



**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:

«Программирование в информационных системах»

Информационная система «Электронный журнал»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

08335 – 01 81 01

Листов 27

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	7
5. Организация данных.....	21
6. Результаты работы программ и их оценка	23
Заключение	25
Перечень сокращений.....	25
Список используемой литературы	26
Приложение А	27

1. ВВЕДЕНИЕ

Целью данной курсовой работы является разработка настольного приложения «ElectronicJournal_Desktop», являющееся частью информационной системы «Электронных журнал».

2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ

Основание для разработки является учебный план по дисциплине «Программирование в информационных системах»

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;
- добавление новых пользователей в информационную систему;
- отображение подробной информации
- редактирование пользовательских данных;
- удаление аккаунтов пользователей;
- добавление новых учебных групп;
- редактирование информации о группе;
- отображение подробной информации о группе;
- удаление учебных групп;
- просмотр занятий группы;
- просмотр подробной информации о занятии группы;
- добавление новых занятий для группы;
- удаление занятий

Требования надежности

Для надежного функционирования программы необходимо обработать следующие исключительные ситуации:

- попытка добавить нового пользователя с логином, который уже зарезервирован
- удаление аккаунтов пользователей
- удаление групп

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;
- .NET Core — платформа разработки общего назначения с открытым кодом, предназначенная для создания кроссплатформенных приложений;
- MvvmNavigation – библиотека, позволяющая настраивать поведение WPF приложения и реализовать навигацию на уровне слоя ViewModel. Библиотека полностью придерживается паттерна MVVM.

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:

- .NET Core 3.1 Runtime – среда выполнения .NET Core используется для запуска приложений, созданных с помощью .NET Core;
- Windows 10;
- 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 настольного приложения была выбрана технология WPF (Windows Presentation Foundation), которая является частью экосистемы платформы .NET и представляет собой подсистему для построения графических интерфейсов.

При разработке приложения использовался архитектурный паттерн Model-View-ViewModel (MVVM). Данный паттерн используется для разделения модели и её представления, что необходимо для изменения их отдельно друг от друга.

Модель (Model) представляет собой фундаментальные данные, необходимые для работы приложения;

Представление (View) является графическим интерфейсом, с которым непосредственно взаимодействует пользователь. Представление является подписчиком на событие изменения значений свойств или команд, предоставляемых моделью представления.

Модель представления (ViewModel) является, с одной стороны, абстракцией Представления, а с другой, предоставляет обёртку данных из Модели, которые подлежат связыванию. То есть, она содержит Модель, которая преобразована к Представлению, а также содержит в себе команды, которыми может пользоваться Представление, чтобы влиять на Модель.

Преимущества паттерна MVVM:

- тестируемость MVVM-приложений. Приложения, разработанные с использованием MVVM, обладают очень хорошим основанием для проведения модульного тестирования с целью проверки работы отдельных классов и методов.
- Меньшее количество кода. Объем кода, необходимого для управления представлением немного снижается при использовании

MVVM, а это означает, что снижается риск допустить ошибки и уменьшается код для написания модульных тестов.

- Улучшенное проектирование приложений. Разработчики и дизайнеры могут самостоятельно работать над разными частями приложения.
- Легкость понимания логики представления. MVVM предусматривает хорошо организованную и легкую для понимания конструкцию построения графического интерфейса за счет использования механизмов привязок, команд и шаблонов данных.

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 настольному приложению будут иметь доступ только те пользователи, у которых модификатор доступа «декан» или «админ». На рисунке 1 показан функционал, который будет доступен пользователям с модификатором доступа «декан» и с модификатором доступа «админ».

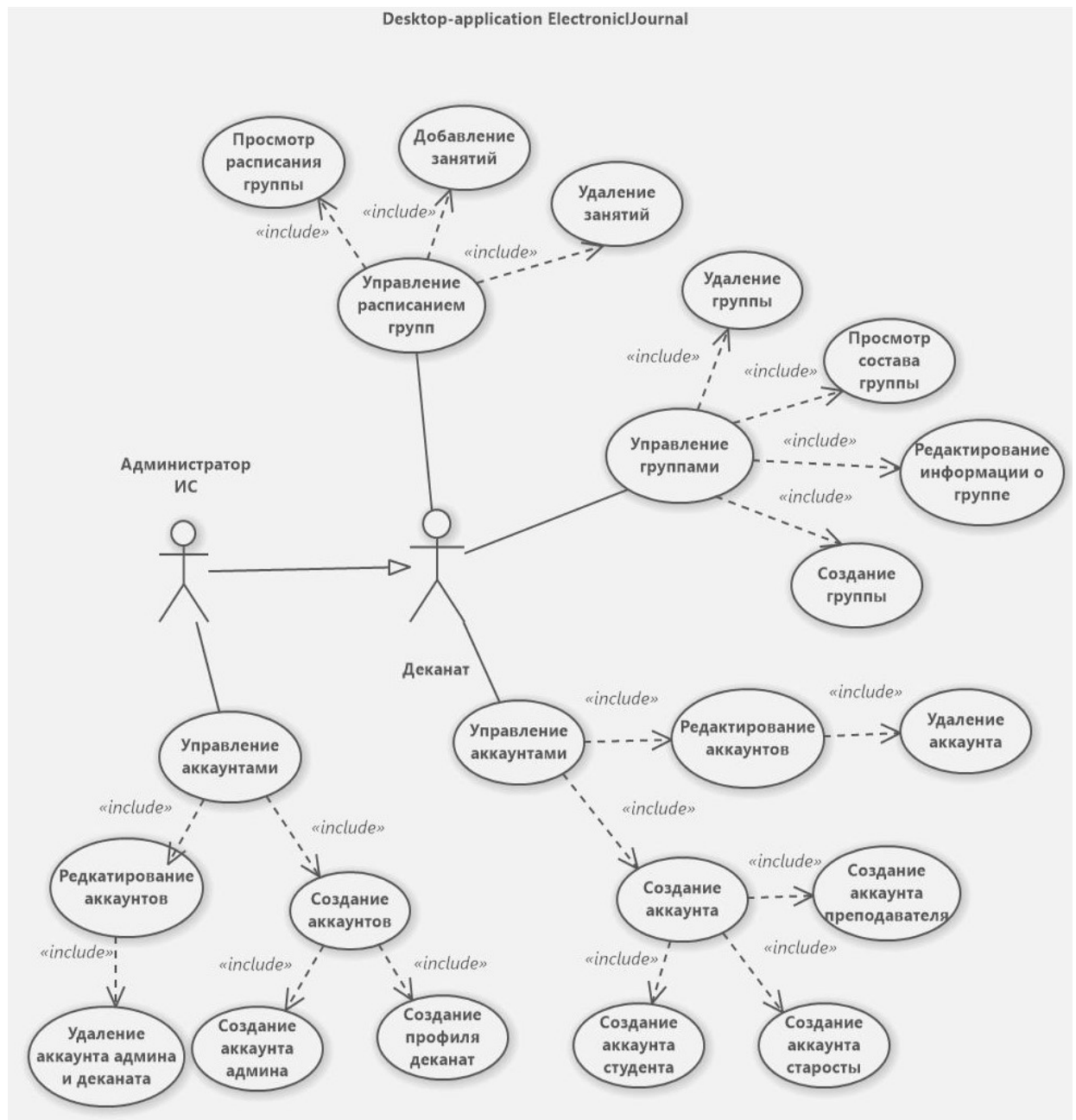
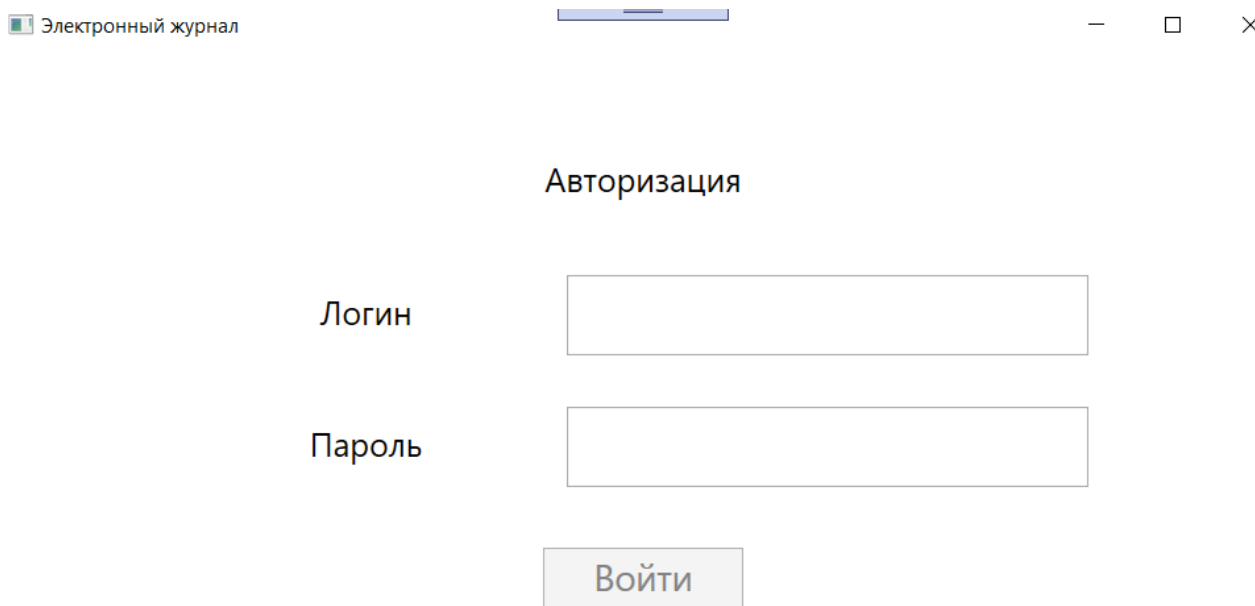


Рис. 1. Диаграмма вариантов использования настольного приложения

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 «декан» 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. Для 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. Стартовое окно приложения представлена на рисунке 2, блок схема алгоритма представлена на рисунке 3.



Электронный журнал

Авторизация

Логин

Пароль

Войти

Рис. 2. Окно авторизации

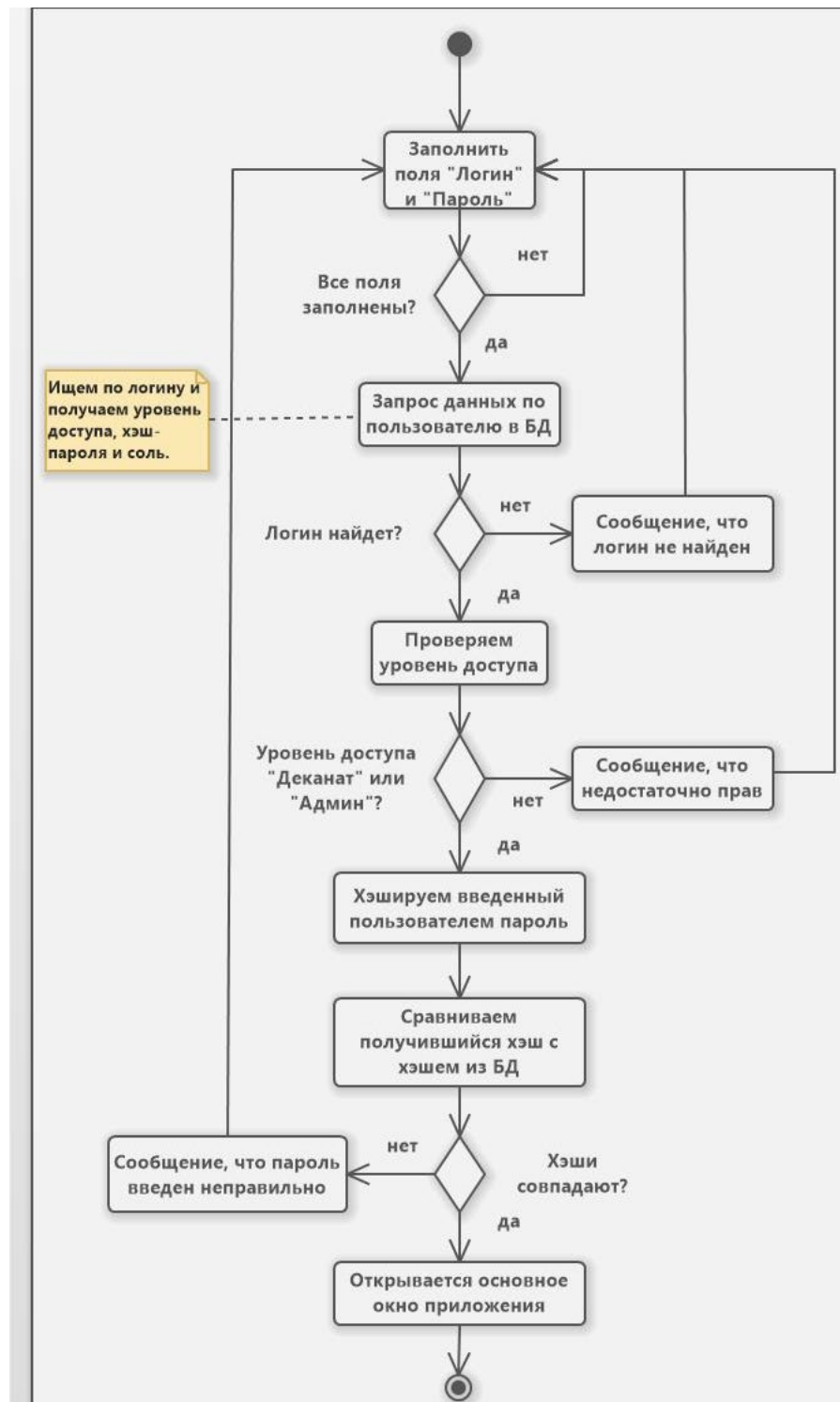


Рис. 3. Авторизация пользователя

После авторизации открывает главное окно приложения и пользователю предлагаются следующие варианты использования приложения: управление аккаунтами, управление группами и управление расписанием, как показано на рисунке 4.

Главный экран

Milakov Alexander

Админ

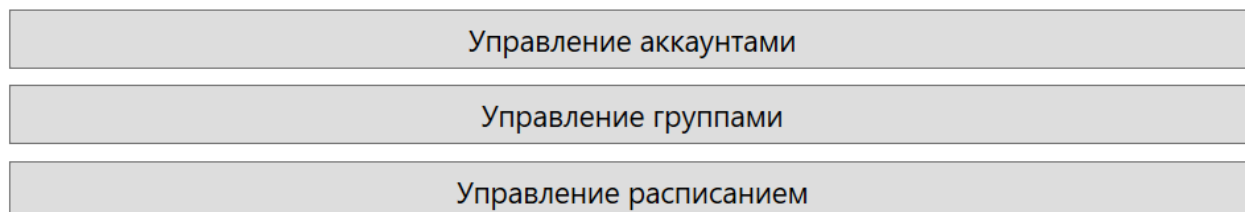


Рис. 4. Главное окно приложения

Рассмотрим, как работает часть приложения, связанная с управлением аккаунтами. После нажатия кнопки «Управление аккаунтами» открывает окно с списком пользователей, к которым имеет доступ данный пользователь. В случае, если авторизовался пользователь с модификатором доступа админ, то вернется список со всеми пользователями, которые хранятся в базе данных. Если войдет пользователь с модификатором доступа деканат, тогда вернется список пользователей с модификаторами доступа «студент», «староста» и «преподаватель». Также в данном окне предоставляется доступ к следующему функционалу: добавление аккаунта, редактирование аккаунтов, отображение подробной информации о пользователе и удаление аккаунта, как показано на рисунке 5. Для просмотра подробной информации, редактирования аккаунта и удаления аккаунта предварительно нужно выбрать строку с необходимым пользователем.

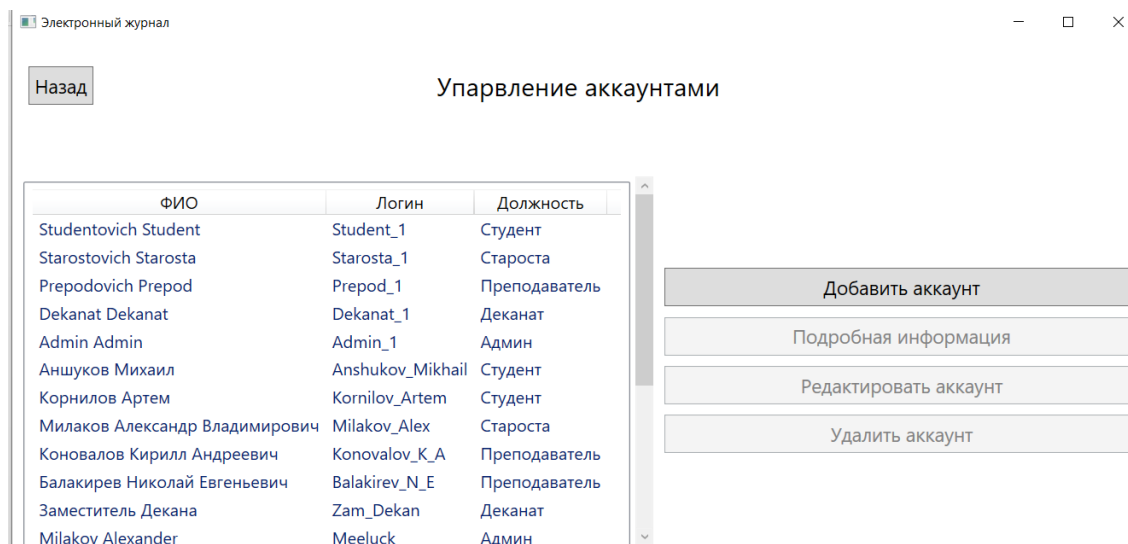


Рис. 5. Управление аккаунтами

Для добавления нового пользователя необходимо нажать «добавить аккаунт». Алгоритм добавление нового пользователя представлен на рисунке 6.

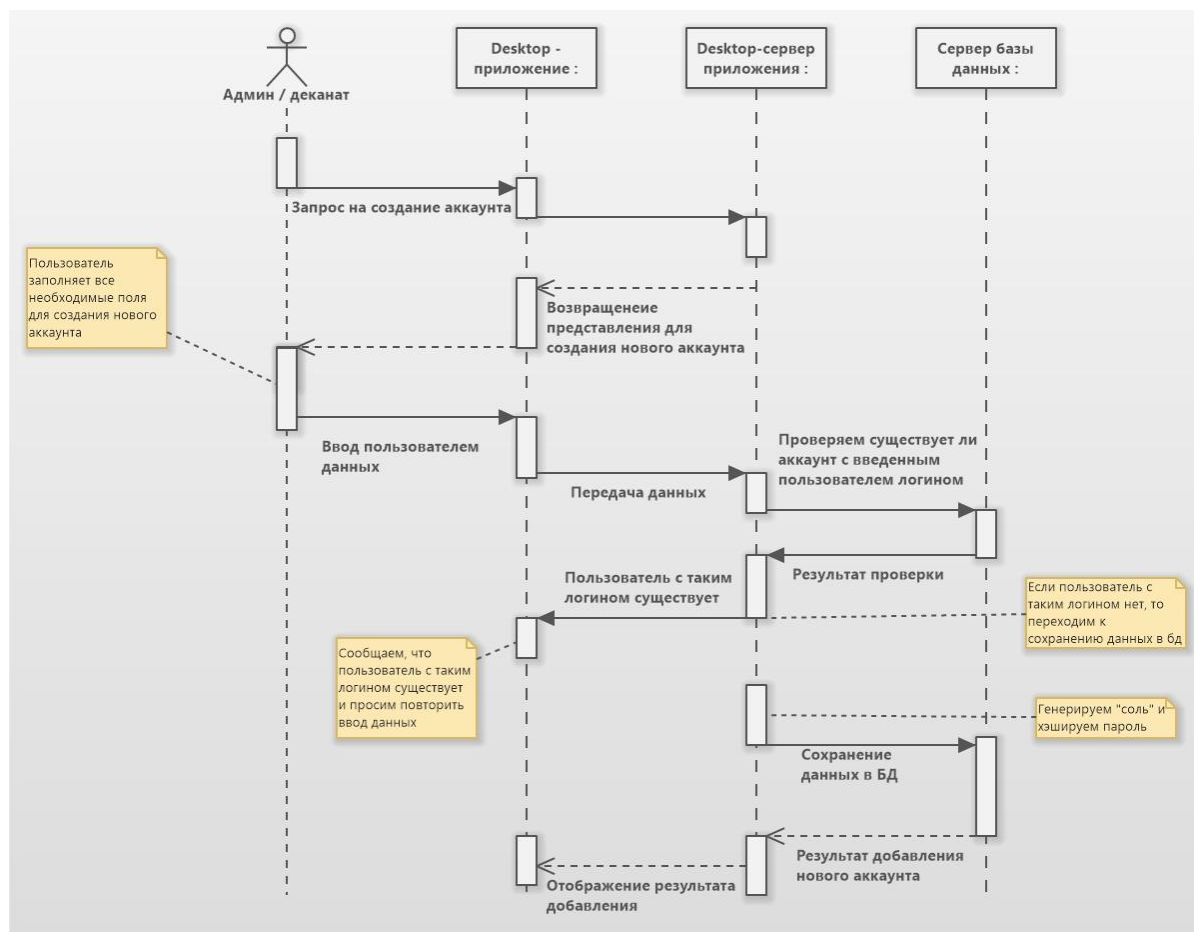


Рис. 6. Добавление нового пользователя

Для редактирования аккаунта нужно выбрать интересующий аккаунт и нажать «Редактировать аккаунт». Алгоритм редактирования представлен на рисунке 7. Для удаления аккаунта также необходимо выбрать интересующий нас аккаунт нажать «Удалить аккаунт». Алгоритм аналогичен редактированию аккаунта.

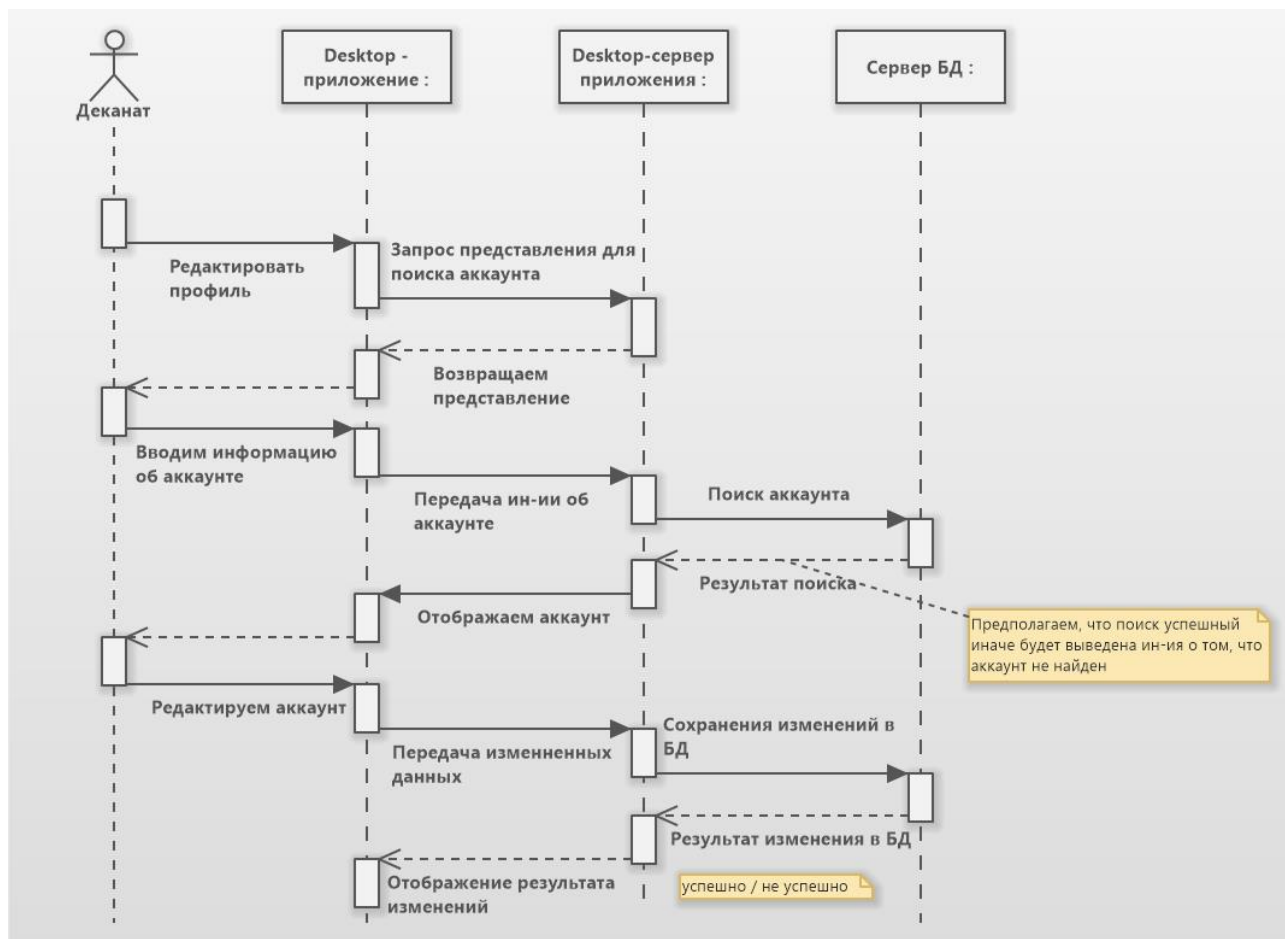


Рис. 7. Редактирование аккаунта

Рассмотрим, как работает часть приложения, связанная с управлением группами. При нажатии на кнопку «Управление группами» открывается окно как на рисунке 8, в котором отображается список групп, которые сохранены в базе данных. У пользователя есть возможность добавлять новые группы, просматривать подробную информацию по группе, редактировать информацию по группе и удалять группу.

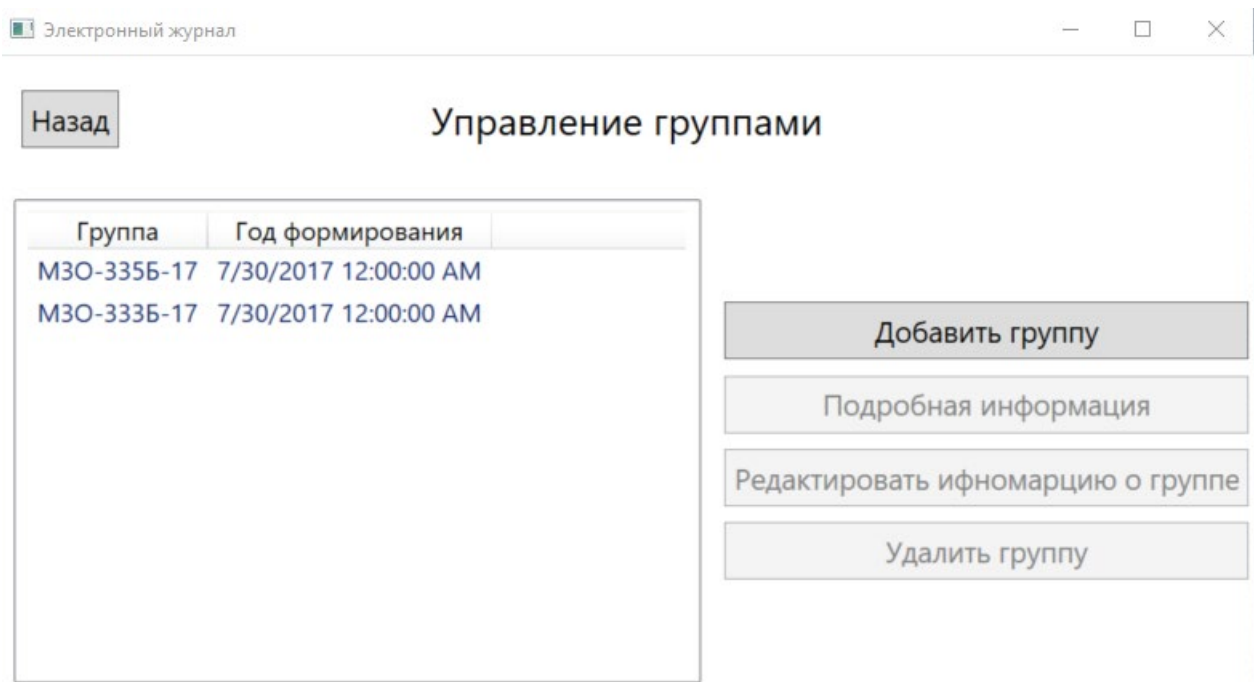


Рис. 8. Управление группами

Для добавления группы необходимо заполнить форму, как показано на рисунке 9 и нажать добавить. Введенная информация будет сохранена в бд.

Рис. 9. Добавление новой группы

Для просмотр подробной информации необходимо предварительно выбрать интересующую группу и нажать «Подробная информация». Будет выведена подробная информация о группе, как показано на рисунке 10.

Алгоритм формирования данных для представления продемонстрирован на рисунке 11.

Электронный журнал

Назад

Подробная информация о группе

Название МЗО-335Б-17

Год формирования 30 июля 2017 г.

Курс 3

Студенты группы

Студент	Модификатор доступа
Аншуков Михаил	Студент
Корнилов Артем	Студент
Милаков Александр Владимирович	Староста
testChange AddChange TestChange	Студент

Редактировать данные о группе Удалить группу

Рис. 10. Подробная информация о группе

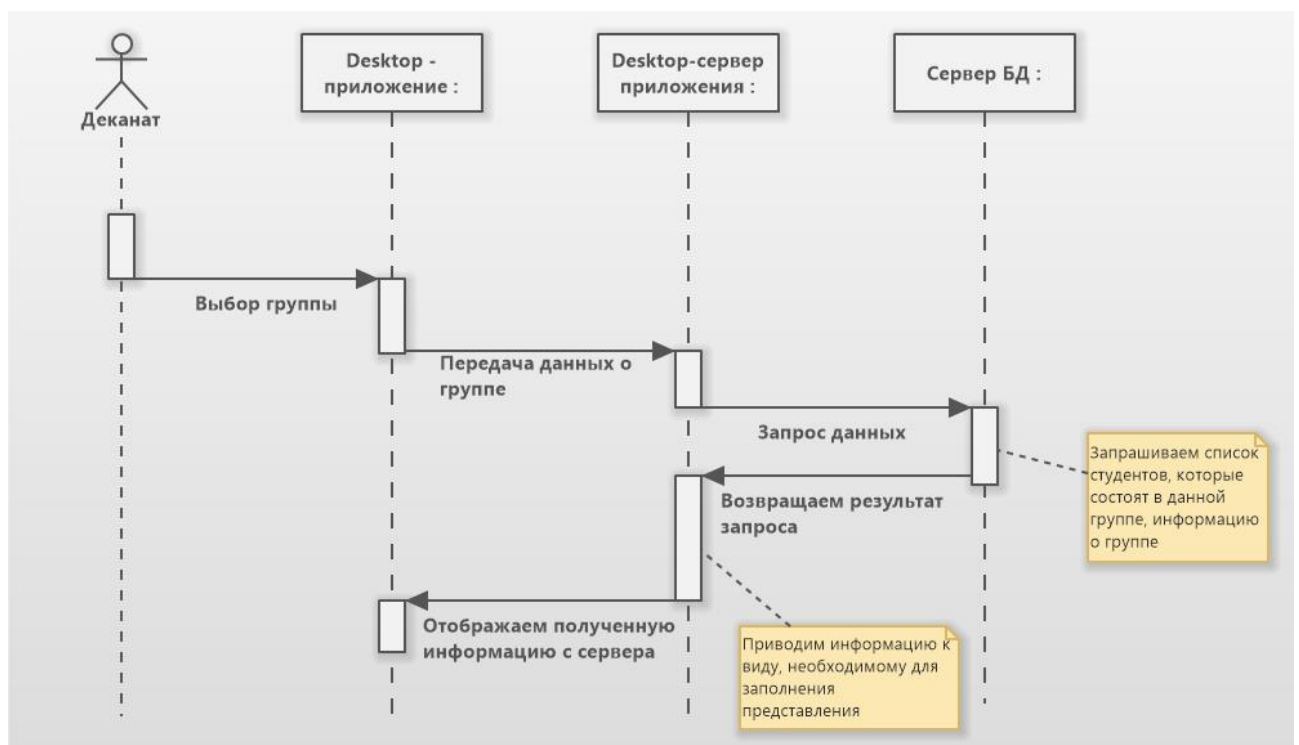


Рис. 11. Формирование подробной информации о группе

Рассмотрим часть программы, которая отвечает за управление расписанием. После перехода во вкладку управление расписанием открывается вкладка, как на рисунке 12. Изначально пользователю открывается доступ к добавлению занятий и к просмотру занятий для выбранной группы.

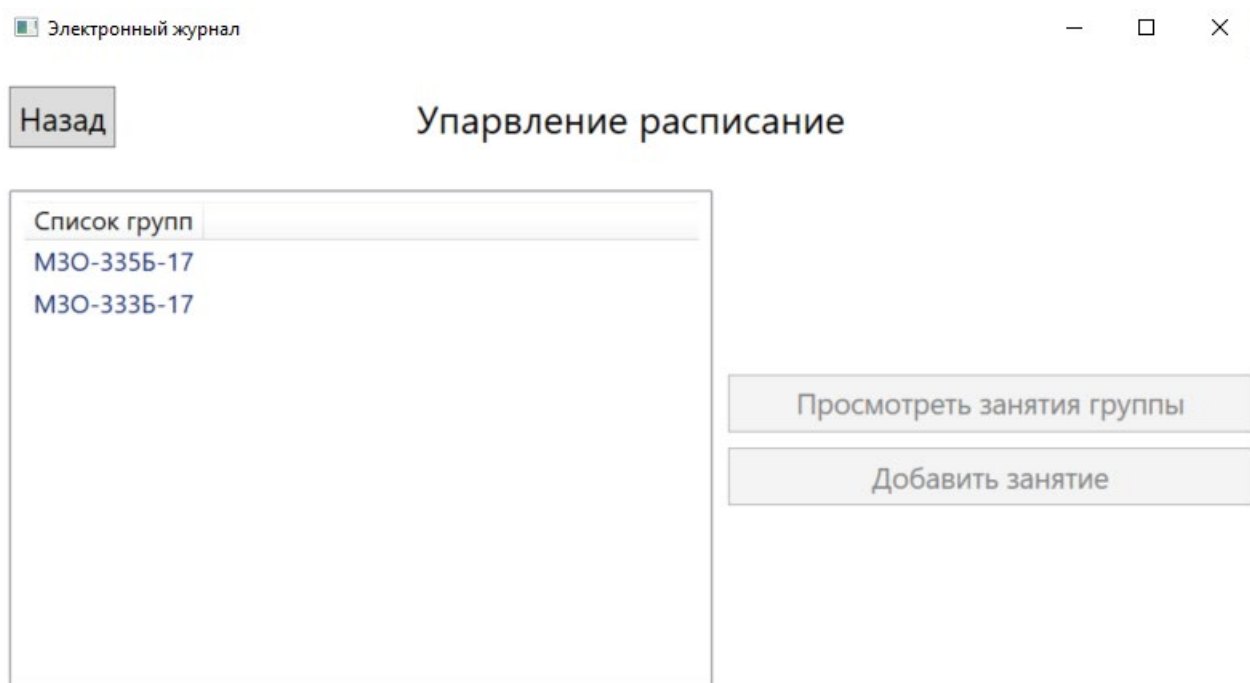


Рис. 12. Управление расписание

Рассмотрим алгоритм формирования списка занятий. Сначала нам необходимо получить информацию, для какой группы мы хотим отобразить список занятий. После выбора группы мы делаем запрос в базу данных для получения необходимых данных и на выходе получаем список предметов. Алгоритм представлен на рисунке 13. Результат работы алгоритма представлен на рисунке 14.

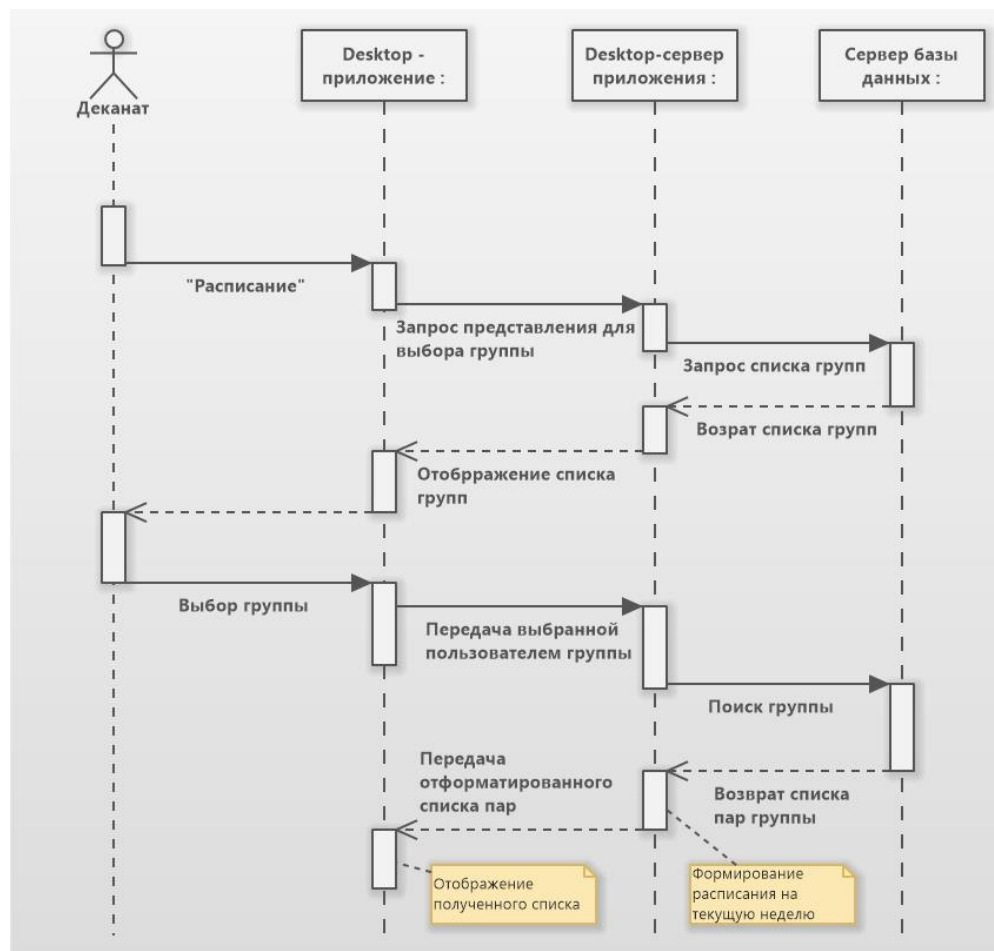


Рис. 13. Формирование списка занятий

Электронный журнал

Назад

Список занятий гр. МЗО-335Б-17

Предмет	Дата занятия
Операционные системы	25 апреля 2020 г.
Операционные системы	25 апреля 2020 г.
Web-программирование	28 апреля 2020 г.
Web-программирование	28 апреля 2020 г.
Web-программирование	28 апреля 2020 г.
Программирование в информационных системах	30 апреля 2020 г.
Программирование в информационных системах	30 апреля 2020 г.
Программирование в информационных системах	30 апреля 2020 г.
Программирование в информационных системах	30 апреля 2020 г.
Программирование в информационных системах	30 апреля 2020 г.
Программирование в информационных системах	31 мая 2020 г.
Web-программирование	1 июня 2020 г.
Web-программирование	2 июня 2020 г.

Добавить новое занятие

Подобрать информацию по занятию

Удалить занятие

Рис. 14. Список занятий группы

Для добавления новых занятий необходимо заполнить форму и нажать добавить занятие. Алгоритм для добавления новых занятий представлен на рисунке 15.

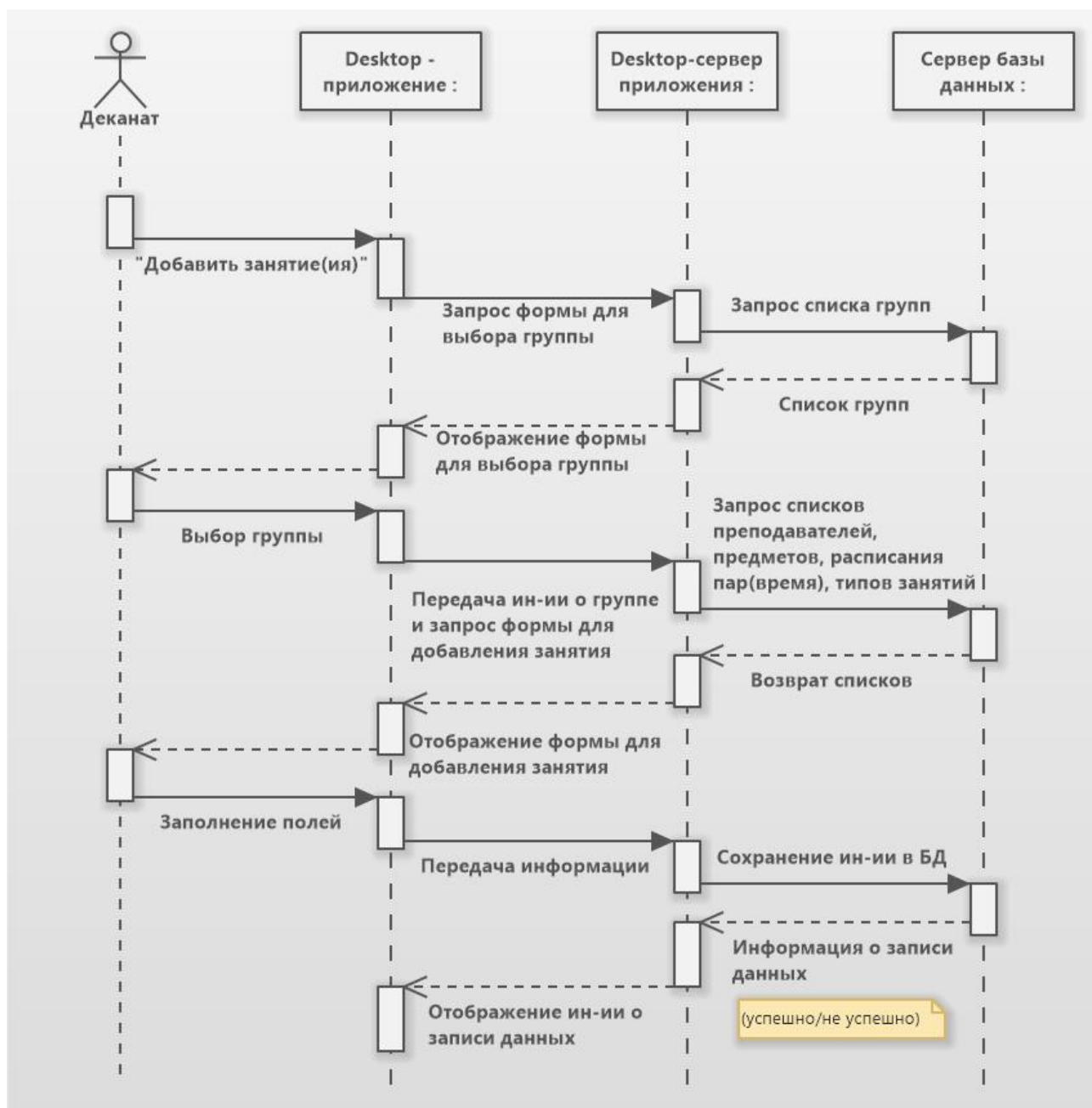


Рис. 15. Добавление новых занятий

При разработке были созданы следующие классы:

- AuthorizationModel – класс, отвечающий за проверку пользователя при авторизации;

- `DataRequestForAdmin` – класс, отвечающий за формирования списка пользователей для пользователей с модификатором доступа «админа»;
- `DataRequestForDekanat` – класс, отвечающий за формирования списка пользователей для пользователей с модификатором доступа «деканат»;
- `GeneralUserInformation` – класс, для формирования моделей;
- `NavigationKeys` - данный класс используется для определения правил навигаций. А если точнее, то для определения имен Представлений(View), которые используются при регистрации пар View и ViewModel;
- `ElectronicalJournalContext` - контекст данных для взаимодействия с БД;
- `DialogManager` - класс, позволяющий из `viewmodel` вызывать `MessegeBox`. Его удобно использовать для вывода сообщений пользователю;
- `RelayCommand` - класс, реализующий интерфейс `ICommand`. На основе объектов данного класса создаются команды по обработке событий в `ViewModel`, которые произошедших в `View`;
- `ViewModelBase` - Чтобы объект мог полноценно реализовать механизм привязки, надо реализовать в его классе интерфейс `INotifyPropertyChanged`. `ViewModelBase` - класса, который реализует данный интерфейс и является базовым для всех `ViewModel`;
- `UserSession` - класс, отвечающий за отслеживание текущей сессии пользователя в приложении;
- `ManagingUserAccountsModel` – класс, инкапсулирующий весь функционал связанные с управлением аккаунтами.

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

В папке View находятся представления с пользовательским интерфейсом.

В папке ViewModel находятся классы моделей представлений являющиеся связующим звеньями представлений и моделей.

5. ОРГАНИЗАЦИЯ ДАННЫХ

Организация входных данных.

Входными данными для программы являются:

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

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. Структура базы данных изображена на рисунке 16.

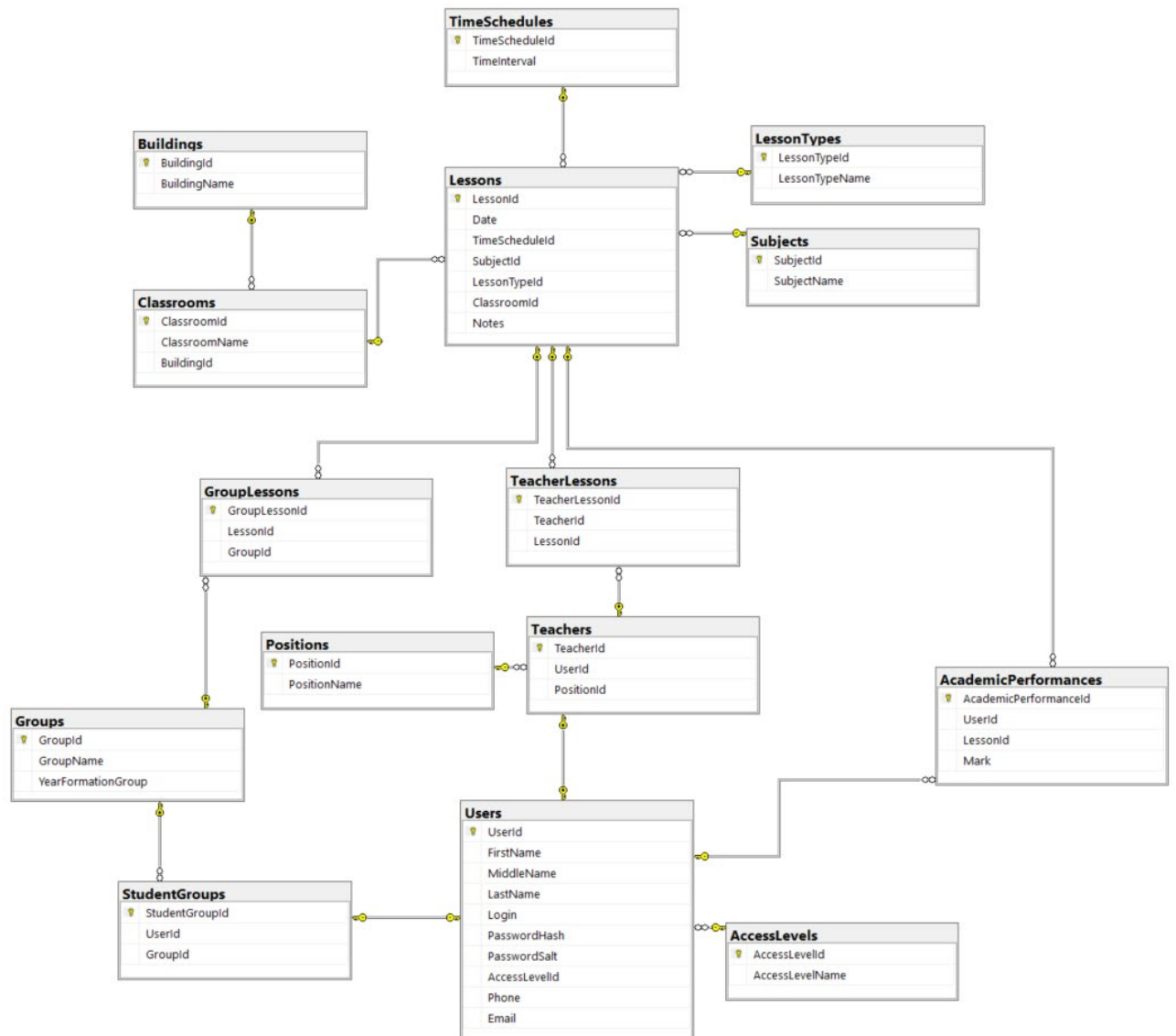


Рис. 16. Структура базы данных.

6. РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММ И ИХ ОЦЕНКА

Объект испытания

Объектом испытания является настольно приложение «ElectronicJournal_Desktop»

Цель испытания

Целью испытания является проверка работоспособности настольного приложения «ElectronicJournal_Desktop».

Требования к программе

Во время испытания было необходимо проверить следующий функционал программы:

- авторизация пользователя в приложении;
- добавление новых аккаунтов;

Метод испытания

Тестирования приложения «ElectronicJournal_Desktop» проводилось методом «черного ящика». Результаты тестирования приведены в таблице 1.

Таблица 1 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 1

	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 добавление нового аккаунта	Все поля заполнены, но введенный логин уже зарезервирован	Вывод сообщения «Пользователь с таким логином существует» и просьба ввести другой логин	Вывод сообщения «Пользователь с таким логином существует» и просьба ввести другой логин
	Все поля заполнены корректно	Вывод подробной информации о пользователе	Вывод подробной информации о пользователе

Оценка результатов тестирования

В тестовых примерах, приведенных в таблице 1, ожидаемая реакция совпадает с фактической, что доказывает работоспособность программы.

ЗАКЛЮЧЕНИЕ

В соответствии с поставленной задачей было разработано настольное 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. Эндрю Троелсен «Язык программирования C# и платформа .NET 4.5 6-е издание. – М: ООО «И.Д. Вильямс», 1310 с.;
3. 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.
4. Руководство по WPF: <https://metanit.com/sharp/wpf/>;
5. Windows Presentation Foundation: <https://docs.microsoft.com/ru-ru/dotnet/framework/wpf/>;
6. Руководство по Entity Framework Core: <https://metanit.com/sharp/entityframeworkcore/>;
7. Руководство по работе с библиотекой MvvmNavigation: <https://github.com/Egor92/MvvmNavigation/blob/master/README.RUS.md>

08335 – 01 81 01

ПРИЛОЖЕНИЕ А

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