



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)»**

Кафедра 319 «Системы интеллектуального мониторинга»

Курсовая работа по дисциплине:

«Web-программирование»

Система управления контентом «Электронный журнал»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

08335 – 01 81 01

Листов 28

Выполнил: студент группы
МЗО-335Б-17

(подпись) Милаков А.В.

« ____ » _____ 2020 г.

Принял:

(подпись) Коновалов К.А.,
старший
преподаватель

« ____ » _____ 2020 г.

Оглавление

1. Введение.....	3
2. Основание для разработки	3
3. Требования к приложению.....	3
4. Описание алгоритма	6
5. Организация данных.....	17
6. Результаты работы программ и их оценка	24
Заключение	26
Перечень сокращений.....	26
Список используемой литературы	27
Приложение А	28

1. ВВЕДЕНИЕ

Целью данной курсовой работы является разработка системы управления контентом «ElectronicJournal_WEB», являющееся частью информационной системы «Электронных журнал».

2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ

Основание для разработки является учебный план по дисциплине «Web-программирование»

3. ТРЕБОВАНИЯ К ПРИЛОЖЕНИЮ

Требования к базе данных:

- хранение информации о пользователях приложения;
- хранение информации о составах учебных групп;
- хранение информации о проведенных занятиях;
- хранение информации о успеваемости студентов.

Требования к пользовательскому интерфейсу:

- авторизация в приложении;
- учет успеваемости студентов;
- формирование отчетов успеваемости;
- просмотр расписания преподавателя;
- просмотр личной успеваемости.

Требование к инструментарию при разработке

Во время разработки использовался следующий инструментарий:

- Microsoft Visual Studio Community 2019 – полнофункциональная интегрированная среда разработки для написания, отладки, тестирования и развертывания кода на любой платформе
- Entity Framework Core – объектно-ориентированная технология от компании Microsoft для доступа к данным. Является ORM-

инструментом (object-relational mapping - отображения данных на реальные объекты), которая позволяет работать с базами данных;

- ASP.NET Core – свободно-распространяемый кросс-платформенный фреймворк для создания веб-приложений с открытым исходным кодом.

Технические требования при разработке:

- Windows 10 версии 1703 и выше: Домашняя, Pro, для образовательных учреждений и Корпоративная;
- процессор с тактовой частотой не ниже 1,8 ГГц. Рекомендуется использовать как минимум двухъядерный процессор;
- 2 ГБ ОЗУ (рекомендуется 8 ГБ ОЗУ);
- место на жестком диске: от 20 до 50 ГБ свободного места;
- скорость жесткого диска: для повышения производительности установите Windows и Visual Studio на твердотельный накопитель;
- видеоадаптер с минимальным разрешением 720p (1280 на 720 пикселей); для оптимальной работы Visual Studio рекомендуется разрешение WXGA (1366 на 768 пикселей) или более высокое;
- манипулятор типа «мышь» и «клавиатура».

Системные требования при эксплуатации

Требования к клиентской машине:

- браузер;
- процессор частотой 1.8 ГГц;
- ОЗУ 512 Мб;
- монитор SVGA, поддерживающий разрешение не менее 1366x768;
- манипулятор типа «мышь» и клавиатура.

Требования к серверу:

- Microsoft SQL Server — система управления реляционными базами данных, разработанная корпорацией Microsoft;
- .NET Framework;
- Windows 10 TH1 1507 или более поздней версии / Windows Server 2016 или более поздней версии
- для SQL Server требуется как минимум 6 ГБ свободного места на диске;
- монитор Super VGA с разрешением 800x600 пикселей или более высоким;
- для обеспечения оптимальной производительности требуется не менее 4 ГБ;
- процессор x64 с тактовой частотой 1,4 ГГц. Рекомендуется: 2,0 ГГц и выше.

4. ОПИСАНИЕ АЛГОРИТМА

Для разработки системы управлением контентом была выбрана кроссплатформенная технология для создания веб-приложений ASP.NET Core с использованием архитектурного паттерна MVC (Model-View-Controller).

Концепция паттерна MVC предполагает разделение приложения на три компонента.

Модель – это связанная нашим веб-приложением сущность, класс, который описывает определенные данные. Модели можно разделить на два типа:

- Анемическая модели – модель, не имеющая поведения, а хранящая только набор характеристик для этой модели свойств;
- «Полная модель» (Rich Domain / Thick Model) – это модель, которая помимо хранения набора характеристик для нее свойств, определяет так же и поведение.

Представления – это характерный для данного сетевого ресурса (сайта) набор html-документов. Другими словами, это все связанные с сайтом веб-страницы. Подобно первой технологии ASP, представления ASP.NET Core могут обладать C#-код-вставками, которые будут производить определенную работу (к примеру, помогают серверу генерировать определенные html-теги с определенными значениями). Так же, помимо прочего, представления могут включать статические файлы, такие как стили и JavaScript – коды

Контроллер – это класс, отвечающий за обработку поступающих на определенный заданный разработчиком маршрут запросов. Посредством контроллеров и осуществляется работа сервера с клиентом. Задача контроллера – обработка входящих запросов, произведения определенных операций на стороне сервера и генерация ответа – если это предусмотрено вызываемым методом контроллера. Как правило, в виде ответа контроллеры возвращают различные html-представления и связанную с ними информацию

В информационной системе определены 5 модификаторов доступа и в зависимости от уровня доступа к приложению, пользователям будет доступен разный функционал. Ниже перечислены модификаторы доступа:

- студент;
- староста;
- преподаватель;
- декан;
- админ.

К веб-приложению будут иметь доступ только те пользователи, у которых модификатор доступа «студент, «староста» или «преподаватель». На рисунке 1 показан функционал, который будет доступен пользователям с данными модификаторами доступа.

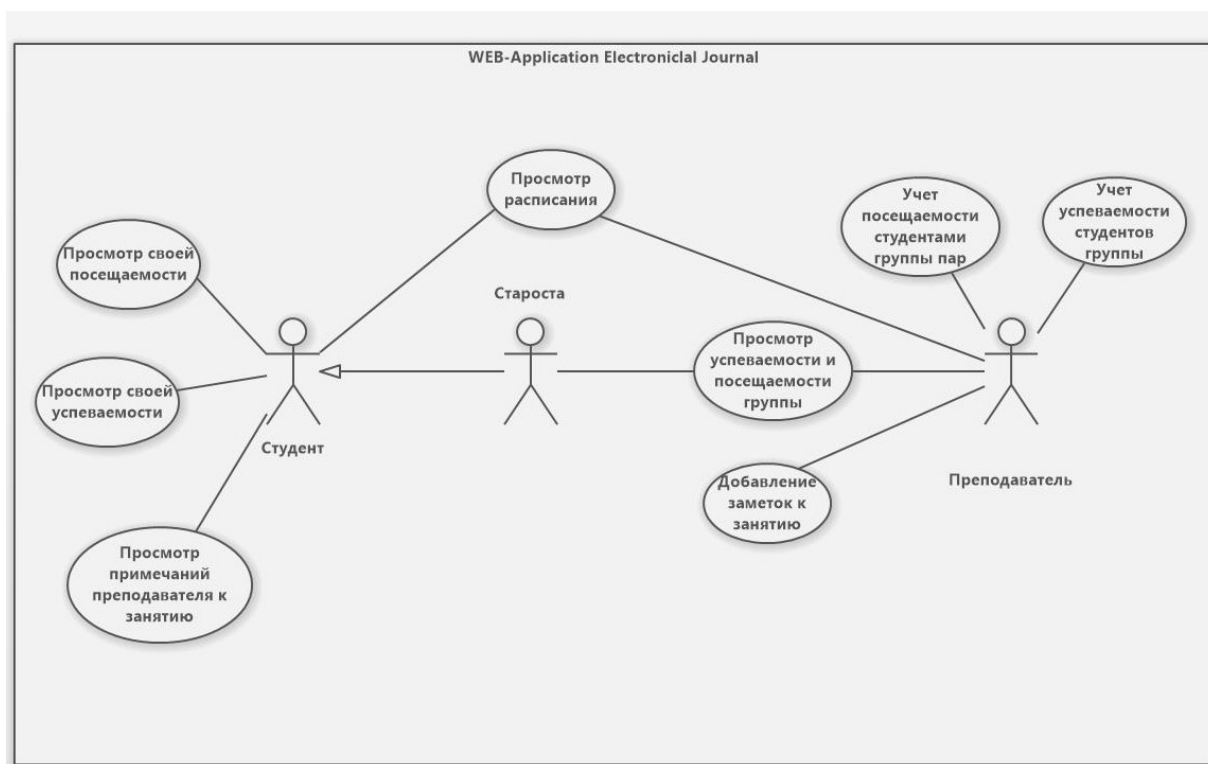
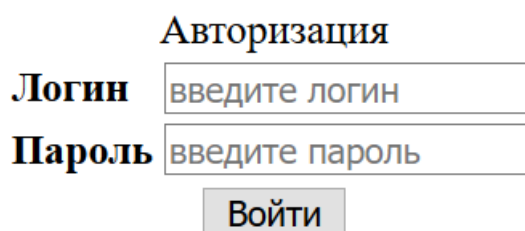


Рис. 1. Диаграмма вариантов использования для CMS «Электронный журнал»

При запуске приложения пользователь должен будет авторизоваться. Авторизация пользователя проходит по следующему алгоритму. Сначала пользователь должен ввести логин и пароль. Пользователь не сможет отправить запрос на вход в систему, пока не заполнит поля. Далее проверяем введенные пользователем данные. Сначала проверяем логин, существует ли аккаунт с таким логином в базе данных, если не существуем, то сообщаем пользователю, что пользователь с таким логином не найден и просим повторить ввод данных еще раз. Если пользователь с данным логином найдет, тогда проверяем уровень доступа к приложению. Если у пользователя модификатор доступа не «студент», не «староста» и не «админ», то сообщаем, что недостаточно прав для входа в приложение и просим повторить ввод данных еще раз. Если пользователь с таким логином существует и у него достаточно прав для авторизации в приложении, тогда проверяем введенный пароль. Хешируем введенный пользователем пароль и сравниваем с хэшем, который хранится в базе данных. Если пароль введен неверно, тогда сообщаем пользователю, что пароль введен неправильно и просим его повторить ввод данных. Стартовая страница приложения представлена на рисунке 2, блок схема алгоритма представлена на рисунке 3.



Авторизация

Логин

Пароль

Рис. 2. Окно авторизации

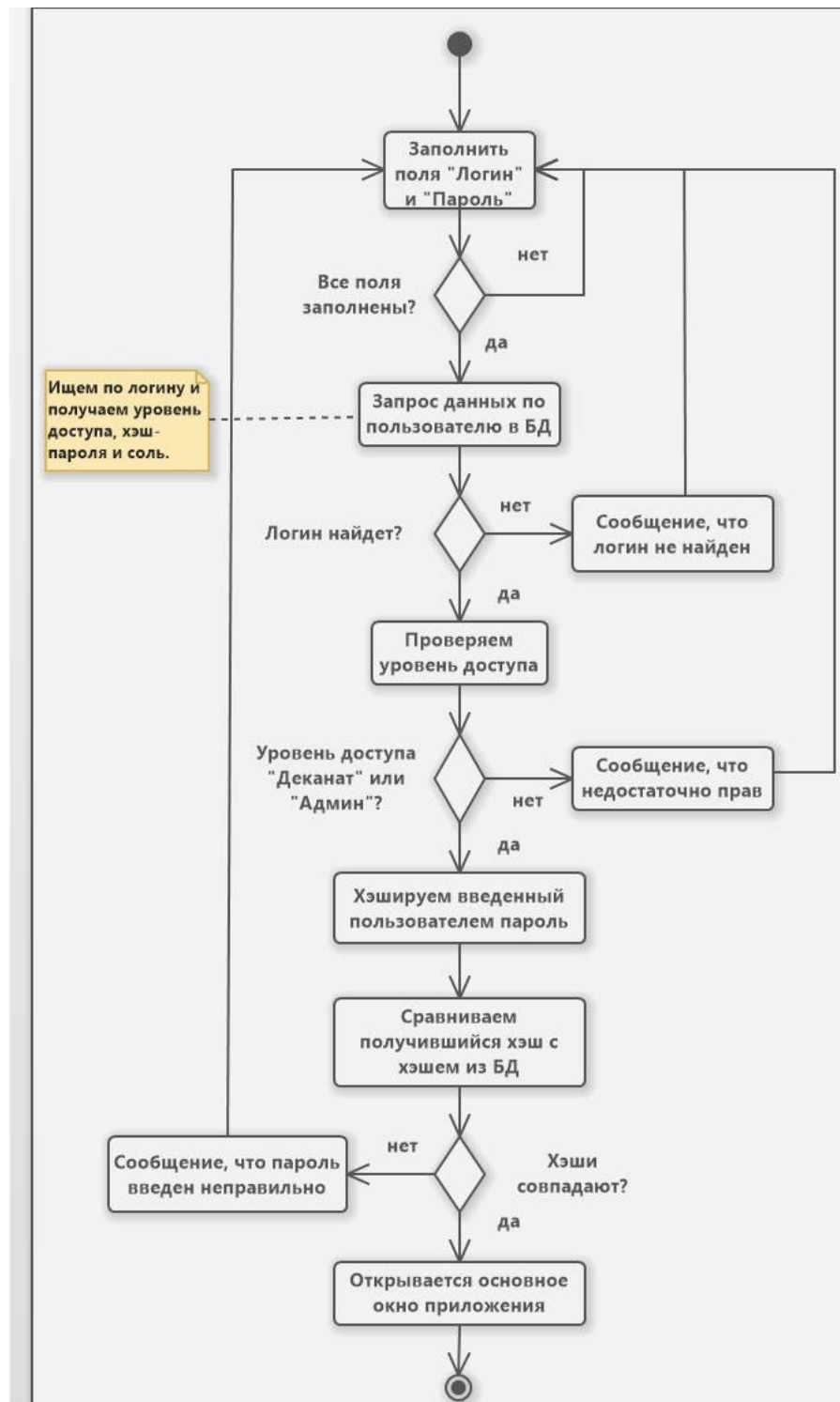


Рис. 3. Авторизация пользователя

В зависимости от модификатора доступа к приложению, в браузере откроется стартовая страница с основным функционалом. Так при авторизации пользователя с модификатором доступа «студент» или «староста» откроется

стартовая страница, на котором будет выведено расписание предстоящих пар его группы, как показано на рисунке 4.

Электронный журнал [Расписание](#) [Успеваемость](#) [Успеваемость группы](#)

Расписание пар

Предмет	Тип занятия	Преподаватель	Дата занятия	Время	Аудитория
Web-программирование	Зачет	Коновалов Коновалов Андреевич	2 июня 2020 г.	10:45 – 12:15	Оршанская 403В
Web-программирование		Коновалов Коновалов Андреевич	3 июня 2020 г.	09:00 – 10:30	
Машинно-ориентированное программирование	Лабораторная работа	Балакирев Балакирев Евгеньевич	5 июня 2020 г.	09:00 – 10:30	
Web-программирование	Зачет	Коновалов Коновалов Андреевич	9 июня 2020 г.	13:00 – 14:30	

Рис. 4. Расписание студента

Алгоритм формирования расписания занятий для студентов представлен на рисунке 5.

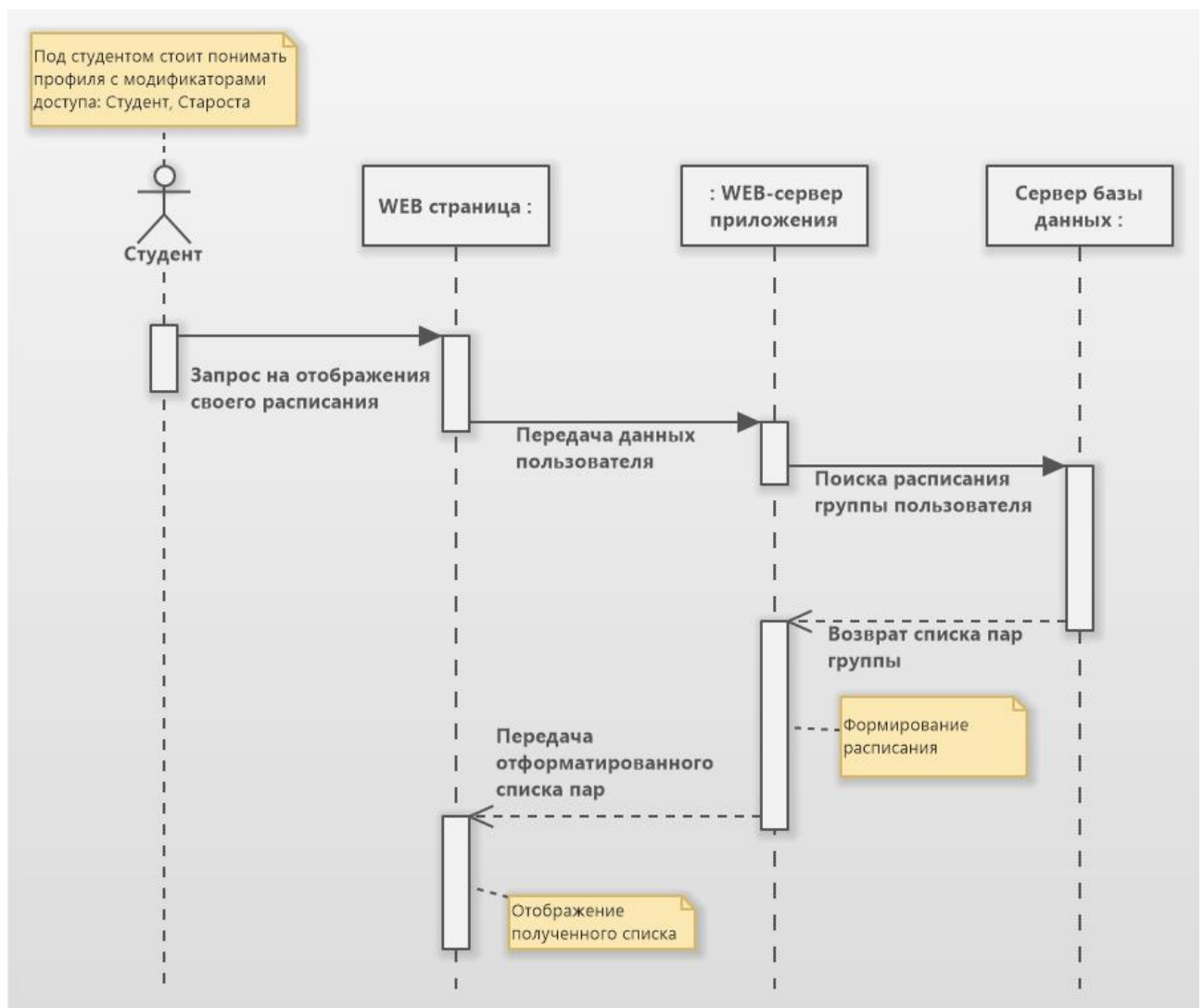


Рис. 5. Формирование расписания.

Для просмотр своей успеваемости, пользователям с модификаторами доступа «студент» и «староста» необходимо перейти на страницу «Успеваемость». После перехода формируется сводная таблица, как показано на рисунке 6, в которой перечисляются предметы текущего семестра, оценки или отметки о присутствии на занятии и заметки преподавателя к занятию.

Электронный журнал [Расписание](#) [Успеваемость](#) [Успеваемость группы](#)

Личная успеваемость

Предмет	28.04.2020 Лекция	28.04.2020 Лабораторная работа	28.04.2020 Лабораторная работа	02.06.2020 Зачет
Web-программирование	3	+	4	3
Комментарий к занятию	Test commit			test commit for web
Предмет	31.05.2020 Практическое занятие			
Программирование в информационных системах	+			
Комментарий к занятию	test			
Предмет				
Операционные системы				
Комментарий к занятию				

Рис. 6. Успеваемость студента

Алгоритм формирования успеваемости студента показан на рисунке 7.

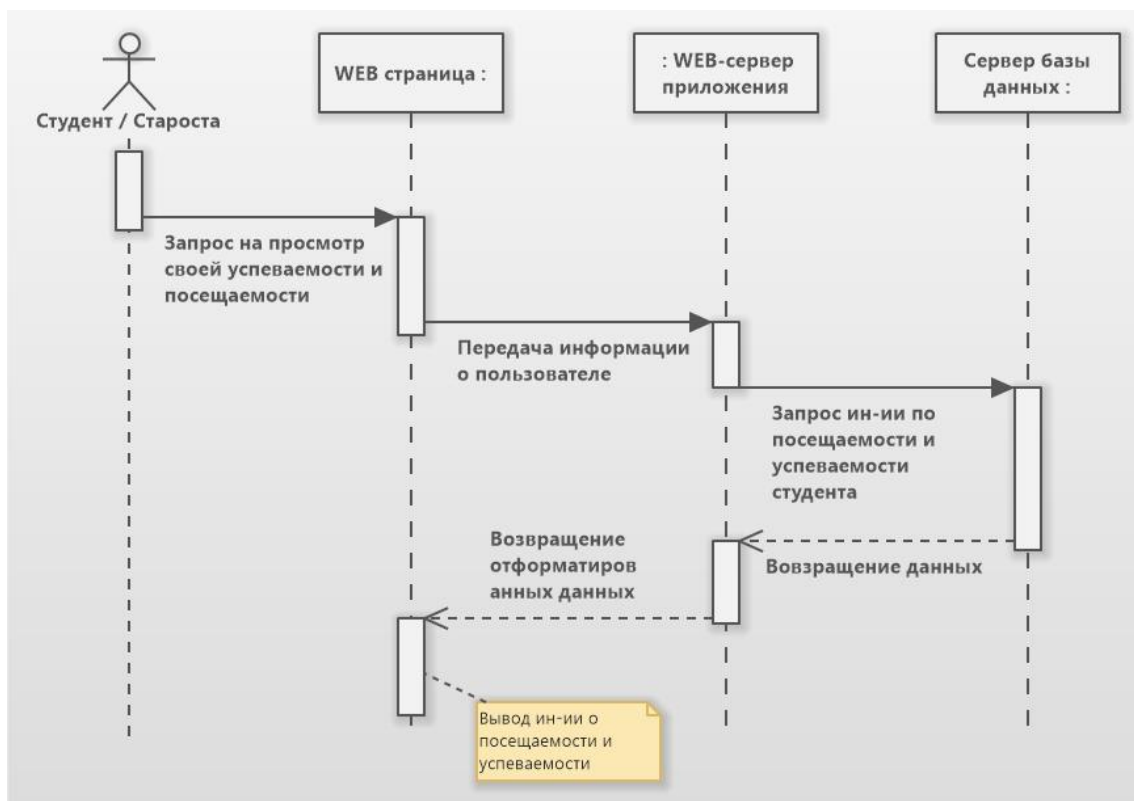


Рис. 7. Формирование таблицы успеваемости студента

У пользователей с модификатором доступа «староста» есть возможность просматривать успеваемость группы, в которой они состоят. Для этого нужно перейти на страницу «успеваемость группы» и выбрать предмет, по которому интересуется успеваемость. Алгоритм просмотра успеваемости группы показан на рисунке 8.

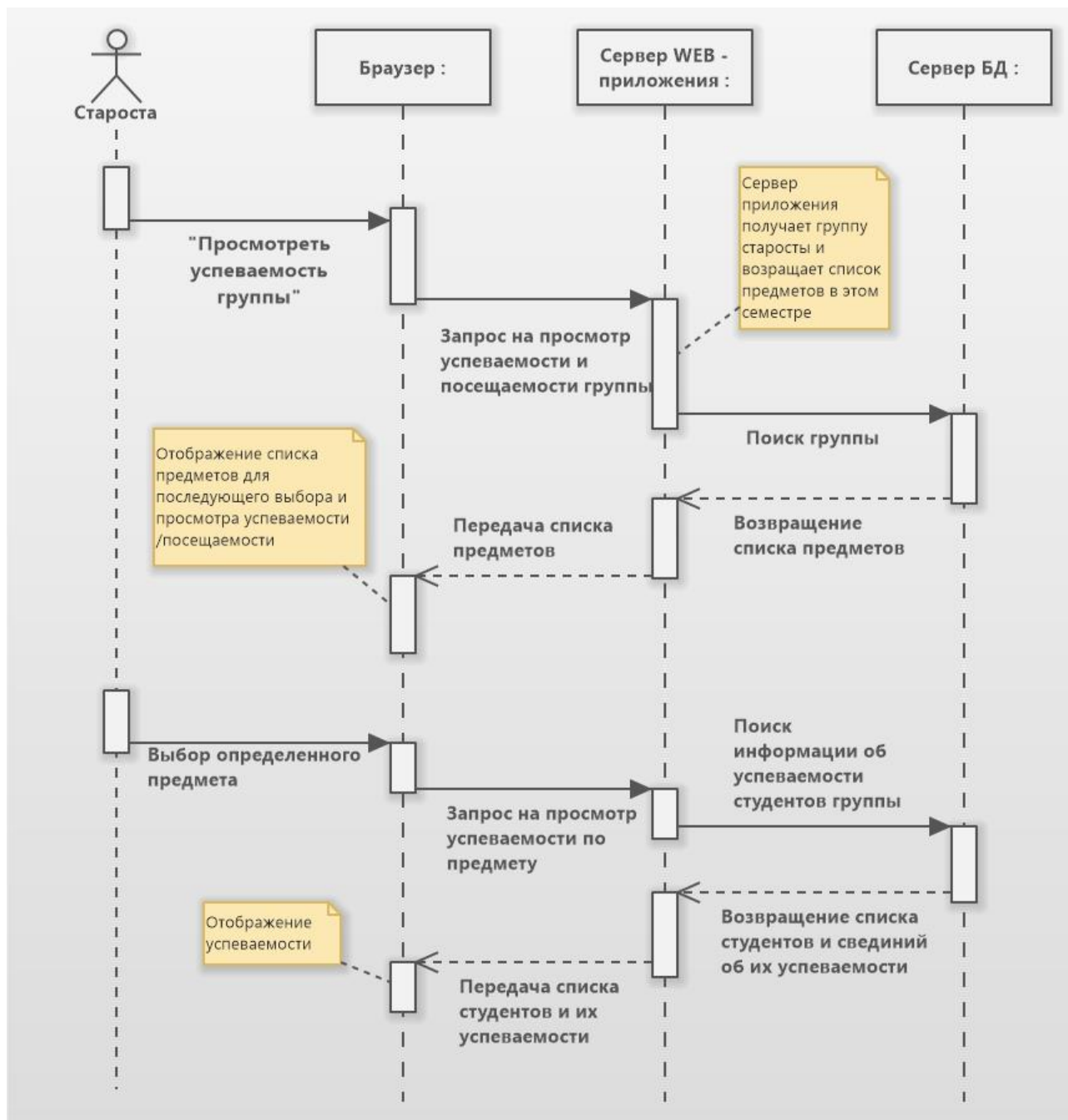


Рис. 8. Просмотр успеваемости группы для старосты.

Если в веб-приложение авторизовался пользователь с модификатором доступа «преподаватель», откроется стартовая страница с расписание предстоящих пар, как показано на рисунке 9.

Электронный журнал [Расписание](#) [Успеваемость групп](#) [Учет успеваемости](#)

Расписание предстоящих занятий

Предмет	Тип занятия	Группа	Дата занятия	Время занятия	Аудитория
Web-программирование	Зачет	МЗО-335Б-17	2 июня 2020 г.	10:45 – 12:15	Оршанская 403В
Web-программирование	Лекция	МЗО-333Б-17	2 июня 2020 г.	10:45 – 12:15	
Web-программирование		МЗО-335Б-17	3 июня 2020 г.	09:00 – 10:30	
Web-программирование	Зачет	МЗО-335Б-17	9 июня 2020 г.	13:00 – 14:30	

Рис. 9. Расписание предстоящих пар преподавателя.

У преподавателя есть возможность вести учет успеваемости и посещаемости студентами его занятий. Для этого нужно перейти на страницу «Учет успеваемости», на которой будет список групп, у которых ведет данный преподаватель, выбрать интересующую группу. После этого отобразится список проведенных занятий у этой группы. Преподаватель выбирает интересующее его занятие и оценивает студентов, как показано на рисунке 10.

Электронный журнал [Расписание](#) [Успеваемость групп](#) [Учет успеваемости](#)

Оценивание занятия по Web-программирование гр. МЗО-335Б-17 за 01.06.2020

ФИО студента	Оценка
Аншуков Михаил	+ <input type="text"/>
Корнилов Артем	+ <input type="text"/>
Милаков Александр Владимирович	+ <input type="text"/>
testChange AddChange TestChange	+ <input type="text"/>
stud enywer werwer	+ <input type="text"/>
Заметка к занятию	<input type="text"/>
Оценить	

Рис. 10. Оценивание занятия

Алгоритм учета успеваемости и посещаемости студентам занятия представлен на рисунке 11.

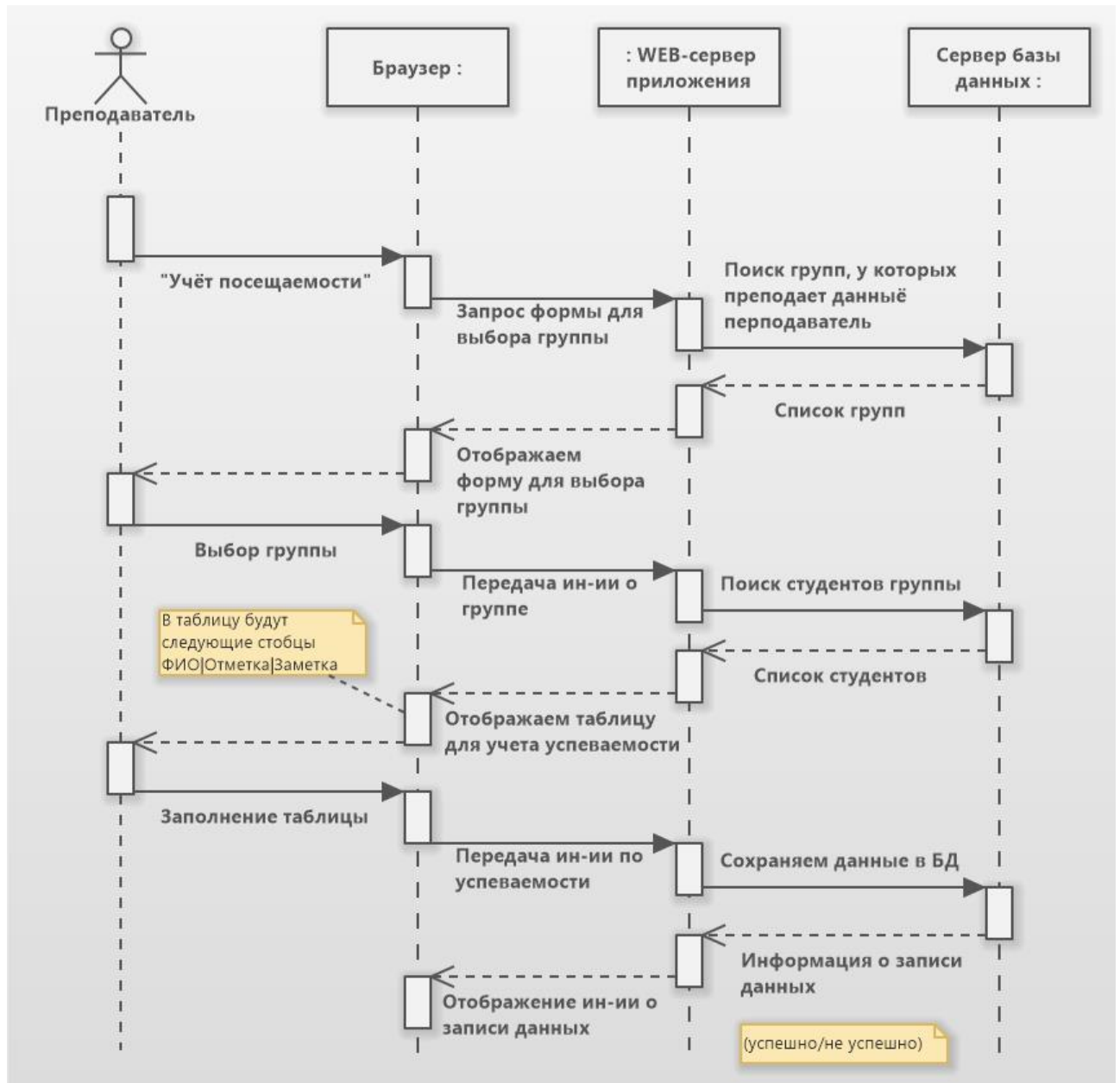


Рис. 11. Учет успеваемости.

Для отображения успеваемости групп, нужно перейти на страницу «Успеваемость групп», на которой будет отображен список групп, у которых ведет преподаватель, далее выбрать интересующий его предмет, как показано на рисунке 12.

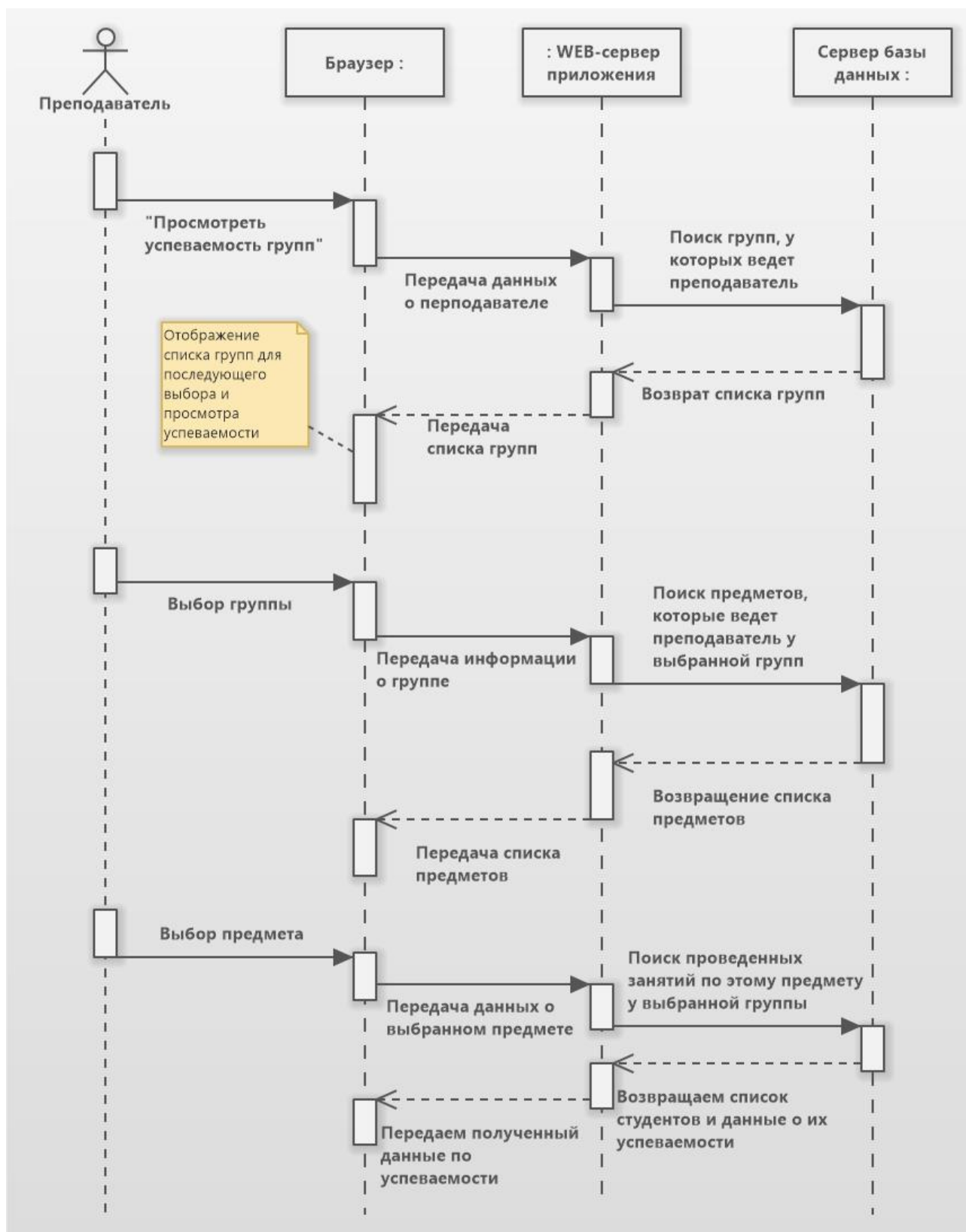


Рис. 12. Просмотр успеваемости группы.

При разработке были созданы следующие классы:

- IndexTeacherViewModel – модель представления для передачи данных из контроллера TeacherHomeController в представление Index;
- SelectedSubjectViewModel – модель представления для передачи данных из контроллера TeacherHomeController в представление SelectSubject;
- StudentScheldueViewModel – модель представления для передачи данных из контроллера StudentHomeController в представление Index;
- StudentsPerfomancesViemModel – модель представления для передачи данных из контроллера StudentHomeController в представление GroupPerfomance;
- PerfomancesViewModel модель представления для передачи данных из контроллера StudentHomeController в представление StudentPerfomance;
- StudentsPerfomanceViewModel – модель представления для передачи данных из контроллера TeacherHomeController в представление AcademicPerfomanceUpdate;
- UserSession - класс, отвечающий за отслеживание текущий сессии пользователя в приложении;

В папке Model.DataBaseModel находятся классы-модели для взаимодействия с базой данных.

В папке Controllers находятся класс контроллеров, которые связывают слой данных с представлением.

В папке View находятся представления с пользовательским интерфейсом.

5. ОРГАНИЗАЦИЯ ДАННЫХ

Организация входных данных.

Входными данными для программы являются:

- полученные текстовые данные при заполнении полей в представлениях;
- запросы, возникающие при нажатии кнопок в представлении.

Организация выходных данных.

Выходными данными программы являются представления, сформированные в зависимости от входных данных программы.

Описание таблицы базы данных

Для хранения информации в информационной системе «электронный журнал» используется SQL Server 2016 с базой данных «ElectronicalJournalDB». В базе данных для хранения и структурирования информации реализованы пятнадцать связанных между собой таблиц.

Таблица Users содержит информацию о пользователе, её структура представлена в таблице 1. Поле UserId является ключевым в данной таблице.

Таблица AccessLevel является справочником модификаторов доступа, её структура представлена в таблице 2. Данная таблица формирует связи вида один ко многим с таблицей Person посредством поля AccessLevelId.

Таблица 1 Описание таблицы Users

Имя столбца	Тип	Ключ	Описание
UserId	INT NOT NULL	PK	Идентификатор пользователя
FirstName	NVARCHAR(50) NOT NULL		Имя
MiddleName	NVARCHAR(50)		Отчество
LastName	NVARCHAR(50) NOT NULL		Фамилия
Login	NVARCHAR(30) NOT NULL UNIQUE		Логин пользователя
PasswordHash	NVARCHAR(64) NOT NULL		Хэш пароля
PasswordSalt	NVARCHAR(50)		“Соль” пароля
AccessLevelId	INT NOT NULL	FK	Уровень доступа пользователя
Phone	NVARCHAR(12) UNIQUE		Телефон пользователя
Email	NVARCHAR(50) UNIQUE		Почта пользователя

Таблица 2 Описание таблицы AccessLevel

Имя столбца	Тип	Ключ	Описание
AccessLevelId	INT NOT NULL	PK	Идентификатор модификатора доступа уровня пользователя
AccessLevel	NVARCHAR(30) NOT NULL		Модификатор уровня доступа

Таблица 3 Описание таблицы Group

Имя столбца	Тип	Ключ	Описание
GroupId	INT NOT NULL	PK	Идентификатор студенческой группы
GroupName	NVARCHAR(30) NOT NULL		Наименование группы
YearFormationGroup	DATETIME		Дата формирования группы

Таблица StudentGroup предназначена для формирования студенческих групп, ее структура приведена в таблице 4.

Таблица 4 Описание таблицы StudentGroup

Имя столбца	Тип	Ключ	Описание
StudentGroupId	INT NOT NULL	PK	Идентификатор таблицы
Userd	INT UNIQUE NOT NULL	FK	Идентификатор студента
GroupId	INT UNIQUE NOT NULL	FK	Идентификатор группы студента

Таблица Teachers предназначена для формирования педагогического состава, ее структура приведена в таблице 5.

Таблица Position является справочником должностей преподавателей, её структура приведена в таблице 6. Данная таблица формирует связь вида один ко многим с таблицей Teachers посредством поля TeacherId

Таблица 5 Описание таблицы Teachers

Имя столбца	Тип	Ключ	Описание
TeacherId	INT NOT NULL	PK	Идентификатор преподавателя
UserId	INT UNIQUE NOT NULL	FK	Идентификатор пользователя
PositionId	INT NOT NULL	FK	Идентификатор должности

Таблица 6 Описание таблицы Position

Имя столбца	Тип	Ключ	Описание
PositionId	INT NOT NULL	PK	Идентификатор должности
Position	NVARCHAR(50) NOT NULL		Наименование должности

Таблица Lessons содержит информации о расписании пар, о месте проведения пар, о преподавателе ведущем пару, о времени и дате проведения занятия, её структура приведена в таблице 7. Данная таблица формирует связи

вида многие к одному с таблицей LessonType посредством поля LessonTypeId, с таблицей TimeSchedule посредством поля TimeScheduleId, с таблицей Classrooms посредством поля ClassroomId и с таблицей Subjects посредством поля SubjectId. Также данная таблица формирует связи многие ко многим с таблицей Groups посредством таблицы GroupLessons, ее структура представлена в таблице 8, и с таблицей Teachers посредством таблицы TeacherLessons, ее структура представлена в таблице 9.

Таблица LessonType является справочником типов занятий (лекция, семинар и т.д.), её структура приведена в таблице 10.

Таблица TimeSchedule является справочником времени расписания пар, её структура приведена в таблицу 11.

Таблица Classrooms является справочником аудиторий, её структура приведена в таблице 12. Данная таблица формирует связь вида один ко многим с таблицей Buildings посредством поля BuildingId.

Таблица Buildings является справочником корпусов, её структура приведена в таблице 13.

Таблица Subjects является справочником учебных дисциплин, её структура приведена в таблице 14.

Таблица 7 Описание таблицы Lessons

Имя столбца	Тип	Ключ	Описание
LessonId	INT NOT NULL	PK	Идентификатор занятия
GroupId	INT NOT NULL	FK	Учебная группа
TeacherId	INT NOT NULL	FK	Преподаватель
LessonTypeId	INT NOT NULL	FK	Тип занятия
TimeScheduleId	INT NOT NULL	FK	Время занятия
SubjectId	INT NOT NULL	FK	Дисциплина
DateTime	DATETIME NOT NULL		Дата занятия
ClassroomId	INT NOT NULL	FK	Аудитория
Notes	NVARCHAR(500)		Заметка по занятию

Таблица 8 Описание таблицы GroupLessons

Имя столбца	Тип	Ключ	Описание
GroupLessonId	INT NOT NULL	PK	Идентификатор
GroupId	INT NOT NULL	FK	Учебная группа
LessonId	INT NOT NULL	FK	Занятие

Таблица 9 Описание таблицы TeacherLessons

Имя столбца	Тип	Ключ	Описание
TeacherLessonId	INT NOT NULL	PK	Идентификатор
TeacherId	INT NOT NULL	FK	Преподаватель
LessonId	INT NOT NULL	FK	Занятие

Таблица 10 Описание таблицы LessonType

Имя столбца	Тип	Ключ	Описание
LessonTypeId	INT NOT NULL	PK	Идентификатор типа занятия
LessonType	NVARCHAR(50) NOT NULL		Тип занятия

Таблица 11 Описание таблицы TimeSchedule

Имя столбца	Тип	Ключ	Описание
TimeScheduleId	INT NOT NULL	PK	Идентификатор времени занятия
TimeSchedule	NVARCHAR(50) NOT NULL		Время занятия.

Таблица 12 Описание таблицы Classrooms

Имя столбца	Тип	Ключ	Описание
ClassroomId	INT NOT NULL	PK	Идентификатор аудитории
BuildingId	INT NOT NULL	FK	Идентификатор корпуса
Classroom	NVARCHAR(30) NOT NULL		Аудитория

Таблица 13 Описание таблицы Buildings

Имя столбца	Тип	Ключ	Описание
BuildingId	INT NOT NULL	PK	Идентификатор корпуса
Building	NVARCHAR (50) NOT NULL		Наименование корпуса

Таблица 14 Описание таблицы Subjects

Имя столбца	Тип	Ключ	Описание
SubjectId	INT NOT NULL	PK	Идентификатор предмета
SubjectName	NVARCHAR(100) NOT NULL		Наименование предмета

Таблица AcademicPerfomances предназначена для отслеживания посещаемости и успеваемости студентов, её структура приведена в таблице 15.

Таблица 15 Описание таблицы AcademicPerfomances

Имя столбца	Тип	Ключ	Описание
AcademicPerfomanceId	INT NOT NULL	PK	Идентификатор
UserId	INT NOT NULL		Идентификатор студента
Mark	NVARCHAR(20)		Оценка
LessonId	INT NOT NULL	FK	Идентификатор занятия.

Структура базы данных изображена на рисунке 13.

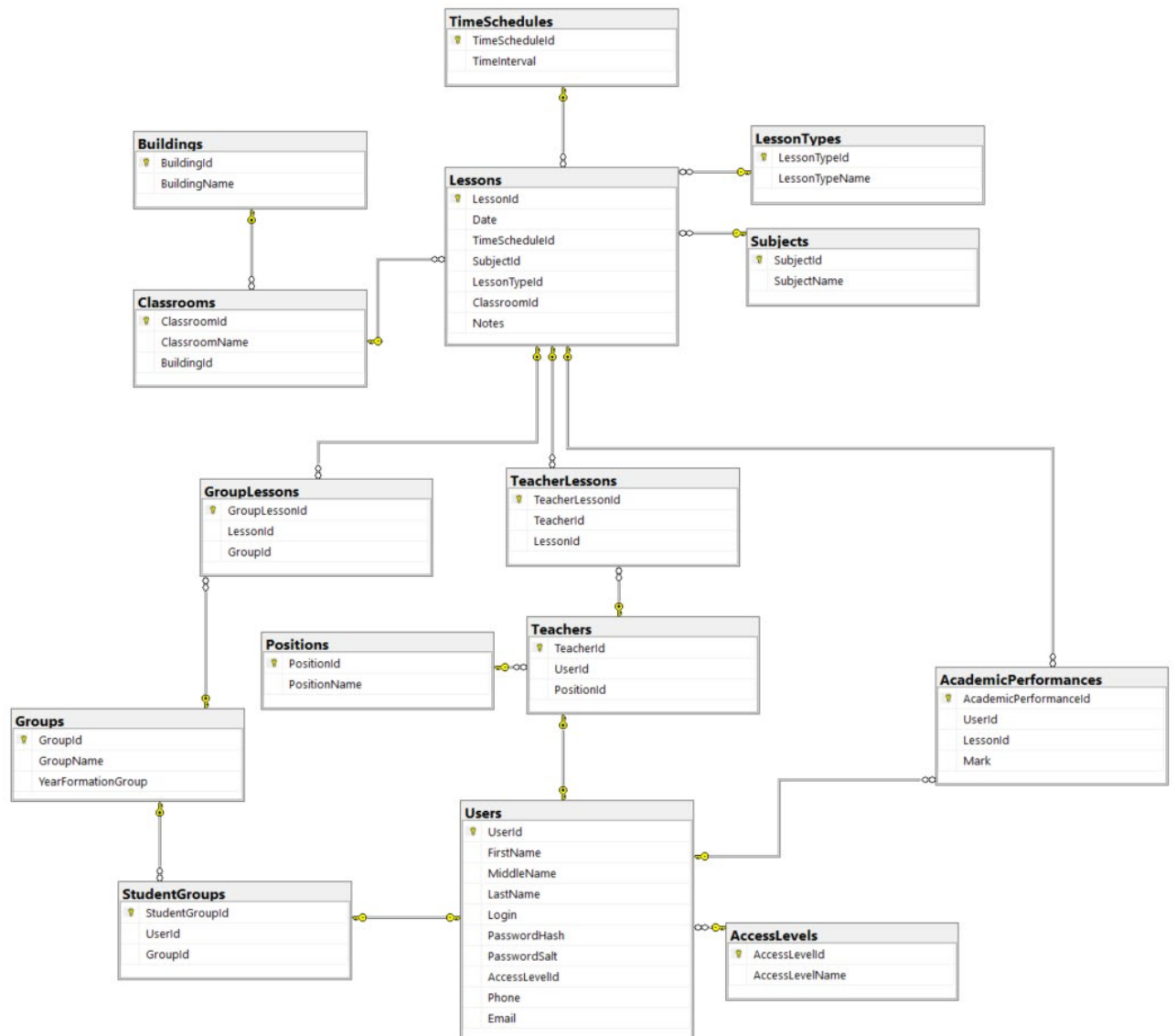


Рис. 13. Структура базы данных.

6. РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММ И ИХ ОЦЕНКА

Объект испытания

Объектом испытания является веб-приложение «ElectronicJournal_Web»

Цель испытания

Целью испытания является проверка работоспособности веб-приложения «ElectronicJournal_Web».

Требования к программе

Во время испытания было необходимо проверить следующий функционал программы:

- авторизация пользователя в приложении;
- оценивание студентов;

Метод испытания

Тестирования приложения «ElectronicJournal_Desktop» проводилось методом «черного ящика». Результаты тестирования приведены в таблице 16.

Таблица 16 Результаты тестирования

Назначение теста	Входные данные	Фактическая работа программы	Ожидаемая работа программы
Проверка реакции приложения при авторизации	Не заполнены поля для пароля и логина	Отказ в авторизации	Отказ в авторизации
	Логин, которого нет в базе данных	Вывод сообщения «Пользователь с таким логином не найден»	Вывод сообщения «Пользователь с таким логином не найден»
	Логин, который есть в базе данных и неправильный пароль	Вывод сообщения «Неправильный пароль»	Вывод сообщения «Неправильный пароль»

Продолжение таблицы 16

	Корректный логин и пароль, но пользователю недостаточно прав для авторизации	Вывод сообщения «Недостаточно прав для входа в приложение»	Вывод сообщения «Недостаточно прав для входа в приложение»
	Корректный логин и пароль и достаточно прав для авторизации.	Успешный вход и отображение главного окна приложения	Успешный вход и отображение главного окна приложения
Проверка реакции приложения на оценивание студентов	Выбранные значения оценок пользователем	Успешной сохранение данных	Проверка реакции приложения на оценивание студентов
Проверка реакции приложения на повторное оценивание студентов	Выбранные значения оценок пользователем	Обновление данных в базе данных	Обновление данных в базе данных

Оценка результатов тестирования

В тестовых примерах, приведенных в таблице 16, ожидаемая реакция совпадает с фактической, что доказывает работоспособность программы.

ЗАКЛЮЧЕНИЕ

В соответствии с поставленной задачей было разработано веб-приложение для работы информационной системы «Электронный журнал». Программа выполнена в полном соответствии с поставленной задачей, была проверена и отлажена. Выполнение курсовой работы помогла закрепить материал и знания, полученные на лекциях и лабораторных работах.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

SVGA – SuperVideoGraphicsArray – улучшенный видеографический адаптер, обеспечивающий разрешающую способность 1280x1024 точек и более чем 256 цветов.

Мб – Мегабайт – единица измерения объема информации.

ГГц – Гигагерц – единица измерения тактовой частоты процессора

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Лекции по дисциплине «Программирование в информационных системах»;
2. Лекции по дисциплине «Web-программирование»;
3. Эндрю Троелсен «Язык программирования C# и платформа .NET 4.5 6-е издание. – М: ООО «И.Д. Вильямс», 1310 с.;
4. Фримен Адам Раттц-мл., Джозеф С. LINQ. Язык интегрированных запросов в C# 2010 для профессионалов. : Пер. с англ. – М.:ООО «И.Д.Вильямс» 2011-656с.
5. Руководство по Entity Framework Core:
<https://metanit.com/sharp/entityframeworkcore/>;
6. Руководство по ASP.NET Core 3: <https://metanit.com/sharp/aspnet5/>;
7. Руководство по HTML5 и CSS3 <https://html5book.ru/>.

08335 – 01 81 01

ПРИЛОЖЕНИЕ А

Ссылка	на	исходный	код	приложения:
https://github.com/Meeluck/ElectronicJournal_WEB .				