типу относятся следующие команды:

- CREATE: создание объектов базы данных;
- ALTER: редактирование объектов базы данных;
- DROP: удаление объектов базы данных;
- TRUNCATE: полное удаление всех данных из таблиц базы данных;

DML («Data Manipulation Language» – Язык манипуляции над данными). Этот тип включает команды для манипулирования данными в базу данных: осуществление выборки, редактирование, добавление, удаление.

К этому типу относятся следующие команды:

- SELECT: извлечение данные из таблиц базы данных;
- UPDATE: редактирование данных в таблицах базе данных;
- INSERT: добавление новых данных в существующие таблицы базы данных;
- DELETE: удаление данных из таблиц базы данных;

DCL («Data Control Language» – Язык управления доступа к данным). Этот тип команд предназначен предоставления/отзыва прав доступа к танным [8].

К этому типу относятся следующие команды:

- GRANT: предоставляет права для доступа к данным;
- REVOKE: отзывает права на доступ к данным;

3.1.5 IDE Visual Studio

Visual Studio .NET – это среда визуального программирования Microsoft для создания веб-служб на основе использования расширяемого языка разметки (XML). Набор продуктов обеспечивает визуальный интерфейс для идентификации программы как веб-службы, форм для создания пользовательского интерфейса (включая поддержку интерфейсов мобильных устройств), функций для интеграции существующих данных приложения и для отладки. Visual Studio .NET поставляется с .NET Framework, включая Common Language Runtime, и включает в себя несколько языков программирования, такие как Visual Basic, C++ и C#.

Visual Studio .NET издается в трех уровнях поставки: Community, Professional, Enterprise Developer (включая Microsoft SQL Server). По мнению Microsoft, Visual Studio .NET нацелена на установление эталона легкости в разработке приложений для интернета

в текущем десятилетии, так же как его Visual Basic установила ориентир для визуального программирования в 1990-х годах [9].

3.2 FRONTEND технологии

Фронтендом (англ. front-end) называют клиентскую сторону пользовательского интерфейса по отношению к программно-аппаратной части сервиса.

Говоря простым языком, frontend – это разработка функциональности и пользовательского интерфейса, работающих на клиентской стороне приложения или веб-сайта. Сюда относят всё, что пользователь видит, открывая веб-страницу. Для создания удобного и востребованного продукта фронтенд-разработчики сотрудничают с программистами, дизайнерами, UX-аналитиками.

К frontend-разработке относят [10]:

- HTML – язык разметки документов, посредством которого формируют структуру веб-страницы: заголовки, списки, абзацы и т. п.;

- CSS – язык для стилизации внешнего вида документа, его описания. Благодаря CSS-коду браузер понимает, как отображать элементы. С помощью CSS задаются цвета и параметры шрифтов, определяется расположение блоков веб-сайта и т. п. Также он даёт возможность выводить один и тот же документ в нескольких стилях, например, для печати и т. д.;

- JavaScript – язык, в буквальном смысле оживляющий веб-страницы. Он обеспечивает реагирование интерфейса на действия пользователя, обрабатывает клики мышкой, нажатия клавиш, перемещения курсора. Также посылает запросы на сервер, загружает данные, позволяет вводить сообщения и много чего ещё.

3.2.1 Angular 5

Angular является структурной основой для динамических веб-приложений, которая позволяет использовать HTML в качестве языка шаблонов, а затем расширить синтаксис HTML для выражения компонентов приложения. С помощью привязки данных и внедрения зависимостей вы можете удалить большую часть кода, который вам придется написать.

Основные факты об Angular:

- Typescript используется в качестве языка шаблонов, который является языком программирования от Microsoft;

- хотя Typescript является основным языком для Angular, приложения также могут быть написаны на таких языках, как Dart или JavaScript;

- Angular поддерживается Google;
- Angular нацелен на разработку одностраничных приложений, то есть SPA-решений (Single Page Application). Примером являются популярные приложения для социальных сетей (Twitter, Instagram и Facebook);
- Angular предлагает клиентскую инфраструктуру MVC, которую можно использовать для запуска и создания динамических приложений с современным уровнем качества;
- приложения, написанные на Angular, совместимы с различными браузерами. Angular автоматически обрабатывает код JavaScript, подходящий для каждого браузера;
- чистый и точный дизайн пользовательского интерфейса;
- простая маршрутизация;
- структура Angular облегчает расширение синтаксиса HTML и легко создает повторно используемые компоненты по директивам.

В целом, Angular – это платформа для создания больших, мощных и простых в обслуживании веб-приложений.

Стоит также отметить, что Angular предлагает рендеринг на стороне сервера, который ускоряет загрузку домашней страницы и, следовательно, улучшает поисковую оптимизацию и облегчает сканирование динамических страниц. Быстрый рендеринг страниц значительно улучшает восприятие веб-приложений следующего поколения, написанных на Angular. Также, с помощью данного фреймворка можно [11]:

- легко тестировать код;
- легко создавать персонализированные объектные модели документа (Document Object Model, DOM);
- моделировать данные ограничено для использования небольших моделей данных, что делает код простым и легким для тестирования.

3.2.2 TypeScript

TypeScript – это язык программирования на основе JavaScript.

Развитие TypeScript началось в конце 2012 года. Несмотря на то, что он исходит от Microsoft, а его реальным создателем является программист Андерс Халесберг, который также известен как создатель таких языков, как Delphi, C #, этот проект сразу начал развиваться как OpenSource. И с самого начала новый язык быстро распространялся благодаря своей гибкости и производительности. Многие проекты, разработанные на JavaScript, были портированы на TypeScript. Популярность и актуальность идей нового

языка привели к тому, что некоторые из этих идей впоследствии стали частью нового стандарта JavaScript. Новая версия одной из самых популярных веб-платформ, Angular, была полностью написана на TypeScript компаниями Microsoft и Google.

Однако, казалось бы, зачем нужен еще один язык программирования для клиентской стороны в среде Web, если со всей той же самой работой прекрасно справляется и традиционный JavaScript, который используется почти на каждом сайте, которым владеют многие разработчики, и который достаточно популярен в сообществе программистов. Однако TypeScript – это не просто новый JavaScript.

Прежде всего, следует отметить, что TypeScript – это строго типизированный и скомпилированный язык, который, вероятно, ближе к Java, C # и другим строго типизированным языкам. Выходные данные компилятора создают тот же JavaScript, который затем выполняется браузером. Однако, строгая типизация сокращает количество потенциальных ошибок, которые могут возникнуть при разработке JavaScript.

Во-вторых, TypeScript реализует множество концепций, характерных для объектно-ориентированных языков, как, например, наследование, полиморфизм, инкапсуляция и модификаторы доступа и так далее.

В-третьих, потенциал TypeScript позволяет быстрее и проще писать большие, сложные встроенные программы, а также легче поддерживать, разрабатывать, масштабировать и тестировать, чем стандартный JavaScript.

В-четвертых, TypeScript разрабатывается как проект с открытым исходным кодом и, как и многие другие проекты, размещается на github. Кроме того, он кроссплатформенный, что означает, что мы можем использовать как Windows, так и MacOS или Linux для разработки.

В то же время TypeScript – это расширенный набор JavaScript, что означает, что каждая JavaScript-программа является программой TypeScript. В TypeScript вы можете использовать все конструкции, используемые в JavaScript – те же операторы, условные, циклические конструкции. Кроме того, код TypeScript скомпилирован в Javascript. В конечном итоге, TypeScript – это всего лишь инструмент, облегчающий разработку приложений.

Подавляющее большинство браузеров поддерживают код JS, сгенерированный компилятором TypeScript, поскольку он в основном ориентирован на стандарт ECMAScript 3, хотя TS также поддерживает стандарты ECMAScript 5 и ECMAScript 2015 / 2017 [12].

3.2.3 HTML 5

HTML (HyperText Markup Language) – это язык гипертекстовой разметки, разработанный в первую очередь для создания документов в Интернете. HTML был создан в начале 1990-х годов как примитивный язык для создания веб-страниц. Большинство веб-сайтов в настоящее время используют в большей или меньшей степени HTML.

В 2014 году работа над новым стандартом была официально завершена – HTML5, который фактически произвел революцию и привнес в HTML много нового.

Основные нововведения, привнесенные HTML5:

- HTML5 определяет новый алгоритм синтаксического анализа для создания структуры DOM;
- добавление новых элементов и тегов, таких как элементы video, audio и ряд других;
- переопределение правил и семантики уже существовавших элементов HTML;

С добавлением новых функций HTML5 стал не только новой версией языка разметки для создания веб-сайтов, но и платформой для создания приложений, и сфера его использования вышла далеко за пределы веб-среды Интернета: HTML5 также используется для создания мобильных приложений для Android, iOS, Windows Mobile и даже используются для создания настольных приложений для обычных компьютеров (особенно в Windows 8 / 8.1 / 10). HTML 5 применяется в основном в двух значениях:

- HTML 5 как обновленный язык разметки гипертекста, некоторое развитие предыдущей версии HTML 4;
- HTML 5 как новая мощная платформа для создания веб-приложений, которая включает не только непосредственно язык разметки гипертекста, обновленный HTML, но и язык программирования JavaScript и каскадные таблицы стилей CSS3;

Следует отметить, что всегда существовал разрыв между спецификацией HTML5 и использованием этой технологии в веб-браузерах. Большинство браузеров начали внедрять стандарты HTML5 еще до их официального выпуска. Пока что последние версии браузеров поддерживают большинство функций HTML5 (Google Chrome, Firefox, Opera, Internet Explorer 11, Microsoft Edge). В то же время, многие старые браузеры, такие как Internet Explorer 8 и более ранние версии, не используют стандарт, Internet Explorer 9, 10 используют только частично [13].

3.2.4 CSS3

CSS (каскадные таблицы стилей) – это формальный язык, используемый для описания внешнего вида документа, написанного на языке разметки.

CSS3 (Каскадные таблицы стилей 3 – каскадные таблицы стилей третьего поколения) – это активно разработанная спецификация CSS. Это формальный язык, который реализован с использованием языка разметки. Самое большое издание по сравнению с CSS1, CSS2 и CSS2.1. Главной особенностью CSS3 является возможность создавать анимированные элементы без использования JS, поддержка линейных и радиальных градиентов, теней, сглаживания и многое другое.

Он в основном используется для описания и оформления внешнего вида веб-страниц, написанных с использованием языков разметки HTML и XHTML. Однако его также можно применять ко всем документам XML, например, SVG или XUL.

В отличие от предыдущих версий, спецификация разделена на модули, разработка и развитие которых являются независимыми. CSS3 основан на CSS2.1, дополняет существующие свойства и значения и добавляет новые [14].

3.2.5 Архитектура взаимодействия frontend и backend

В настоящее время существует несколько базовых архитектур, которые определяют, как взаимодействуют бэкэнд и фронтэнд [15]:

1) Серверные приложения. В этом случае HTTP-запросы отправляются непосредственно на сервер приложений, а сервер отвечает HTML-страницей. Между получением запроса и ответом сервер обычно ищет информацию в базе данных по запросу и встраивает ее в шаблон (ERB, Blade, EJS, Lenker). Когда страница загружается в браузер, HTML определяет, что отображается, CSS - как это будет выглядеть и JS - все виды специальных взаимодействий.

2) Связь с использованием AJAX. Архитектура использует для связи AJAX (Asynchronous JavaScript and XML). В архитектуре для связи используется AJAX (асинхронный JavaScript и XML). Это означает, что JavaScript, загруженный в браузер, отправляет HTTP-запрос (XHR, XML-HTTP-запрос) на странице и получает XML-ответ. Теперь вы также можете использовать формат JSON для ответов. Это означает, что сервер должен иметь конечную точку, которая отвечает на запросы с помощью кода JSON или XML. Для этого используются два примера протоколов: REST и SOAP.

3) Клиентские (одностраничные) приложения. AJAX позволяет загружать данные без обновления страницы. Больше всего это используется в таких фреймворках, как Angular

и Ember. После сборки такие приложения отправляются в браузер, и любой последующий рендеринг выполняется на стороне клиента (в браузере). Такой фронтенд общается с бэкендом через HTTP, используя JSON- или XML-ответы.

4) Универсальные/изоморфные приложения. Некоторые библиотеки и фреймворки, например, React и Ember, позволяют вам исполнять приложения как на сервере, так и в клиенте. В этом случае для связи фронтенда с бэкендом приложение использует и AJAX, и обрабатываемый на сервере HTML.

В анализируемой информационной системе используется архитектура одностраничного приложения. Для отправки ответа фронтенд использует формат JSON.

4 Анализ информационной системы

4.1 Модель вариантов использования

Модель вариантов использования представлена в виде USECASE диаграммы. Диаграмма описывает возможности использования двумя ролями: учитель и завуч.

В информационной системе у учителя имеется возможность формирования отчетов только по тому классу, в котором учитель является классным руководителем. У завуча имеется возможность формирования отчетов по всем классам независимо от того, является ли завуч классным руководителем класса или нет (рисунок 4).

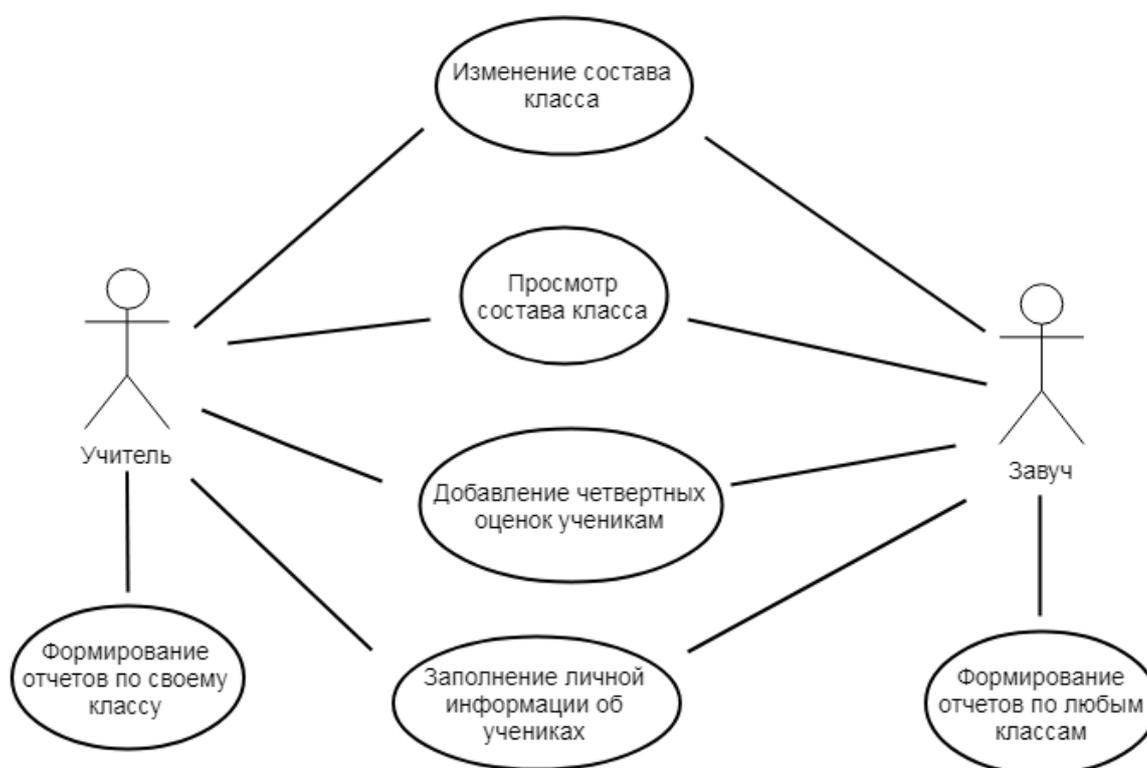


Рисунок 4 – Диаграмма вариантов использования

4.2 Логическая модель информационной системы

На схеме ниже представлена логическая модель информационной системы (рисунок 5).

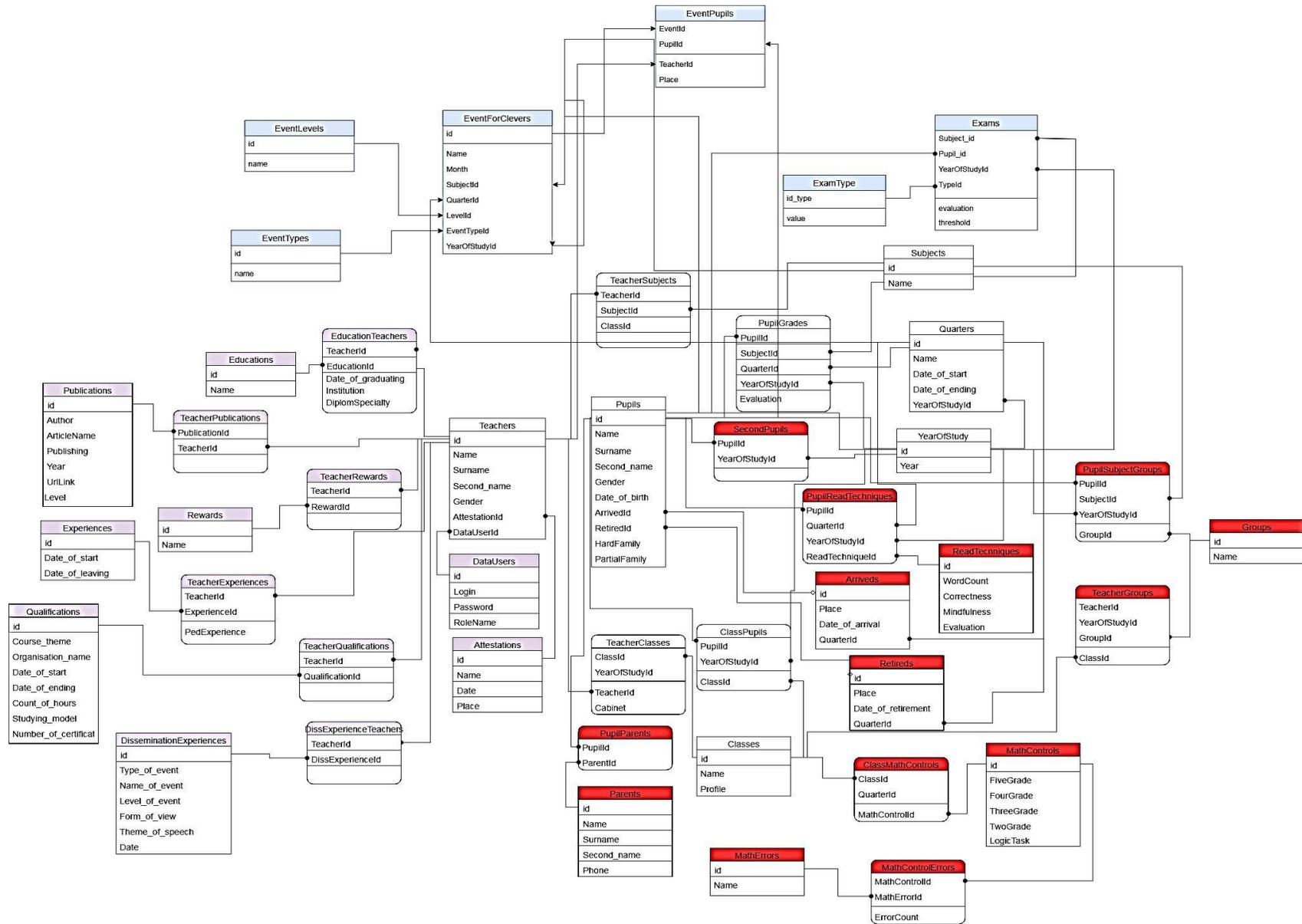


Рисунок 5 – Логическая модель информационной системы

4.3 Физическая модель

Физическая модель базы данных описывает реализацию логической модели информационной системы.

Все сущности логической модели информационной системы делятся на несколько групп:

- 1) общие сущности;
- 2) сущности модуля начальных классов;
- 3) сущности модуля учителя;
- 4) сущности модуля мероприятий;

4.3.1 Общие сущности

Таблица 2 – Сущность Teachers

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id (PK)	int	false
2	Name	nvarchar(30)	false
3	Surname	nvarchar(50)	false
4	Second_name	nvarchar(50)	false
5	Gender	nvarchar(7)	false
6	AttestationId (FK)	int	true
7	DataUserId (FK)	int	true

Сущность Teachers хранит информацию об учителях.

Таблица 3 – Сущность Pupils

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id (PK)	int	false
2	Name	nvarchar(30)	false
3	Surname	nvarchar(50)	false
4	Second_name	nvarchar(50)	false
5	Gender	nvarchar(7)	false
6	ArrivedId (FK)	int	true
7	RetiredId (FK)	int	true
8	HardFamily	bit	true
9	PartialFamily	bit	true

Сущность Pupils хранит информацию об учениках. Атрибут HardFamily указывает на тяжелую обстановку в семье ученика, атрибут PartialFamily указывает на то, что ученик воспитывается в неполной семье.

Таблица 4 – Сущность – YearOfStudy

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id (PK)	int	false
2	Year	nvarchar(50)	false

Сущность YearOfStudy хранит перечень годов обучения. Сущность необходима для формирования статистических данных и отчетов.

Таблица 5 – Сущность Quarters

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id (PK)	int	false
2	Name	tinyint	false
3	Date_of_start	date	false
4	Date_of_ending	date	false
5	YearOfStudyId	int	false

Сущность хранит информацию о сроках четвертей в учебных годах.

Таблица 6 – Сущность Classes

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id (PK)	int	false
2	Name	nvarchar(20)	true
3	Profile	nvarchar(30)	true

Сущность хранит неизменяемый перечень классов в гимназии.

Таблица 7 – Сущность TeacherClasses

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	ClassId (PK) (FK)	int	false
2	YearOfStudyId	int	false
3	TeacherId	int	false
4	Cabinet	nvarchar(20)	false

Сущность предназначена для хранения информации о том, в каком году какой учитель является классным руководителем для какого класса.

Таблица 8 – Сущность ClassPupils

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	PupilId (PK) (FK)	int	false
2	YearOfStudyId (PK) (FK)	int	false
3	ClassId (FK)	int	false

Сущность предназначена для хранения информации о том, в каком классе в каком году учится ученик.

Таблица 9 – Сущность Subjects

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	id (PK)	int	false
2	Name	nvarchar(70)	false

Сущность хранит неизменяемый перечень предметов в гимназии.

Таблица 10 – Сущность TeacherSubjects

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	TeacherId (PK) (FK)	int	false
2	SubjectId (PK) (FK)	int	false
3	ClassId (PK) (FK)	int	false

Сущность хранит информацию о том, какой учитель какой предмет ведет в определенном классе.

Таблица 11 – Сущность PupilGrades

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	PupilId (PK) (FK)	int	false
2	SubjectId (PK) (FK)	int	false
3	QuarterId (PK) (FK)	int	false
4	YearOfStudyId (PK) (FK)	int	false
5	Evaluation	int	false

Сущность хранит информацию о четвертных оценках учеников.

4.3.2 Сущности, относящиеся к модулю начальных классов

Таблица 12 – Сущность Parents

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	id	int	false
2	Name	nvarchar(50)	false
3	Surname	nvarchar(80)	false
4	Second_name	nvarchar(80)	false
5	Phone	nvarchar(30)	false

Сущность предназначена для хранения информации о родителях учеников.

Таблица 13 – Сущность PupilParents

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	PupilId (PK) (FK)	int	false
2	ParentId (PK) (FK)	int	false

Сущность для осуществления связи «многие-ко-многим» между учениками и родителями.

Таблица 14 – Сущность PupilSubjectGroups

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	PupilId (PK) (FK)	int	false
2	SubjectId (PK) (FK)	int	false
4	YearOfStudyId (PK) (FK)	int	false
5	GroupId (FK)	int	false

Сущность предназначена для распределения учеников по группам иностранного языка и группам информатики.

Таблица 15 – Сущность TeacherGroups

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	TeacherId (PK) (FK)	int	false
2	GroupId (PK) (FK)	int	false
4	YearOfStudyId (PK) (FK)	int	false
5	ClassId (FK)	int	false

Сущность предназначена для хранения информации о том, в каком году обучения для какой группы обучения преподаватель проводит занятия.

Таблица 16 – Сущность Groups

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	id	int	false
2	Name	nvarchar(50)	false

Сущность хранит неизменяемый перечень групп.

Таблица 17 – Сущность SecondPupils

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	PupilId (PK) (FK)	int	false
2	YearOfStudyId (PK) (FK)	int	false

Сущность хранит информацию об учениках, оставшихся на второй год.

Таблица 18 – Сущность PupilReadTechniques

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	PupilId (PK) (FK)	int	false
2	YearOfStudyId (PK) (FK)	int	false
3	QuarterId (PK) (FK)	Int	false
4	ReadTechniqueId (PK) (FK)	int	false

Сущность предназначена для осуществления связи «многие-ко-многим» между сущностями ученика и техники чтения в определенном учебном году и четверти.

Таблица 19 – Сущность ReadTechniques

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	id	int	false
2	WordCount	int	false
3	Correctness	int	false
4	Mindfulness	bit	false
5	Evaluation	tinyint	false

Сущность для хранения сведений по итогам проверки техники чтения.

Таблица 20 – Сущность Arriveds

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	id	int	false
2	Place	nvarchar(90)	false
3	Date_of_arrived	date	false
4	QuarterId (FK)	int	false

Сущность хранит информацию о поступлении ученика с указанием даты поступления и предыдущего места обучения.

Таблица 21 – Сущность Retireds

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	id	int	false
2	Place	nvarchar(90)	false
3	Date_of_retirement	date	false
4	QuarterId (FK)	int	false

Аналогична сущности прибытия ученика, но для убывающих учеников.

Таблица 22 – Сущность ClassMathControls

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	ClassId (PK) (FK)	int	false
2	QuarterId (PK) (FK)	int	false
3	MathControlId (FK)	int	false

Сущность хранит информацию о контрольных работах по математике в определенных четвертях для определенных классов.

Таблица 23 – Сущность MathControls

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id	tinyint	false
2	FiveGrade	tinyint	true
3	FourGrade	tinyint	true
4	ThreeGrade	tinyint	true
5	TwoGrade	tinyint	true
6	LogicTask	tinyint	true

Сущность хранит статистическую информацию об успеваемости класса при проведении контрольной по математике.

Таблица 24 – Сущность MathErrors

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	id	int	false
2	Name	nvarchar(50)	false

Сущность хранит статический список вариантов ошибок при проведении контрольной по математике.

Таблица 25 – Сущность MathControlErrors

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	MathControlId (PK) (FK)	int	false
2	MathErrorId (PK) (FK)	int	false
3	ErrorCount	int	false

Сущность для хранения статистической информации о количестве учеников, совершивших определенные ошибки при проведении контрольной по математике.

4.3.3 Сущности модуля учителя

Таблица 26 – Сущность DataUsers

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id (PK)	int	false
2	Login	nvarchar(30)	false
3	Password	nvarchar(30)	false
4	RoleName	nvarchar(15)	false

Сущность хранит информацию об учетных записях пользователей информационной системы.

Таблица 27 – Сущность Attestations

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id (PK)	int	false
2	Name	nvarchar(60)	false
3	Date	date	false
4	Place	nvarchar(100)	false
5	ActNumber	nvarchar(100)	false

Сущность хранит информацию о прохождении аттестации преподавателя с указанием даты, местом и номером акта прохождения аттестации.

Таблица 28 – Сущность EducationTeachers

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	EducationId (PK) (FK)	int	false
2	TeacherId (PK) (FK)	int	false
3	Year_of_graduating	int	true
4	Institution	nvarchar(30)	true
5	DiplomSpecialty	nvarchar(90)	false

Сущность хранит информацию об образовании преподавателя.

Таблица 29 – Сущность Educations

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id (PK)	int	false
2	Name	nvarchar(30)	true

Сущность хранит статическую информацию о типах возможного образования.

Таблица 30 – Сущность TeacherPublications

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	TeacherId (PK) (FK)	int	false
2	PublicationId (PK) (FK)	int	false

Сущность обеспечивает связь «многие-ко-многим» для сущностей учителя и научной публикации.

Таблица 31 – Сущность Publications

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id (PK)	int	false
2	Author	nvarchar(100)	false
3	ArticleName	nvarchar(90)	false
4	Publishing	nvarchar(100)	true
5	Year	int	false
6	UrlLink	nvarchar(max)	true
7	Level	nvarchar(50)	true

Сущность хранит информацию о научных публикациях.

Таблица 32 – Сущность TeacherRewards

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	TeacherId (PK) (FK)	int	false
2	RewardId (PK) (FK)	int	false

Сущность обеспечивает связь «многие-ко-многим» для сущностей учителя и награды.

Таблица 33 – Сущность Rewards

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id (PK)	int	false
2	Name	nvarchar(150)	true

Сущность хранит статический перечень наград в области образования.

Таблица 34 – Сущность TeacherExperiences

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	TeacherId (PK) (FK)	int	false
2	ExperienceId (PK) (FK)	int	false
3	PedExperience	bit	false

Сущность служит для осуществления связи «многие-ко-многим» между сущностями учителя и трудового стажа.

Таблица 35 – Сущность Experiences

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id (PK)	int	false
2	Date_of_start	date	true
3	Date_of_leaving	date	true

Сущность хранит данные о трудовом стаже с указанием даты начала и конца трудового периода.

Таблица 36 – Сущность TeacherQualifications

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	TeacherId (PK) (FK)	int	false
2	QualificationId (PK) (FK)	int	false

Сущность обеспечивает связь «многие-ко-многим» для сущностей учителя и курсов повышения квалификации.

Таблица 37 – Сущность Qualifications

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id (PK)	int	false
2	Course_theme	nvarchar(100)	true
3	Organisation_name	nvarchar(90)	true

4	Date_of_start	date	true
5	Date_of_leaving	date	true
6	Count_of_hours	int	true
7	Studing_model	nvarchar(40)	true
8	Number_of_certification	int	true

Сущность хранит данные о курсах повышения квалификации с указанием дат начала и окончания, количества часов, модели обучения и т.д.

Таблица 38 – Сущность DissExperienceTeachers

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	TeacherId (PK) (FK)	int	false
2	DissExperienceID (PK) (FK)	int	false

Сущность обеспечивает связь «многие-ко-многим» для сущностей учителя и обмена опытом.

Таблица 39 – Сущность DisseminationExperiences

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id (PK)	int	false
2	Type_of_event	nvarchar(40)	true
3	Name_of_event	nvarchar(60)	true
4	Level_of_event	nvarchar(30)	true
5	Form_of_view	nvarchar(40)	true
6	Theme_of_speech	nvarchar(50)	true
7	Date	date	true

Сущность хранит данные о выступлениях по обмену опытом.

4.3.4 Сущности модуля мероприятий

Таблица 40 – Сущность EventLevels

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	id (PK)	int	false
2	name	nvarchar(40)	false

Сущность хранит статический список уровня мероприятий для одаренных детей.

Таблица 41 – Сущность EventTypes

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	id (PK)	int	false
2	name	nvarchar(40)	false

Сущность хранит статический список типов мероприятий для одаренных детей.

Таблица 42 – Сущность EventForClevers

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	Id (PK)	int	false
2	Name	nvarchar(100)	true
3	Month	int	true
4	QuarterId (FK)	int	true
5	LevelId (FK)	int	true
6	EventTypeId (FK)	int	true
7	YearOfStudyId (FK)	int	true

Сущность хранит данные о мероприятиях для одаренных детей.

Таблица 43 – Сущность EventPupils

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	EventId (PK) (FK)	int	false
2	PupilId (PK) (FK)	int	false
3	TeacherId(FK)	int	true
4	Place	nvarchar(100)	true

Сущность обеспечивает связь «многие-ко-многим» для взаимодействия сущностей учеников и мероприятий для одаренных детей.

Таблица 44 – Сущность ExamTypes

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	id (PK)	int	false
2	name	nvarchar(40)	false

Сущность хранит статический список вариантов экзаменов.

Таблица 45 – Сущность Exams

№	Атрибут	Тип данных	Возможность принимать неопределенные значения
1	SubjectId (PK) (FK)	int	false
2	PupilId (PK) (FK)	int	false
3	YearOfStudyId (PK) (FK)	int	false
4	TypeId (PK) (FK)	int	false
5	Evaluation	int	false
6	Threshold	int	true
7	ThresholdMin	int	true

Сущность хранит данные о результатах экзаменов учеников.

5 Основные недостатки информационной системы

По итогам анализа и эксплуатации информационной системы были выявлены недостатки, делающие невозможным эксплуатацию информационной системы в продолжительном периоде времени.

Учетные записи пользователей.

Во время эксплуатации системы было обнаружено, что создание учетных записей пользователей информационной системы было заложено программистами, разрабатывающими систему, но не осуществлена реализация данного процесса.

Отсутствие регистрации пользователей является неприемлемым для информационной системы, в которой используется строгое разграничение прав пользователей, и перечень пользователей не является статическим.

Также в системе не был предусмотрен механизм удаления учетных записей.

Т.к. на одном рабочем месте в учебном заведении могут работать несколько преподавателей в разное время, необходимо иметь представление под какой учетной записью была произведена авторизация в информационной системе. Данный механизм не был предусмотрен.

При изучении базы данных был выявлен критичный недостаток хранения паролей учетных записей, а именно хранение паролей в базе данных в незащищенном виде. Т.е. пароли в базе данных не подвергались предварительному шифрованию перед записью в таблицу.

Формирование учебного года.

В ходе анализа и эксплуатации был выявлен недостаток, непосредственно связанный с учебным процессом – невозможность формирования нового учебного года и четвертей. Т.е. после окончания учебного года, в котором был произведен ввод в опытную эксплуатацию информационная система, дальнейшее ее использование не представлялось возможным.

Добавление учеников.

Добавление учеников было реализовано как заполнение полей формы для каждого отдельного ученика преподавателем после взятия класса под руководство. Этот процесс занимал очень продолжительный период времени и являлся очень трудозатратным.

Перевод учеников.

Одним из самых критических недостатков являлось отсутствие возможности перевода ученика из одного класса в другой как во время учебного года, так и по итогам

окончания учебного года. Данный функционал не был заложен в информационную систему и не имел какой-либо реализации.

Классное руководство.

Одним из замечаний, выданных эксплуатирующим персоналом гимназии №13, являлось отсутствие возможности одному преподавателю быть классным руководителем одновременно у нескольких классов (данная необходимость относилась к классному руководству младших классов). После анализа соответствующих контроллеров на стороне backend'a было выяснено, что возможность классного руководства над несколькими классами не представлялось возможным не только в один учебный год, но и невозможным в нескольких учебных годах. То есть после окончания учебного года поменять класс для руководства было невозможно.

Формирование отчетов по результатам экзаменов.

При анализе физической и логической модели базы данных было выявлено, что таблицы базы данных для нескольких сущностей, необходимых для формирования отчетов по результатам экзаменов не были созданы. После создания соответствующих таблиц в базе данных было обнаружено, что формирование данных отчетов представлялось возможным только для первого сформированного учебного года в информационной системе (в контроллерах backend'a ключевое поле учебного года в запросах к базе данных являлось константой).

Отображение трудового стажа.

В информационной системе был заложен функционал по учету и отображению трудового стажа педагогов, однако данная функция была реализована только частично (было возможно только добавление данных по трудовому стажу педагогов).

6 Устранение замечаний

Для устранения большинства критичных замечаний и разгрузки преподавателей в USECASE-модель информационной системы была добавлена еще одна роль – «Администратор» (рисунок 6).

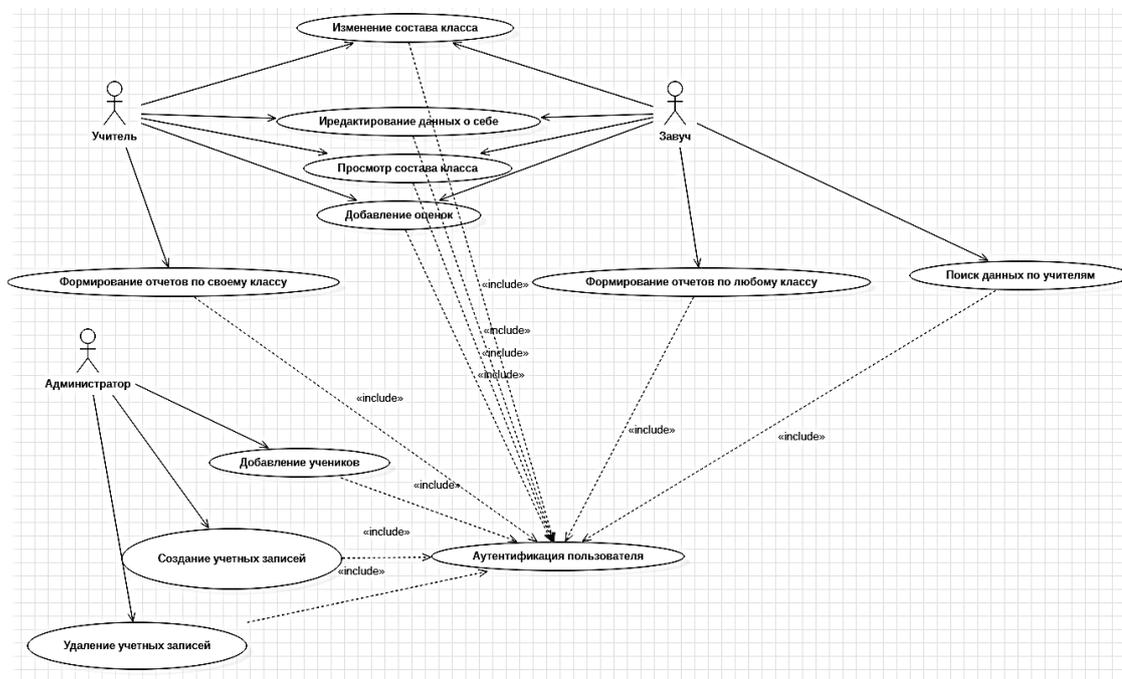


Рисунок 6 – Измененная USECASE-модель информационной системы

Для реализации функционала новой роли на стороне фронтенда информационной системы был сформирован комплекс компонентов и сервисов. При этом, отображение компонентов администратора не осуществляется при аутентификации пользователя с правами, отличными от прав администратора (рисунки 7, 8).

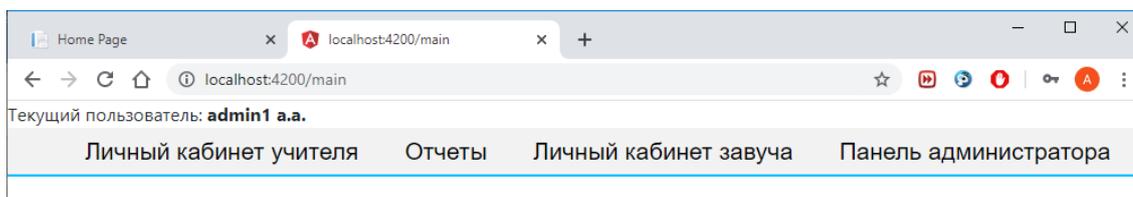


Рисунок 7 – Основное меню при аутентификации пользователя с правами администратора

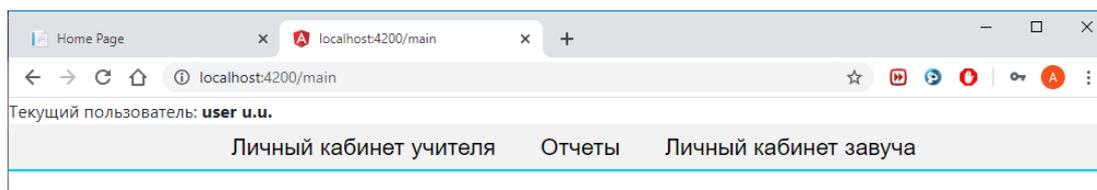


Рисунок 8 – Основное меню при аутентификации пользователя с правами завуча

На рисунках выше можно увидеть, что при работе в информационной системе в верхней части интерфейса после внесения изменений отображается текущий пользователь.

В меню администратора внесены три пункта меню: «Регистрация пользователей», «Формирование учебного года», «Импортировать поступивших учеников» (рисунок 9).

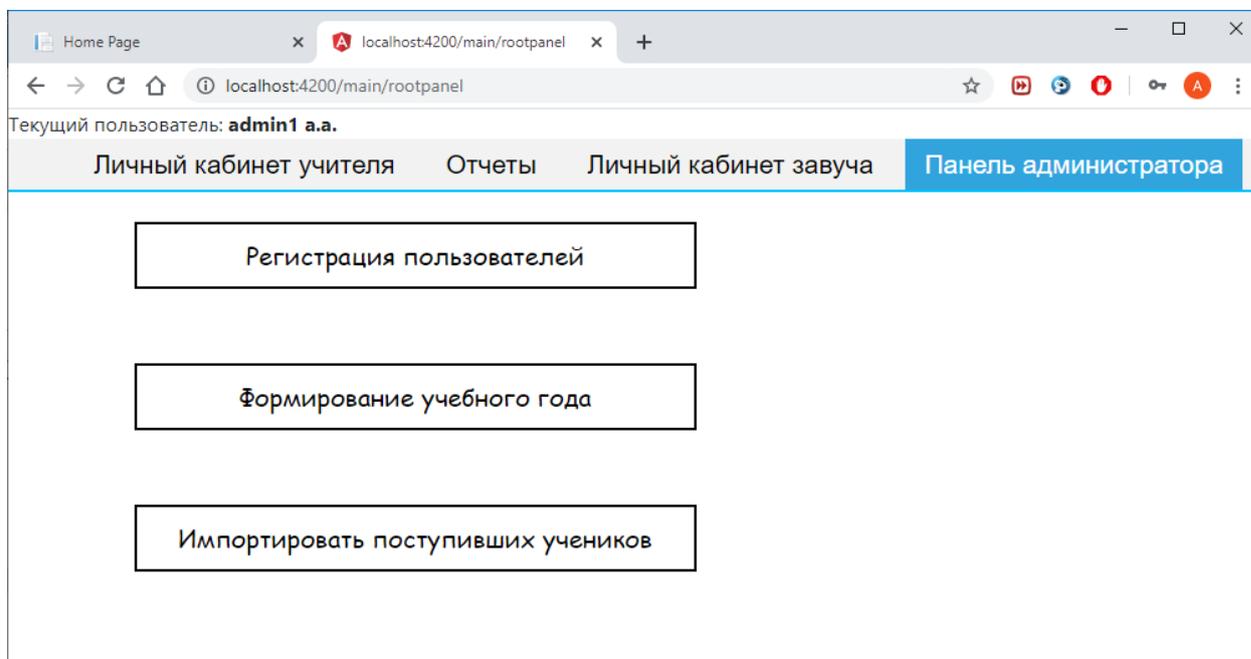


Рисунок 9 – Пункты меню администратора

В меню «Регистрация пользователей» администратора располагается таблица с учетными данными пользователей, их правами и паролями. Также на этой странице имеются поля формы для заполнения при регистрации нового пользователя (рисунок 10).

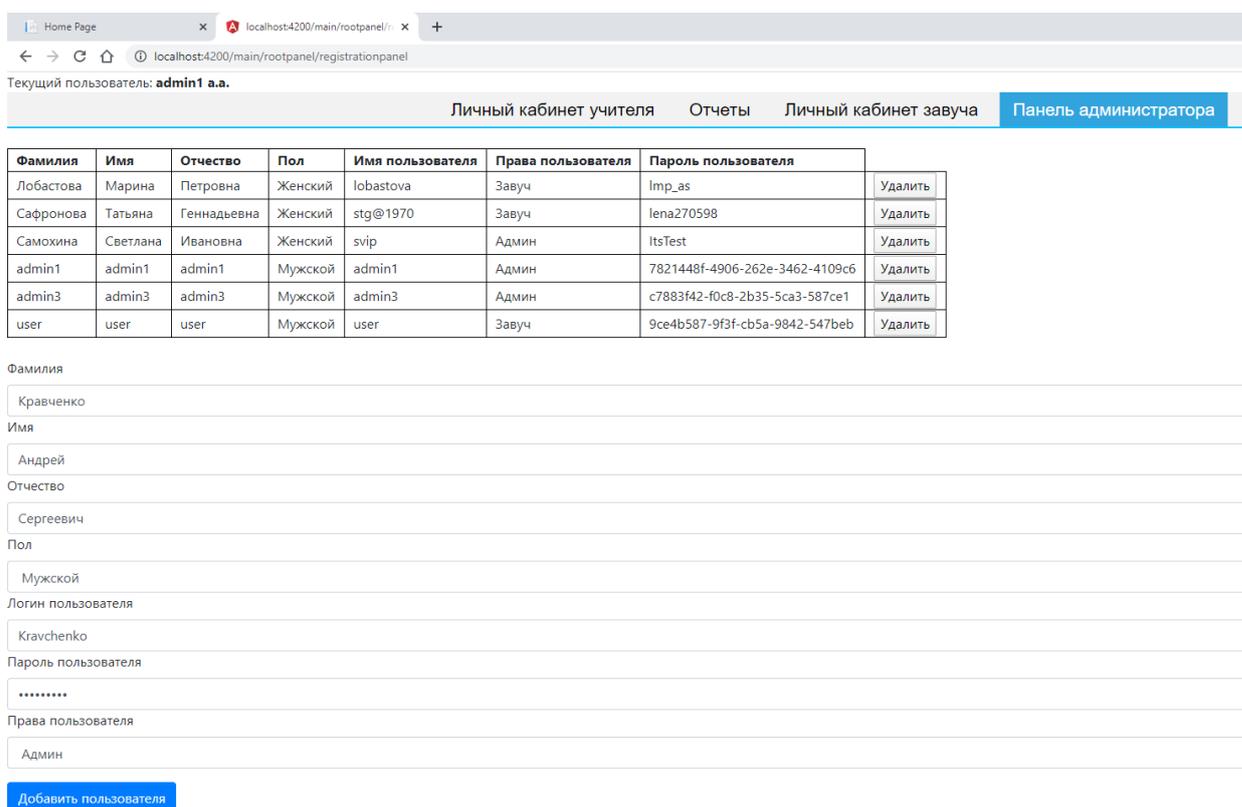


Рисунок 10 – Меню «Регистрация пользователей»

Отображение паролей пользователей не является критичным, т.к. в модуле бэкенда теперь реализован процесс хеширования паролей и использование паролей из таблицы пользователей для аутентификации в системе невозможно. Аналогично пароли теперь хранятся в базе данных в хешированном виде и их невозможно использовать для аутентификации в системе.

При необходимости, учетную запись можно удалить, нажав на соответствующую кнопку в таблице учетных записей.

В меню «Формирование учебного года» в верхней части находится таблица, содержащая записи сроков ранее созданных учебных годов и четвертей с указанием дат начала и окончания. Также на этой странице имеются поля формы для создания нового учебного года (рисунок 11).

Поля формы можно заполнять как простым вводом даты, так и выбором из всплывающей части поля.

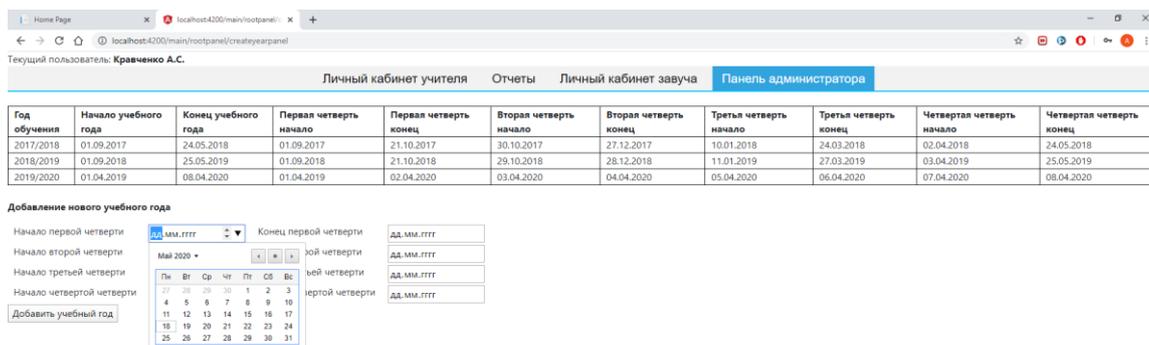


Рисунок 11 – Меню «Формирование учебного года»

В меню «Импортирование поступивших учеников» производится импортирование списка поступивших учеников из ранее созданного файла MS Excel. Для импортирования необходимо выбрать год учебный год и класс, в который импортируется список учеников. Далее нажать на кнопку «Выберите файл», в открывшемся диалоговом окне выбрать необходимый файл со списком учеников и нажать на кнопку «Импортировать» (рисунок 12).

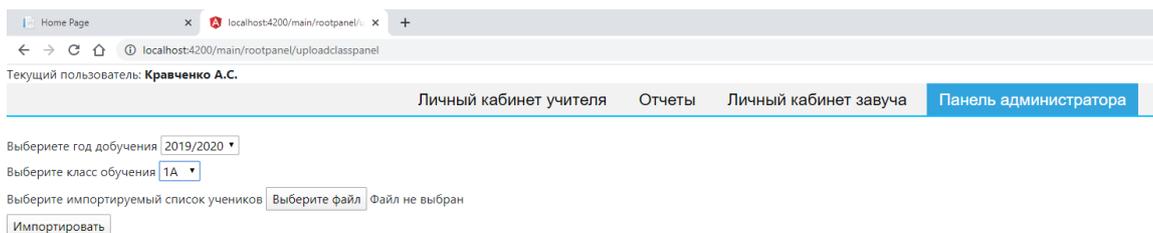


Рисунок 12 – Меню «Импортирование поступивших учеников»

Таблица со списком учеников для корректного импортирования списка должна содержать поля в определенном порядке:

- 1) Фамилия;
- 2) Имя;
- 3) Отчество;
- 4) Дата поступления.

Данная функция сокращает время, затрачиваемое педагогом, для добавления учеников в класс.

В меню «Класс» личного кабинета учителя реализован функционал выбора учебного года и класса в учебном году, в которых преподаватель является классным руководителем (рисунок 13).

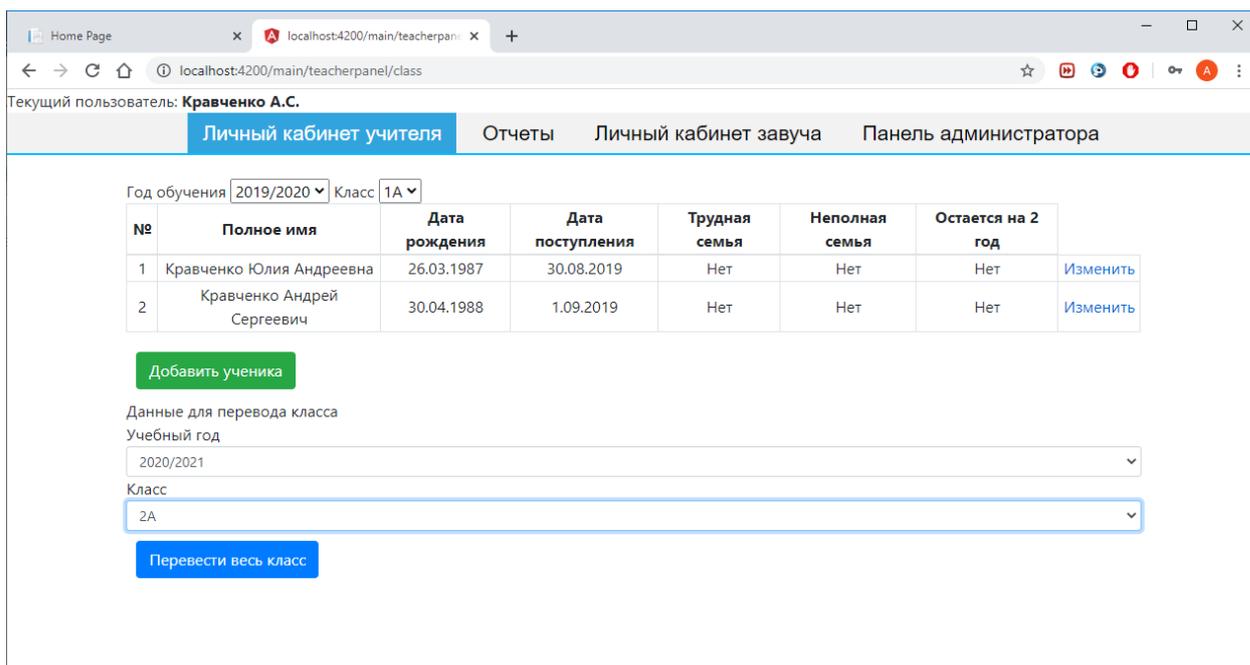


Рисунок 13 – Меню «Класс» в личном кабинете преподавателя

Также в пункт меню добавлена возможность перевода учеников всего класса в другой класс с указанием учебного года и наименования класса для перевода.

Перевод одного ученика в другой класс можно осуществить в меню «Изменение ученика», открывающегося при нажатии на ссылку «Изменить» в соответствующей строке таблицы учеников класса (рисунок 14).

Текущий пользователь: **Кравченко А.С.**

Личный кабинет учителя | **Отчеты** | Личный кабинет завуча

Изменение ученика:

Фамилия: Кравченко

Имя: Андрей

Отчество: Сергеевич

Дата рождения: 30.04.1988

Откуда прибыл(а):

Дата поступления: 01.09.2019

Мужской Женский

Изменить

Выберите год обучения для перевода: 2020/2021

Выберите класс для перевода: 2А

Перевести

Рисунок 14 – Меню «Изменение ученика»

В базу данных добавлены таблицы, необходимые для хранения данных и формирования отчетов по результатам экзаменов. Произведена корректировка контроллеров, необходимых для функционирования формирования отчетов по результатам экзаменов (рисунок 15).

Отчет по экзаменам

класс 9А

количество учащихся 2

классный руководитель Кравченко Андрей Сергеевич

учебный год 2020/2021

Предмет	Кол-во сдававших	% сдававших	«5»		«4»		«3»		«2»		успеваемость	
			количество	%	количество	%	количество	%	количество	%	абсолютная	качественная
Математика	2	100	2	100	0	0	0	0	0	0	100	100

Печать

Рисунок 15 – Сформированный отчет по результатам экзаменов

Для реализации отображения трудового стажа педагогов было принято решение написать новый контроллер на стороне бэкенда взамен неработающего (рисунки 16, 17).

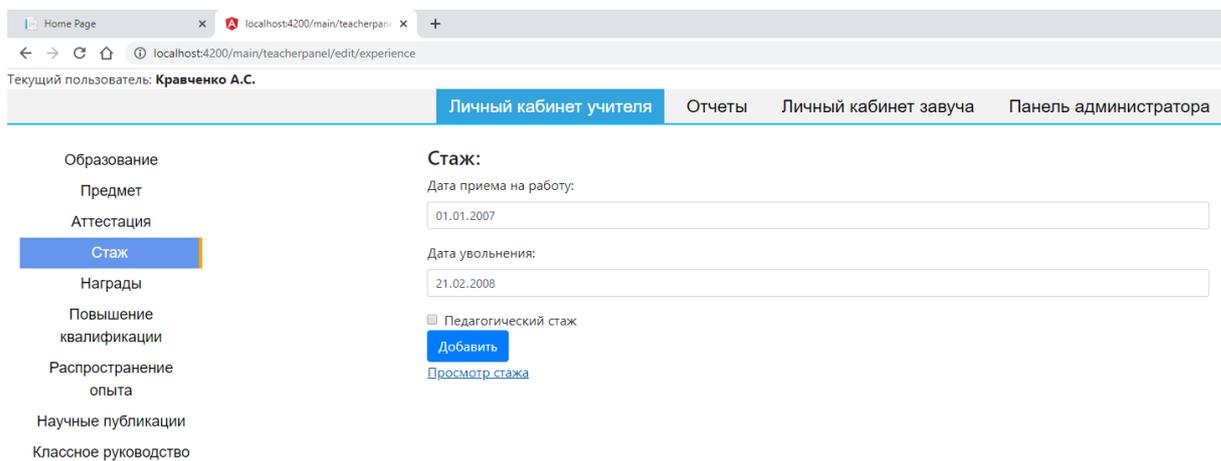


Рисунок 16 – Форма заполнения данных по трудовому стажу педагога

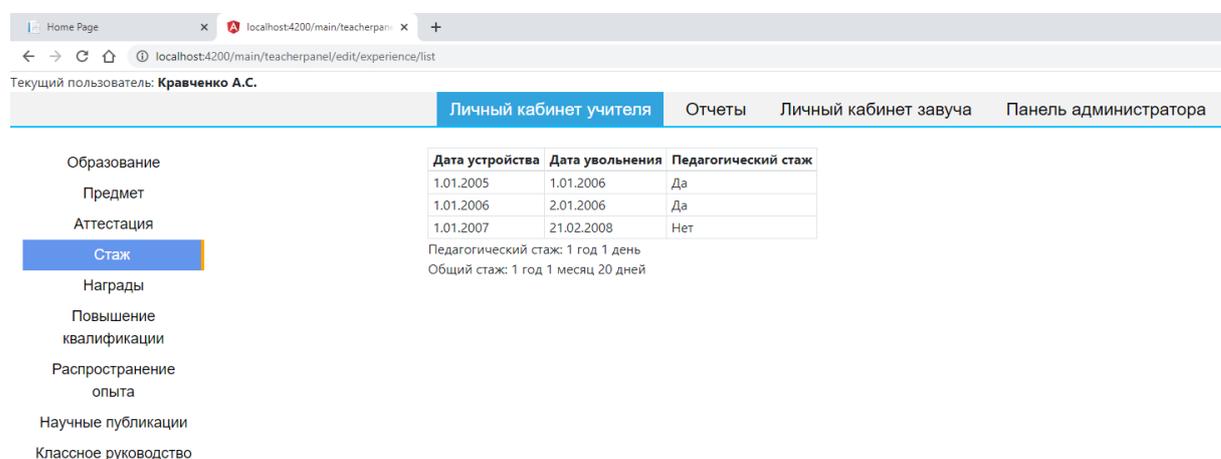


Рисунок 17 – Результат работы нового контроллера отображения рабочего стажа педагога

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы были выполнены следующие работы:

- исследование информационных систем в области образования;
- сравнительный анализ рассмотренных информационных систем;
- глубокий анализ информационной системы, разработанной для МАОУ гимназия №13 г. Томска выпускниками факультета инновационных технологий;
- исследование средств разработки и технологий, применяемых при разработке модернизируемой информационной системы;
- во время глубокого анализа модернизируемой информационной произведено выявление критических неисправностей информационной системы, не позволяющих осуществлять штатную эксплуатацию информационной системы;
- устранение критических неисправностей.

Все поставленные цели выпускной квалификационной работы выполнены в полном объеме.

Дальнейшее развитие информационной системы позволит расширить ее функционал, что позволит облегчить работу преподавателей гимназии в области документооборота и хранения статистических данных в области образования и достижений учеников и преподавателей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Группа компаний Аверс [Электронный ресурс]. – URL: <http://www.xn--80aejja5ch9a.xn--p1ai/produksiya/shkola/> (дата обращения 9.05.2020).
2. Электронная школа [Электронный ресурс]. – URL: <https://e-school.ru/> (дата обращения 9.05.2020).
3. GS-Ведомости [Электронный ресурс]. – URL: <http://gs-vedomosti.ru/about/gsv.php> (дата обращения 9.05.2020).
4. Онлайн-университет Skillbox [Электронный ресурс]. – URL: https://skillbox.ru/media/code/13_instrumentov_krutogo_backend_razrabotchika/ (дата обращения 10.05.2020).
5. Язык программирования сишарп С# [Электронный ресурс]. – URL: <http://www.nookery.ru/language-programming-c-sharp-c/> (дата обращения 9.05.2020).
6. ASP.NET MVC – веб-API [Электронный ресурс]. URL: <https://coderlessons.com/tutorials/microsoft-technologies/izuchite-asp-net-mvc/asp-net-mvc-veb-api> (дата обращения 9.05.2020).
7. Руководство по Entity Framework 6 [Электронный ресурс]. – URL: <https://metanit.com/sharp/entityframework/1.1.php> (дата обращения 13.05.2020).
8. Руководство по MS SQL Server 2017 [Электронный ресурс]. – URL: <https://metanit.com/sql/sqlserver/1.1.php> (дата обращения 13.05.2020).
9. Microsoft Visual Studio [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio (дата обращения 13.05.2020).
10. Что такое backend и frontend? [Электронный ресурс]. – URL: <https://otus.ru/nest/post/943/> (дата обращения 29.04.2020).
11. Что такое Angular и зачем его все учат? [Электронный ресурс]. – URL: <https://agilie.com/ru/blog/chto-takoe-angular-i-zachem-iegho-vsie-uchat> (дата обращения 12.05.2020).
12. Руководство по TypeScript [Электронный ресурс]. – URL: <https://metanit.com/web/typescript/1.1.php> (дата обращения 12.05.2020).
13. Руководство по HTML5 и CSS3 [Электронный ресурс]. – URL: <https://metanit.com/web/html5/1.1.php> (дата обращения 13.05.2020).
14. CSS [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/CSS> (дата обращения 10.05.2020).

15. Простыми словами о «фронтенде» и «бэкенде»: что это такое и как они взаимодействуют [Электронный ресурс]. – URL: <https://tproger.ru/translations/frontend-backend-interaction/> (дата обращения 13.05.2020).

16. Леонов В. В. Информационная система автоматизации составления отчетов в гимназии № 13 / В. В. Леонов, А. С. Колупаев, В. А. Самсонов // Инноватика - 2018 : сборник материалов XIV Международной школы-конференции студентов, аспирантов и молодых ученых, 26-27 апреля 2018 г., г. Томск, Россия. Томск, 2018. С. 385-388. – URL: <http://vital.lib.tsu.ru/vital/access/manager/Repository/vtls:000633384> (дата обращения 13.05.2020).



СПРАВКА

о результатах проверки текстового документа на наличие заимствований

Проверка выполнена в системе Антиплагиат.ВУЗ

Автор работы	Кравченко Андрей Сергеевич
Подразделение	кафедра информационного обеспечения инновационной деятельности
Тип работы	Выпускная квалификационная работа
Название работы	МОДИФИКАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ АВТОМАТИЗАЦИИ ОТЧЕТНЫХ ДОКУМЕНТОВ ПО УЧЕБНЫМ ДОСТИЖЕНИЯМ УЧАЩИХСЯ И ПРОФЕССИОНАЛЬНЫМ ДОСТИЖЕНИЯМ ПЕДАГОГОВ В МАОУ ГИМНАЗИЯ №13 Г. ТОМСКА
Название файла	ВКР Кравченко А.С.18511-исправлено (1).docx
Процент заимствования	23.42 %
Процент самоцитирования	0.00 %
Процент цитирования	4.57 %
Процент оригинальности	72.01 %
Дата проверки	20:19:15 17 июня 2020г.
Модули поиска	Модуль поиска ИПС "Адилет"; Модуль выделения библиографических записей; Сводная коллекция ЭБС; Модуль поиска "Интернет Плюс"; Коллекция РГБ; Цитирование; Модуль поиска переводных заимствований; Модуль поиска переводных заимствований по eLibrary (EnRu); Модуль поиска переводных заимствований по интернет (EnRu); Коллекция eLIBRARY.RU; Коллекция ГАРАНТ; Коллекция Медицина; Диссертации и авторефераты НББ; Модуль поиска перефразирований eLIBRARY.RU; Модуль поиска перефразирований Интернет; Коллекция Патенты; Модуль поиска общеупотребительных выражений; Модуль поиска "ТГУ"; Кольцо вузов
Работу проверил	Гольцова Полина Андреевна ФИО проверяющего
Дата подписи	<i>18.06.2020</i>  Подпись проверяющего

Чтобы убедиться
в подлинности справки,
используйте QR-код, который
содержит ссылку на отчет.



Ответ на вопрос, является ли обнаруженное заимствование
корректным, система оставляет на усмотрение проверяющего.
Предоставленная информация не подлежит использованию
в коммерческих целях.