

МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное  
образовательное бюджетное учреждение  
«Слободской колледж педагогики и социальных отношений»

Дипломный проект допущен к защите  
Заместитель директора по  
воспитательной и методической  
работе

\_\_\_\_\_ к.п.н., Глазырина Т. Г.  
« \_\_\_\_\_ » \_\_\_\_\_ 2024 г.

ДИПЛОМНЫЙ ПРОЕКТ  
РАЗРАБОТКА СИСТЕМЫ ЭЛЕКТРОННОГО ОБУЧЕНИЯ ДЛЯ  
СРЕДНЕЙ ШКОЛЫ ПГТ ВАХРУШИ

Выполнил  
Платунов Павел Андреевич  
студент специальности 09.02.07  
Информационные системы  
и программирование  
группа 21П-1  
Форма обучения: очная

\_\_\_\_\_  
(подпись)

Руководитель:  
Калинин Арсений Олегович

\_\_\_\_\_  
(подпись)

Дипломный проект защищен  
« \_\_\_\_\_ » \_\_\_\_\_ 2024 г.  
Оценка \_\_\_\_\_  
Секретарь ГЭК \_\_\_\_\_

Слободской  
2024

Нормоконтроль:

Дата: \_\_\_\_\_

---

Подпись

---

Расшифровка подписи

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	4
ГЛАВА 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ .....	7
1.1 Анализ предметной области .....	7
1.2 Техническое задание .....	11
ВЫВОД ПО ГЛАВЕ 1 .....	16

## ВВЕДЕНИЕ

В условиях стремительной цифровизации и роста потребности в автоматизации любых производственных процессов, а также повышение эффективности работы с большими объемами информации, возрастает степень компьютеризации различных систем. И система образования не является исключением. А благодаря большой доступности интернета становится все более актуальным обучение дистанционно при помощи СЭО.

Система электронного обучения (СЭО) — это платформа для онлайн-обучения. СЭО помогает организовать весь цикл образовательного процесса: от разработки контента до проверки знаний учеников. Интерактивная платформа позволяет хранить все обучающие материалы в одной виртуальной среде.

СЭО значительно упрощает процесс обучения, позволяя школьникам получать и сдавать задания онлайн, что экономит время и усилия, избавляя от необходимости личного получения и сдачи задания. Также она позволяет ученикам во время учебного процесса выполнять задания в электронном формате, не используя сторонние ресурсы для их получения. Преподаватели же получают возможность легко создавать, распространять и проверять задания, а также оперативно выставлять оценки, оптимизируя свою работу.

Гибкость СЭО позволяет легко организовать дистанционное обучение, например, в случае карантина или нестандартного расписания преподавателя, обеспечивая непрерывность образовательного процесса и равномерную учебную нагрузку для учеников. Благодаря СЭО, управление образовательными ресурсами становится более эффективным, а доступ к качественному образованию становится доступнее для всех, независимо от обстоятельств. Внедрение таких современных технологий в образовательные учреждения подчеркивает актуальность инновационного подхода к обучению и созданию комфортной среды для всех участников образовательного

процесса. Примерами наиболее эффективных СЭО для комплексного управления учебным процессом являются Moodle или eFront.

Сегодня у каждого есть мобильный телефон или компьютер поэтому приложение должно поставлять одновременно на 2 платформы для этой цели отлично подходит веб приложение ведь оно позволяет получать прямой доступ к данным, не нагружая само устройство так как все расчеты производиться на серверных мощностях, а также избежать скачивания и регулярных обновлений приложения пользователем так как доступ можно получить из любого браузера.

В качестве языков программирования будет использоваться как C#, HTML, CSS, JavaScript. C# будет выступать для обработки данных, полученных с сервера базы данных при помощи технологии ADO.NET, HTML для написания графического представления сайта, а CSS для создания единого стиля и JavaScript для анимирования элементов сайта.

В качестве системы управления базой данных (СУБД) будет использоваться SQL Server на котором будет храниться вся база данных задаваемых заданий.

Объект исследования: процесс разработки программного обеспечения для дистанционного обучения учеников в удобном электронном формате.

Предмет исследования: разработка программного обеспечения для дистанционного обучения школьников.

Цель дипломного проекта - заключается в разработке и внедрении электронной системы обучения в школу, для обеспечения непрерывности образовательного процесса и создания более равномерной учебной нагрузки на учеников путем переноса части занятий в дистанционный формат.

Задачи исследования:

- Описать предметную область.
- Разработать технического задание на создание программного продукта.
- Описать архитектуру программы.

- Описать алгоритмы и функционирование программы.
- Провести тестирование и опытную эксплуатацию.
- Разработать руководство оператора

Практическая значимость: заключается в создании программного обеспечения для школы с целью обеспечить непрерывность образовательного процесса и равномерность учебной нагрузки на учеников.

Методы исследования - системный анализ и функциональное моделирование.

## ГЛАВА 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

### 1.1 Анализ предметной области

Электронная система обучения разрабатывается для КОГОВУ Средней Школы пгт. Вахруши.

Данная система разрабатывается с целью более эффективной и удобной выдачи и проверке заданий ученикам школы.

Системы электронного обучения (СЭО) представляет собой организованную структура, которая предлагает выбор класса, для которого составляется задание и предмет, по которому будет выложено задание. Сами задания представляют из себя простые страницы с текстом задания и полем для прикрепления ответа (различные файлы, тесты, ссылки на внешние ресурсы).

Цель создания СЭО состоит в обеспечении непрерывности образовательного процесса и создании равномерной учебной нагрузки на учеников случаях выхода класса на карантин, занятости учителя или для простой оптимизации выдачи и проверки задания.

Работать с СЭО могут:

Администратор – администрирует работу системы, изменяет структуру СЭО, добавляет, изменяет, удаляет пользователей, раздает роли пользователям, имеет возможность создания резервных копий и восстановление системы при помощи резервной копии.

Учитель – добавляет, изменяет, удаляет содержание своего курса (работает с заданиями), скачивает, проверяет и оставляет комментарии к работе ученика.

Ученик – просматривает задания своего курса, прикрепляет решение задания, а также просматривает оценку, данную учителем.

Для удобного представления актеров я использовал UML диаграмму вариантов использования (Рис. 1.1).

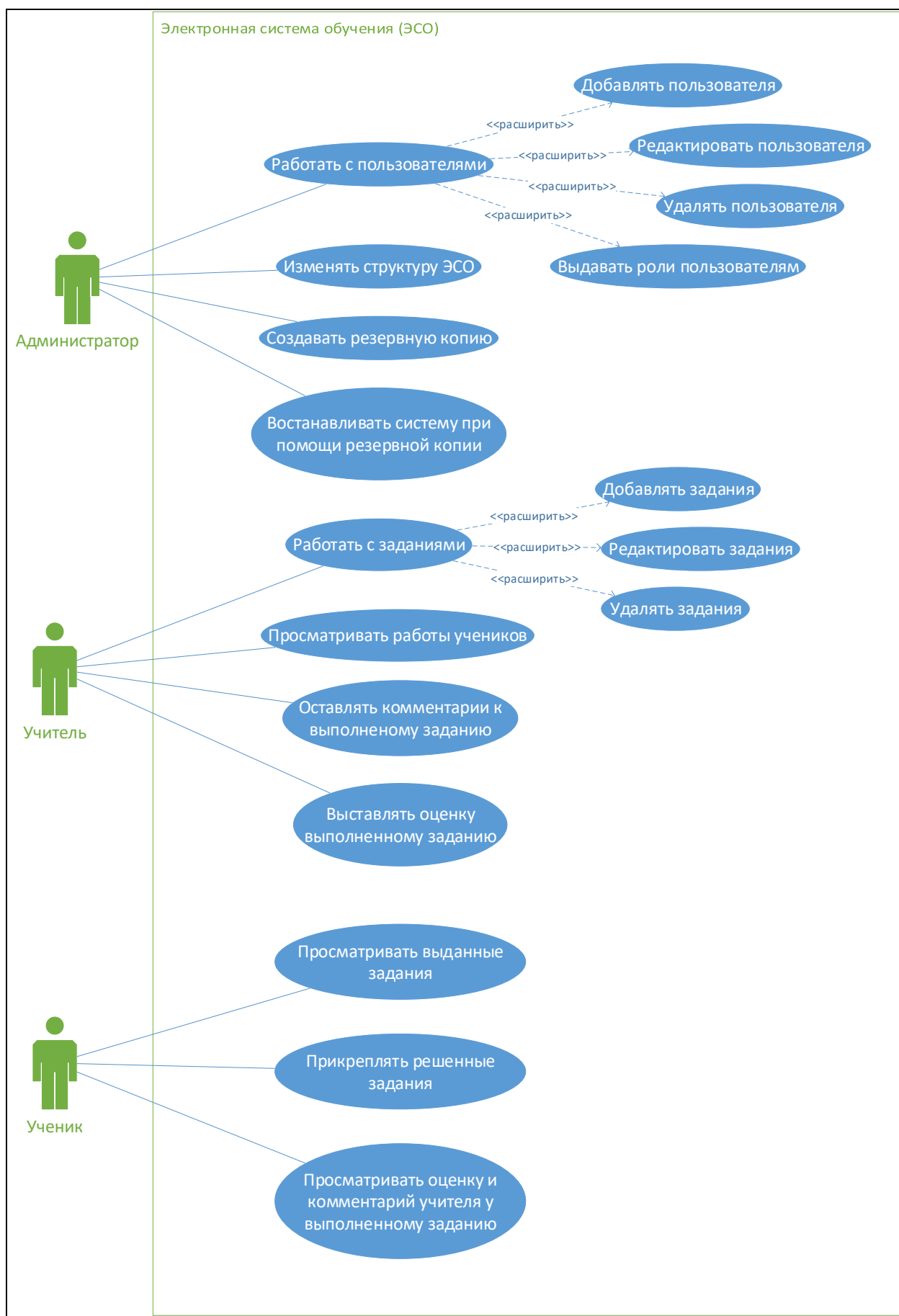


Рисунок 1.1 – UML диаграмма вариантов использования



После проведения анализа предметной области мною были выделены следующие объекты и сущности. На основании которых была создана ER – диаграмма (Рис. 1.4).

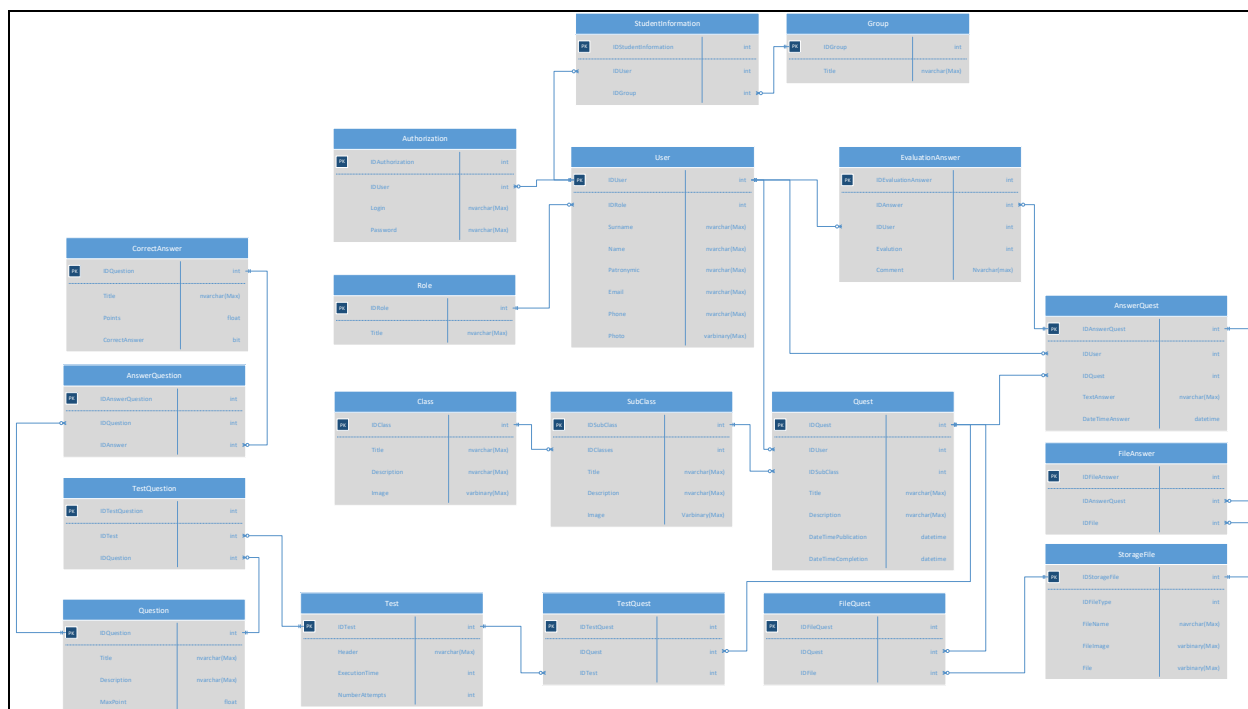


Рисунок 1.4 – ER – диаграмма

Анализ программ аналогов.

Moodle (Modular Object-Oriented Dynamic Learning Environment) — это система управления онлайн-обучением, которая распространяется бесплатно и имеет открытый исходный код (Рис. 1.2).

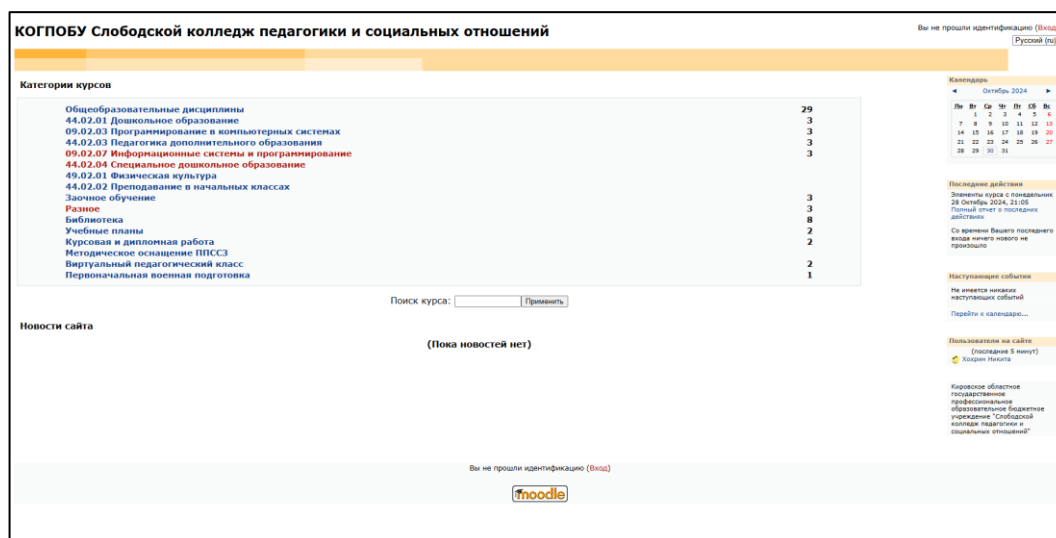


Рисунок 1.2 – Система электронного обучения Moodle

### Достоинства:

- Бесплатная.
- Открытый исходный код.
- Почти неограниченные возможности кастомизации.
- Возможность установки на любой сервер.
- Наличие широкого функционала для обеспечения процесса обучения.

- Возможность обладать несколькими ролями в системе.
- Возможность работы с резервными копиями из интерфейса.

### Недостатки:

- Дизайн решения из коробки.
- Отсутствие профессиональной технической поддержки.
- Необходимость собирать систему с нуля.
- Сложность системы (установка, настройка, работа с системой учителям)
- Требуется технических компетенций в области управления вебсайтом от преподавателя.

eFront — это гибкая корпоративная платформа управления обучением с открытым исходным кодом (Рис 1.3).

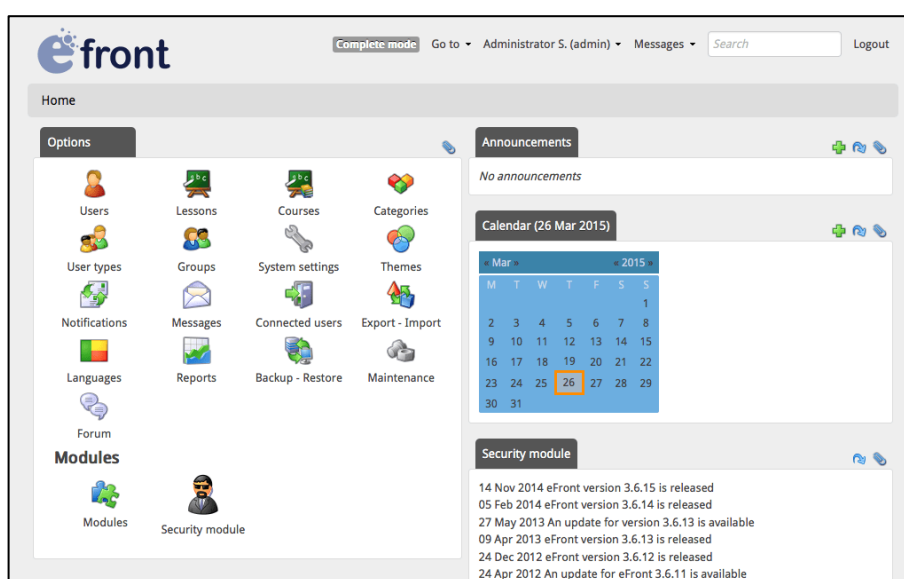


Рисунок 1.3 – Электронная система обучения eFront

Достоинства:

- Простота интуитивно понятный интерфейс.
- Открытый исходный код.
- Применение технологий Ajax, Unicode, LDAP, поддержка SCORM, возможность обучать на нескольких языках.
- Наличие широкого функционала для обеспечения процесса обучения.

Недостатки:

- Отсутствие инструментов разработки системы.
- Невозможность одновременно обладать несколькими ролями.
- Проблемы с переносом данных с других платформ на eFront.
- Высокая стоимость.

## **1.2 Техническое задание**

Техническое задание разрабатывалось на основании заявки (Приложение 1) и требований заказчика, программы «Электронная система обучения», анализа предметной области и с учетом требований ГОСТ 19.201-78 [5].

Наименование программы – «Электронная система обучения». Программа предназначена для обеспечения непрерывности образовательного процесса и создании равномерной учебной нагрузки на учеников случаях выхода класса на карантин, занятости учителя или для простой оптимизации выдачи и проверки задания.

Разработка программы ведется на основании учебного плана и перечня тем утвержденных на заседании предметно цикловой комиссии информатики и программирования.

Функциональным назначением программы является дистанционного обучения учеников в удобном электронном формате.

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

- Управление пользователями;
- Управление заданиями;
- Управление тестами;
- Оценка и комментирование работ учеников;
- Просмотр выданных заданий;
- Просмотр оценки и комментария учителя;
- Скачивание файлов, загруженных учителем или учеником;
- Загрузка файлов учителем или учеником;
- Изменение структуры электронной системы обучения;
- Резервное копирование системы;

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- организация бесперебойного питания технических средств;
- использование лицензионного программного обеспечения;
- отсутствие вредоносного программного обеспечения, наличие антивирусной программы;
- соблюдение правил и требований по эксплуатации технических средств.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 5 минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу пользователя без предоставления ему административных привилегий.

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

В состав технических средств должен входить IBM-совместимый персональный компьютер (ПЭВМ), включающий себя:

- процессор с тактовой частотой, 1 ГГц, не менее;
- оперативную память объемом 512 Мб, не менее;
- жесткий диск со свободным местом 500 Мб, не менее;
- монитор, с разрешением экрана 1270\*720, не менее;
- компьютерная мышь;
- клавиатура;

сервер, включающий в себя:

- 4 ядерный процессор с тактовой частотой, 2.5 ГГц, не менее;
- оперативную память объемом 8 Гб, не менее;
- жесткий диск со свободным местом 25 Гб, не менее;
- монитор, с разрешением экрана 1270\*720, не менее;
- компьютерная мышь;
- клавиатура;

Исходные коды программы должны быть реализованы на языках C#, HTML, CSS, JavaScript. В качестве интегрированной среды разработки программы должна быть использована среда программирования Microsoft Visual Studio 2022 и Microsoft SQL Server 2014 Management Studio.

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 7/8/10/11.

Программное обеспечение поставляется в виде изделия на USB-флэш накопителе.

Требования к транспортировке и хранению должны соответствовать условиям эксплуатации носителей, на которых находится программный продукт.

Программа должна обеспечивать взаимодействие с пользователем посредством графического пользовательского интерфейса.

Предварительный состав программной документации включает в себя следующие документы:

- техническое задание;
- руководство оператора.

Разработка должна быть проведена в следующие стадии и этапы:

1. Анализ требований:

На стадии анализ требований формулируются цели и задачи проекта. Создается основа для дальнейшего проектирования

2. Проектирование:

На стадии проектирование должны быть выполнены перечисленные ниже этапы работ:

- разработка программной документации;

На этапе разработка программной документации должна быть выполнена разработка технического задания.

При разработке технического задания должны быть выполнены перечисленные работы: постановка задачи, определение и уточнение требований к техническим средствам, определение требований к программе, определение стадий, этапов и сроков разработки программы и документации на нее, выбор языков программирования.

- разработка алгоритма программы;

На этапе разработки алгоритма программы должен быть разработан алгоритм работы программы.

- кодирование;

На стадии кодирования происходит реализация алгоритмов в среде программирования.

- тестирование и отладка.

На стадии тестирования и отладки происходит проверка алгоритмов, реализованных в программе на работоспособность в различных ситуациях. Исправление выявленных ошибок, повторное тестирование.

Приемо-сдаточные испытания должны проводиться при использовании технических средств. Приемка программы заключается в проверке работоспособности программы путем ввода реальных или демонстрационных данных.

Во время приемки работы разработчик предоставляет программу и документацию, которая к ней прилагается. Проводятся испытания программы, при успешных испытаниях программа вводится в эксплуатацию. При ошибках, недопустимых для успешной работы программного продукта – отправляется на доработку.

Было описано техническое задание, содержащее в себе информацию о программном продукте, его функциях, эксплуатации и требования, которые должны учитываться при создании программы и документации к ней.

## **ВЫВОД ПО ГЛАВЕ 1**

В этой главе была описана аналитическая часть, содержащая в себе описание учет спортивного инвентаря, который был изучен перед созданием программного продукта, а также техническое задание, содержащее в себе информацию о программном продукте, его функциях, эксплуатации и требования, которые должны учитываться при создании программы и документации к ней.

Определена значимость и требования к будущему программному обеспечению. В техническом задании были определены основные требования к программному продукту и функциональные характеристики, а также состав программной документации.



## ГЛАВА 2. КОНСТРУКТОРСКАЯ ЧАСТЬ

### 2.1 Архитектура проекта

Разработанный программный продукт поставляется в виде проекта или докера для размещения на сервере, также вместе с программным продуктом поставляется файлы миграции базы данных и SQL-скрипт для создания базы данных.

На основании технического задания нами была создана база данных в системе управления базами данных (СУБД) Microsoft SQL Management Studio 2014. На схеме базы данных (Рисунок 2.1) представлены основные таблицы.

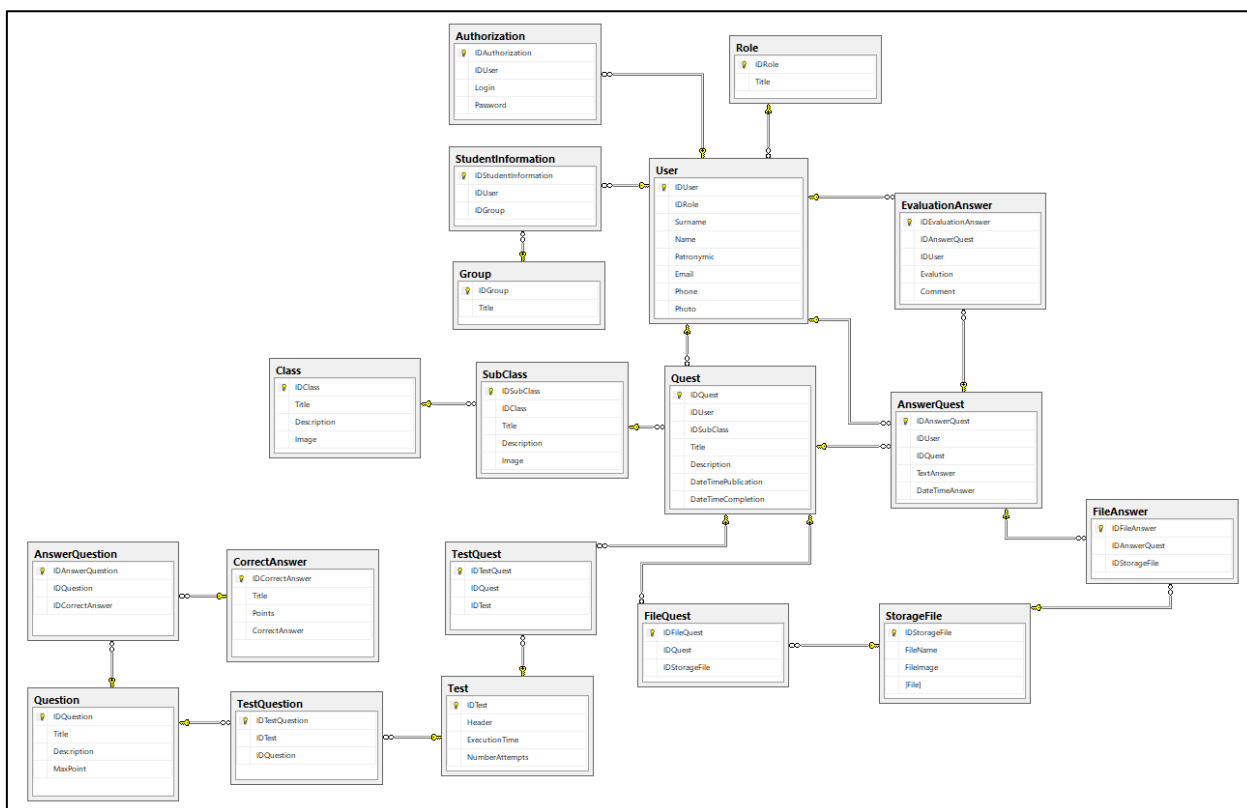


Рисунок 2.1 – Схема базы данных

На указанной схеме базы данных (Рисунок 2.1) отображены объекты и атрибуты, а также связи между ними. Данная база данных используется программой для хранения данных: пользователей, заданий, ответов на задания, тестов, оценок.

Приведем подробное описание нескольких таблиц. В таблице «User» содержатся личные данные о пользователях системы (Таблица 1). В таблице «Quest» содержится информация об выданных заданиях (Таблица 2). В таблице «AnswerQuest» содержится информация о ответах студентов на задания (Таблица 3). В таблице «Test» содержится информация о тестах (Таблица 4).

Таблица 1 – Структура таблицы «User»

Ключ	Наименование поля	Тип данных	Разрешить значение NULL
ПК	IDUser	Int	No
БК	IDRole	Int	No
	Surname	Nvarchar(Max)	No
	Patronymic	Nvarchar(Max)	No
	Email	Nvarchar(Max)	Yes
	Phone	Nvarchar(Max)	No
	Photo	Varbinary(Max)	Yes

Таблица 2 – Структура таблицы «Quest»

Ключ	Наименование поля	Тип данных	Разрешить значение NULL
ПК	IDQuest	Int	No
БК	IDSubClass	Int	No
	Title	Nvarchar(Max)	No
	Description	Nvarchar(Max)	No
	DateTimePublication	Datetime	No
	DateTimeCompletion	Datetime	Yes

Таблица 3 – Структура таблицы «AnswerQuest»

Ключ	Наименование поля	Тип данных	Разрешить значение NULL
ПК	IDAnswerQuest	Int	No
БК	IDUser	Int	No
БК	IDQuest	Int	No
	TextAnswer	Nvarchar(Max)	Yes
	DateTimeAnswer	Datetime	No

Таблица 4 – Структура таблицы «Test»

Ключ	Наименование поля	Тип данных	Разрешить значение NULL
ПК	IDTest	Int	No
	Header	Nvarchar(Max)	No
	ExecutionTime	Int	Yes
	NumberAttempts	Int	Yes

Доступ к базе данных обеспечивается с помощью технологии EntityFrameworkCore, данная технология предоставляет доступ к базе данных.

На рисунке 2.2 показан схема работы программы.

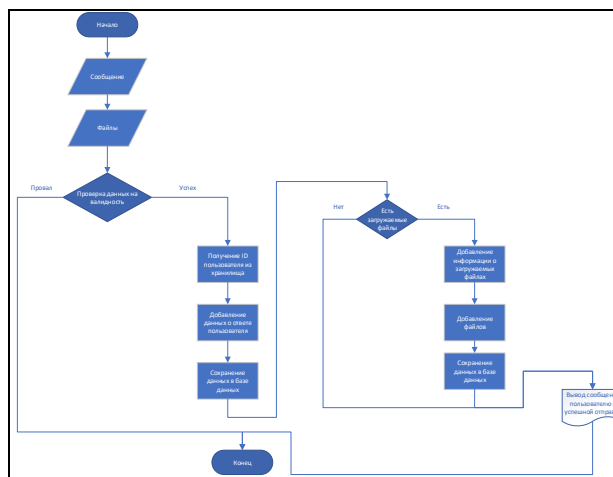


Рисунок 2.2 – Схема работы программы

С полным кодом программы можно ознакомиться в приложении 1.

В программном продукте создано 7 страниц: страница авторизации, страница отображения классов (Index – стартовая страница), страница для отображения предметов, страница для отображения заданий, страница для отображения подробного описания задания, страница для отображения оценок ученика, страница для отображения профиля.

Авторизация в приложении реализована через файлы Cookie, на данном окне пользователь вводит логин и пароль. Данная страница нужна для предотвращения доступа пользователей без авторизации к данным сайта и его инструментам управления.

На странице профиля пользователь может изменить данные своей учетной записи или сменить логин и пароль.

Страницы классов, предметов и заданий служат для отображения соответственно доступных классов, предметов выбранного класса и заданий выбранного предмета.

Страница оценок содержит оценки, полученные за выполнение заданий.

Страница подробного просмотра задания состоит из описания задания, даты выполнения задания, даты до которой необходимо сдать задание и формы для отправки результата работы ученика, состоящей из поля сообщения и поля загрузки файлов.

Для обработки запросов, отправленных веб сайтом на стороне сервера, используются методы OnGet для получения данных от сервера и OnPost для отправки данных на сервер.