

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт вычислительной математики и информационных технологий

Направление подготовки: 09.03.03 – прикладная информатика

**КУРСОВАЯ РАБОТА
ПО ДИСЦИПЛИНЕ ПРОЕКТИРОВАНИЕ ЧЕЛОВЕКО-
МАШИННОГО ИНТЕРФЕЙСА
ПО ТЕМЕ
КОНФЕРЕНЦИЯ**

Студент 2 курса
группы 09-951

“ ” 2020 г.

Балашов А.Е.

Научный руководитель
кандидат физико-математических наук,
доцент кафедры технологий программирования

“ ” 2020 г.

И.С. Балафендиева

Введение

В целях повышения эффективности процесса проведения конференций в университете ставится задача разработки новой системы учета и регистрации участников, а также анализа конференций взамен старой системы. Новая система должна позволять преподавателям создавать конференции, всем желающим пользователям - регистрироваться на конференцию в качестве гостя, выводить на экран длительность конференции, новизну темы и проводить анализ конференции.

Анализ предметной области

1. Требования к системе

Пользователи должны иметь доступ к системе через сеть университета. Создатели конференции должны иметь возможность указывать тему конференции, дату ее начала и длительность. Вся информация о конференциях содержится в файлах. Участники конференций могут запрашивать список предстоящих и прошедших конференций. Информация о конференции должна включать ее тему, дату, продолжительность и информацию о выступающих. Также они могут отписаться от участия в конференции.

2. Основной функционал приложения

Для того чтобы пользователю зарегистрироваться на конференцию, ему необходимо пройти авторизацию под своим логином и паролем, выбрать интересующую его конференцию и подтвердить намерение зарегистрироваться, после чего система регистрирует пользователя на участие в конференции.

В случае авторизации преподавателя, пользователю будет доступно создание конференций, а также редактирование, просмотр списка участников и удаление созданных им конференций.

На основе информации, описанной выше, выделим действующих лиц, варианты использования и однозначно интерпретируем понятия с помощью глоссария.

3. Глоссарий

Термин	Значение
Конференция	Мероприятие, проводимое в университете. Необходимо задать его тему, длительность, дату и описание.
Тема конференции	Предлагаемая тема данной конференции в конкретном семестре. Задается преподавателем при создании конференции
Список конференций	Полный набор всех конференций, предлагаемых университетом.

4. Действующие лица

Студент — регистрируется на конференции в качестве слушателя

Профессор — регистрируется на конференции в качестве слушателя или выступающего, создает новые конференции.

5. Варианты использования

- Войти в систему
- Зарегистрироваться на конференцию
- Изменение информации о конференции

5.1. Вариант использования "Войти в систему"

Краткое описание:

Данный вариант использования описывает вход пользователя в систему.

Основной поток событий:

Данный вариант использования начинает выполняться, когда пользователь хочет войти в систему.

1. Система запрашивает имя пользователя и пароль.
2. Пользователь вводит имя и пароль.
3. Система подтверждает имя и пароль, после чего открывается доступ в систему с определенными привилегиями в зависимости от типа учетной записи.

Альтернативные потоки:

Неправильное имя/пароль: Если во время выполнения основного потока обнаружится, что пользователь ввел неправильное имя и/или пароль, система выводит сообщение об ошибке. Пользователь может вернуться к началу основного потока или отказаться от входа в систему, при этом выполнение варианта использования завершается.

5.2. Вариант использования «Зарегистрироваться на конференцию»

Краткое описание:

Данный вариант использования описывает регистрацию пользователя на конференцию.

Основной поток событий:

Данный вариант использования начинает выполняться, когда пользователь хочет зарегистрироваться на конференцию в качестве слушателя.

1. Пользователь находит необходимую конференцию
2. Нажимает кнопку регистрации

Предусловия:

Перед началом выполнения данного варианта использования пользователь должен войти в систему.

5.3. Вариант использования «Изменение информации о конференции»

Краткое описание:

Данный вариант использования описывает изменение информации о конференции.

Основной поток событий:

Создатель конференции изменяет название конференции, ее дату или длительность.

Предусловия:

Перед началом выполнения данного варианта использования пользователь должен войти в аккаунт с привилегиями преподавателя.

Реализация приложения

1. Хранение данных

Для хранения данных используется набор текстовых файлов. (Рисунок 1).

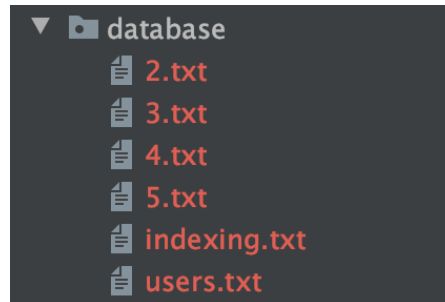


Рисунок 1. Структура хранения данных

2. Классы системы

Необходимо реализовать класс конференций для взаимодействия с базой данных - удаления / добавления участников, обновления информации о конференции, создания новых конференций; класс пользователя - для авторизации и определения типа учетной записи; класс базы данных - для получения и записи информации о конференциях.

2.1 Класс Conference

Содержит информацию о конференции (Рисунок 2), а также методы добавления и удаления участника из конференции (Рисунок 3).

```
public class Conference
{
    public int id;
    public String name;
    public Calendar date;
    public String duration;
    public int professorId;
    public String description;
    public String link;
    public ArrayList<Integer> participants = new ArrayList<>();
}
```

Рисунок 2. Поля класса Conference

```
public void addParticipant(User user)
{
    this.participants.add(user.id);
    Main.database.updateParticipants(this.id, this.participants);
}

public void deleteParticipant(User user)
{
    this.participants.remove((Integer) user.id);
    Main.database.updateParticipants(this.id, this.participants);
}
```

Рисунок 3. Методы класса Conference

2.2. Класс User

Содержит информацию о пользователе (Рисунок 4) и метод авторизации (Рисунок 5).

```
public class User
{
    public int id;
    public String name;
    public boolean isProfessor;
}
```

Рисунок 4. Поля класса User

```

public boolean authorize(String login, String password)
{
    ResultSet resultSet = Main.database.getUsers();
    try
    {
        while (resultSet.next())
        {
            if (resultSet.getString( s: "login").equals(login) &&
                resultSet.getString( s: "password").equals(password))
            {
                this.id = resultSet.getInt( s: "id");
                this.name = resultSet.getString( s: "name");
                this.isProfessor = resultSet.getBoolean( s: "is_professor");
                return (true);
            }
        }
    }
    catch (Exception e)
    {
        System.err.println("[authorize]");
        e.printStackTrace(System.err);
    }
    return (false);
}

```

Рисунок 5. Метод класса User

2.3. Класс Database

Содержит методы взаимодействия с информацией из файлов: получение списка всех пользователей, списка всех конференций, получение имени пользователя по id, добавление / изменение / удаление конференций (Рисунок 6).

```

public ArrayList<Conference>    getConferences()
{...}

public ArrayList<User>          getUsersList(ResultSet resultSet)
{...}

public void                     updateParticipants(int conferenceId, ArrayList<Integer> participants)
{...}

public String                   getUserName(int id)
{...}

public void                     addConference(Conference conference)
{...}

public void                     deleteConference(int id)
{...}

```

Рисунок 6. Методы класса Database

3. Структура системы

Все необходимые файлы для компиляции программы собраны в папке `src`, лежащей в корне проекта. В папке `src` файлы разделены на логические группы (Рисунок 7)

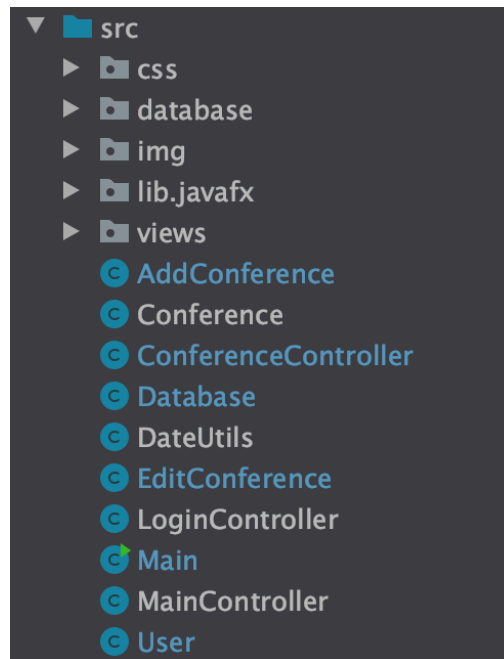


Рисунок 7. Структура папки `src`

4. Проектирование Графического интерфейса

4.1. Геометрия основного интерфейса

По умолчанию приложение запускается в окне размером 850x650, которое можно уменьшить или увеличить. В верхней части окна расположен блок навигации по приложению. На первой вкладке находится список предстоящих конференций (Рисунок 8), на второй - прошедшие конференции (Рисунок 9), также в правой части навигационного блока расположено имя пользователя. [1]

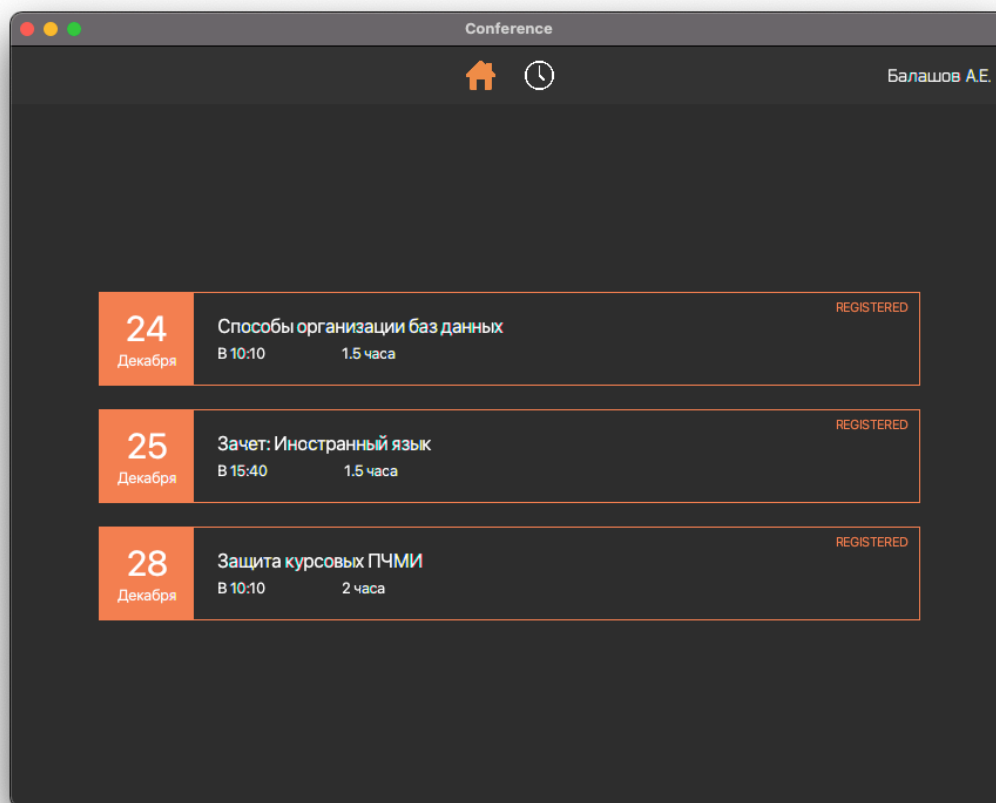


Рисунок 8. Основной интерфейс программы

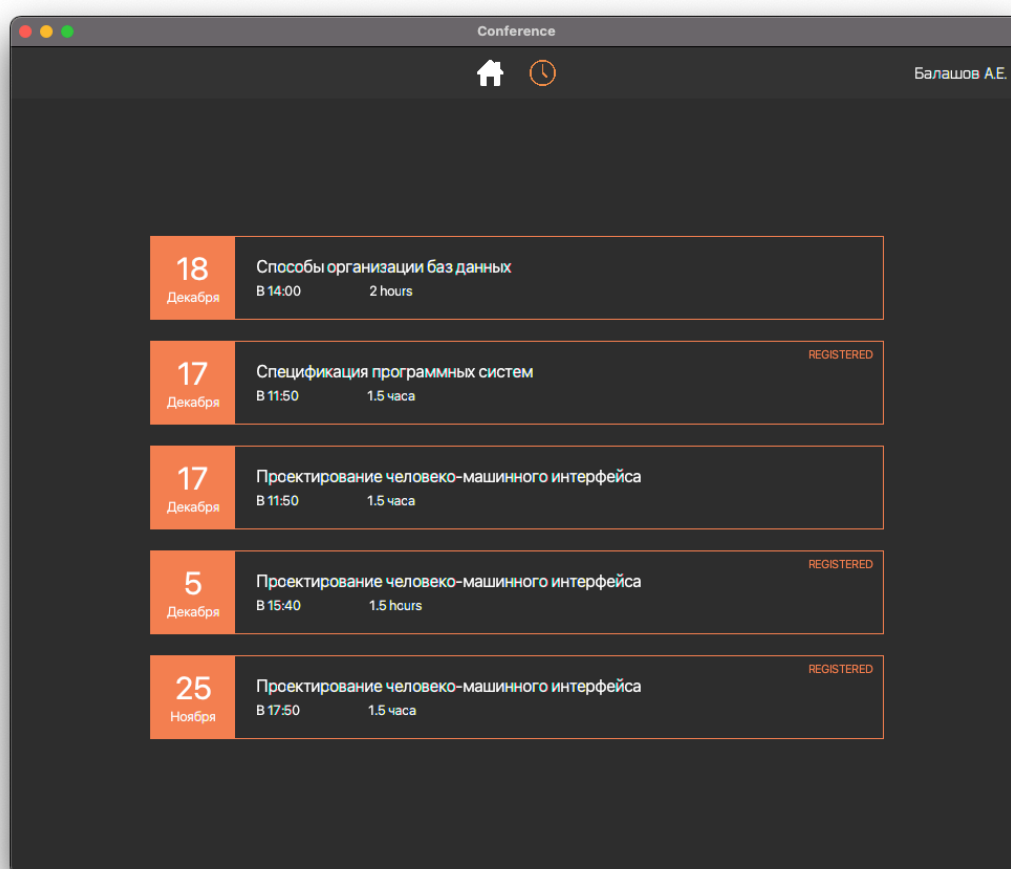


Рисунок 9. Основной интерфейс программы (прошедшие конференции)

4.2. Информация о конференциях

При нажатии на любую конференцию из списка открывается дополнительное окно, содержащее название конференции, дату, ФИО преподавателя, описание конференции, ссылку на нее и две кнопки: close - закрывает окно, subscribe / unsubscribe - регистрация или отписка от конференции (Рисунок 10).

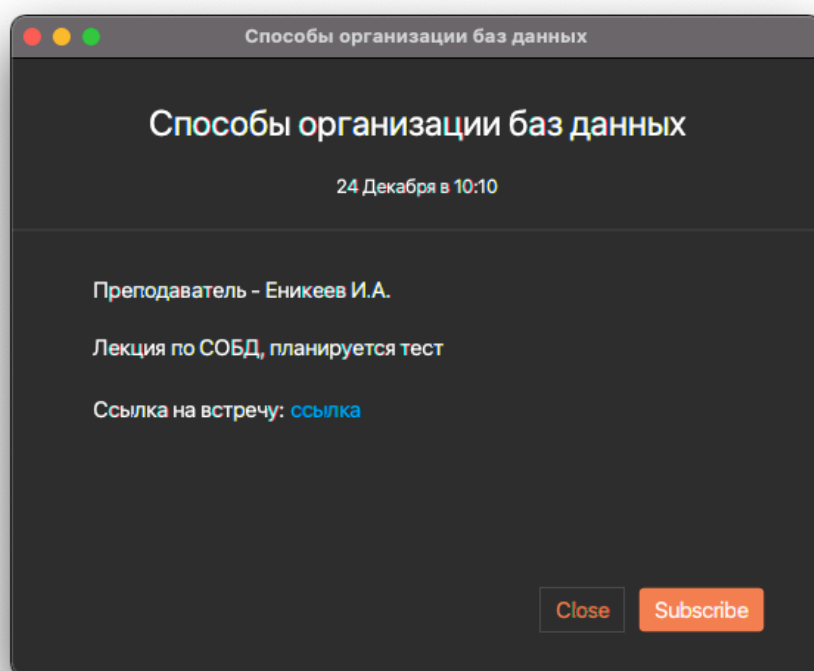


Рисунок 10. Окно дополнительной информации

4.3. Профиль преподавателя

При входе с аккаунта преподавателя в блоке навигации приложения добавляется иконка «+» (Рисунок 11), по нажатию которой открывается окно создания новой конференции (Рисунок 12). Также в правой части созданных преподавателем конференций располагаются кнопки удаления конференции, и при просмотре информации о созданных конференциях у преподавателя добавляется список зарегистрированных на конференцию пользователей (Рисунок 13), а также возможность изменять информацию о конференции по нажатию на кнопку «edit» (Рисунок 14).

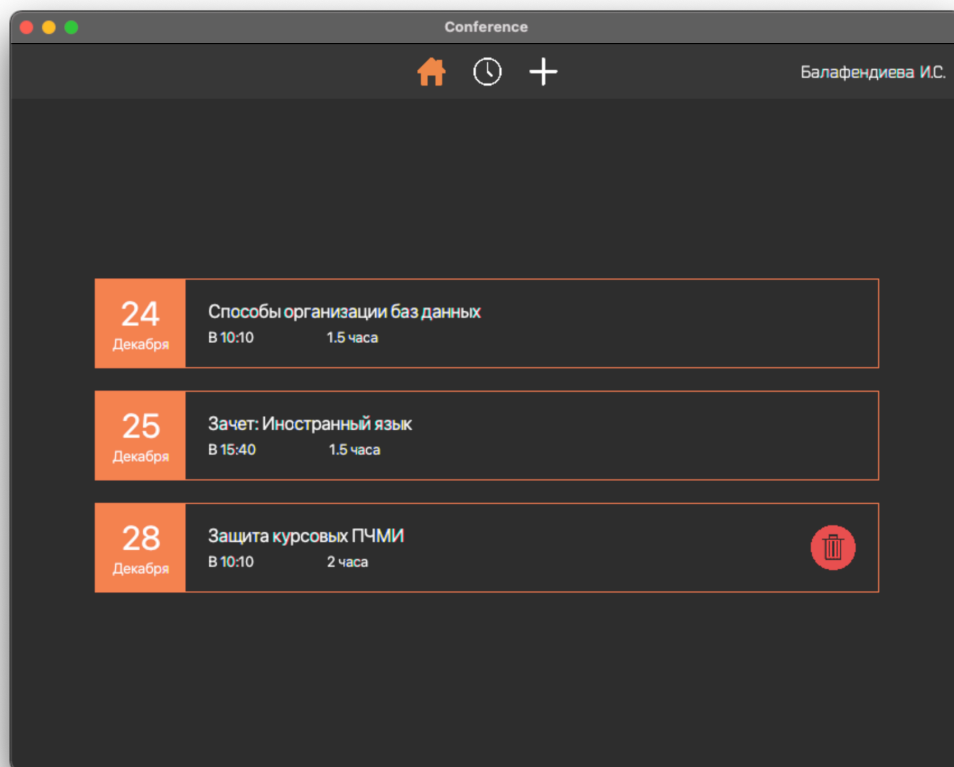


Рисунок 11. Окно преподавателя

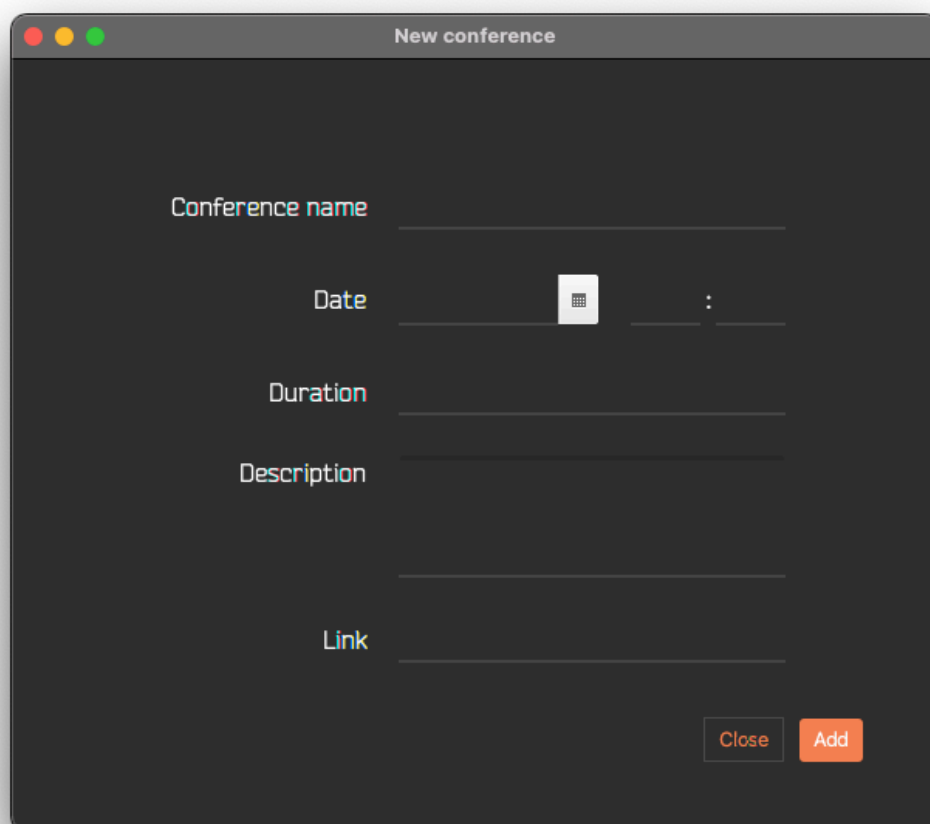


Рисунок 12. Окно добавления новой конференции

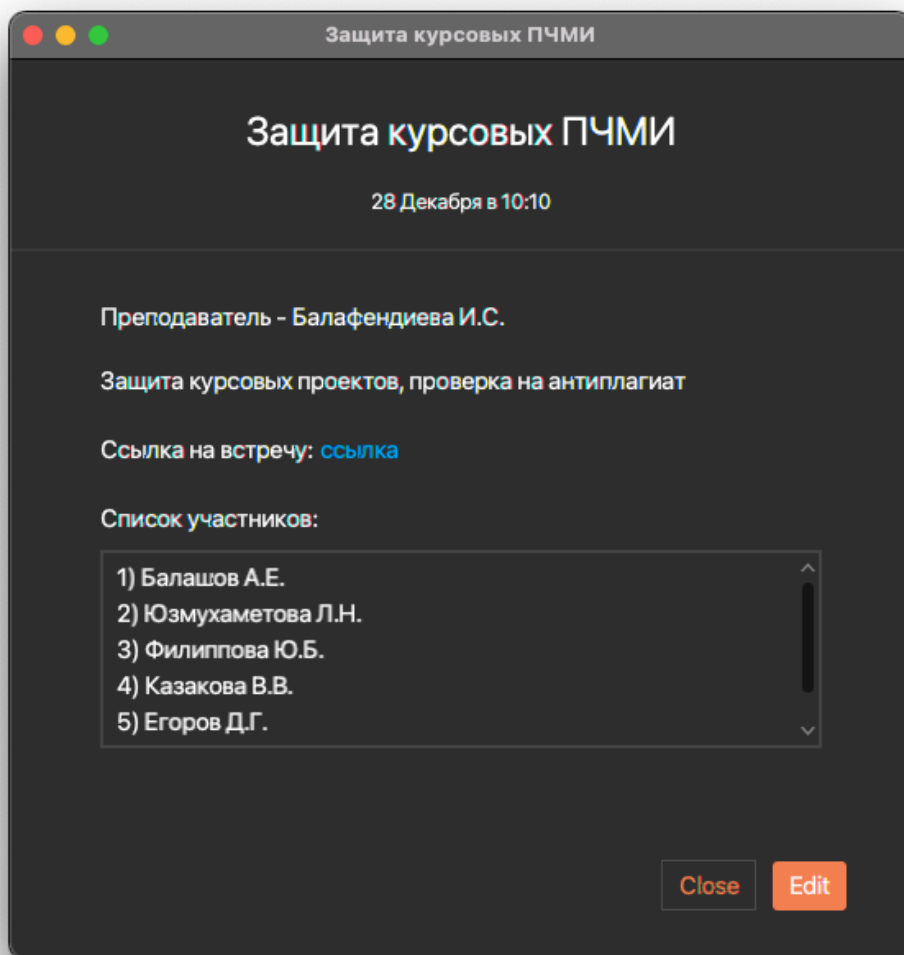


Рисунок 13. Информация о созданной конференции

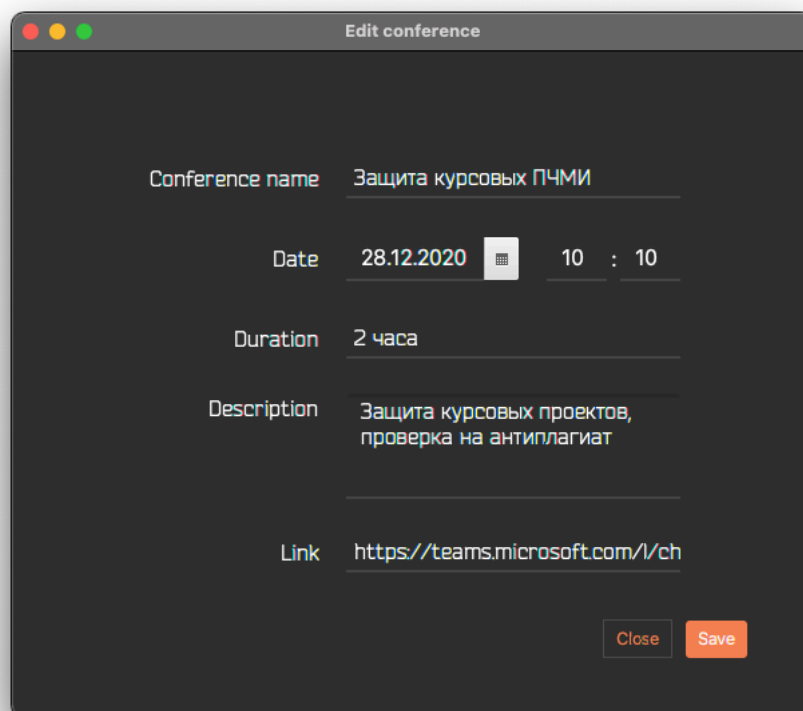


Рисунок 14. Изменение информации о конференции

4.4. Окно авторизации

Окно авторизации имеет поле для ввода логина и поле для ввода пароля (Рисунок 15). В случае успешной авторизации пользователь попадает на главный экран приложения, в противном случае - выводится надпись «Error» (Рисунок 16).

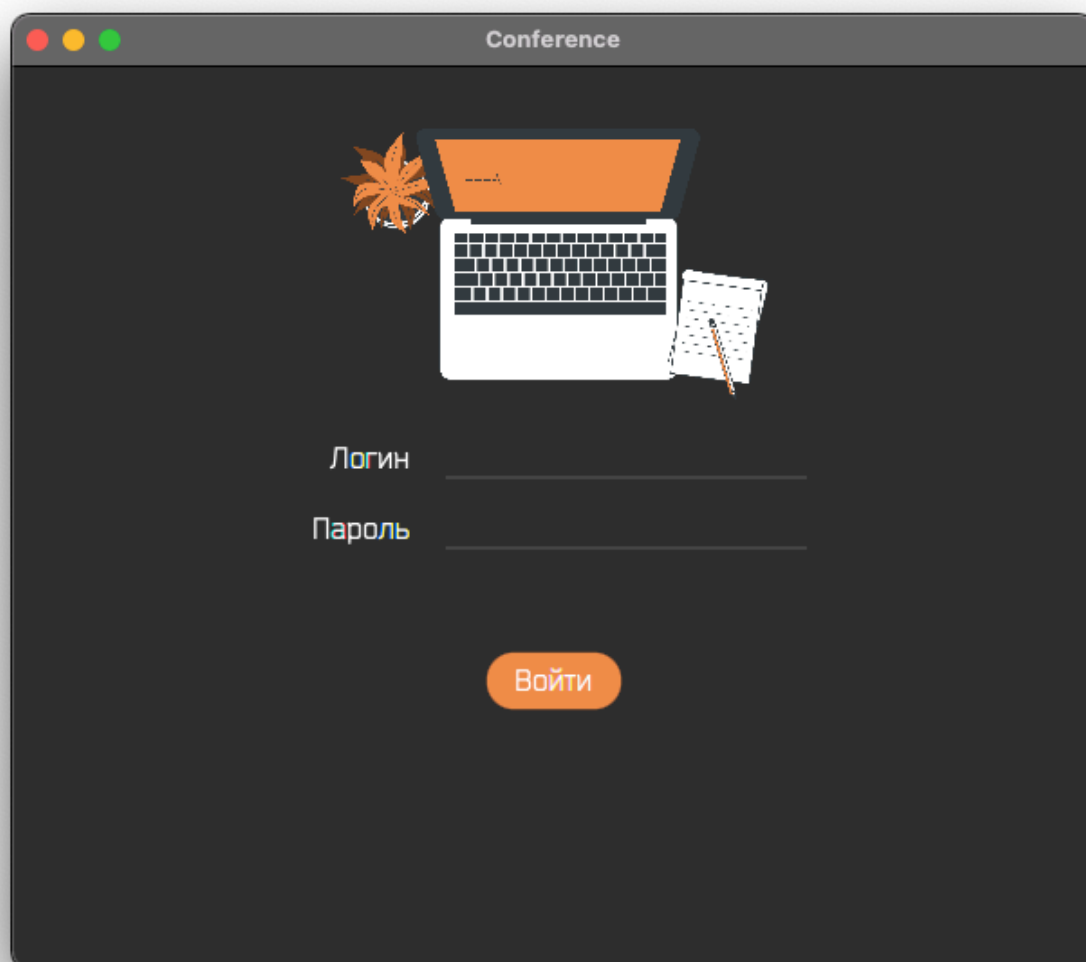


Рисунок 15. Окно авторизации

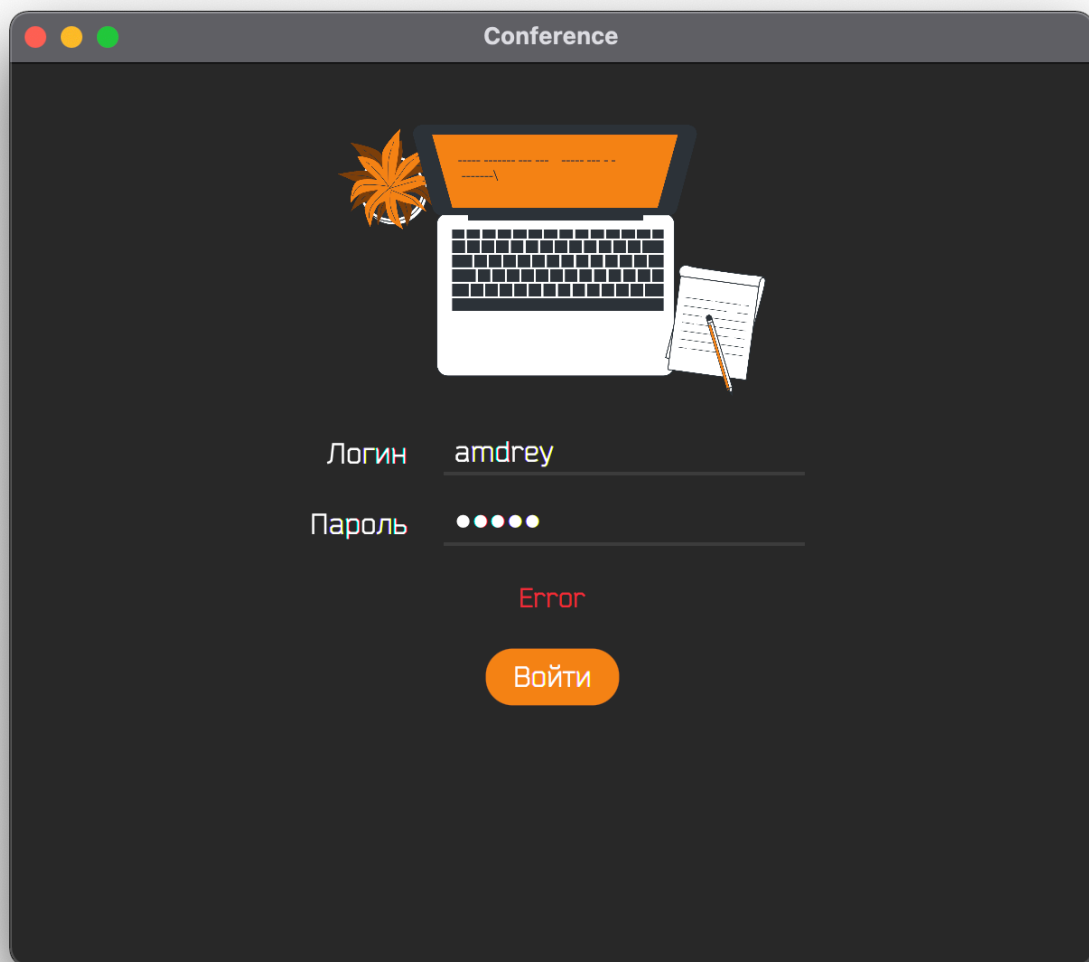


Рисунок 16. Неуспешная авторизация

Заключение

В результате данной курсовой работы было создано приложение для удобной регистрации студентов и преподавателей на конференции, учета слушателей и редактирования информации о предстоящих конференциях. Созданное приложение имеет интуитивно понятный интерфейс, динамически расширяющийся для работы преподавателей, единую базу данных, синхронизирующую работу пользователей, а также возможность его дальнейшей доработки в различных направлениях. В процессе работы автор приобрел для себя полезные навыки проектирования человеко-машинного интерфейса.

Список использованных источников

1. Руководство JavaFX [Электронный ресурс] – Режим доступа: URL: <https://docs.oracle.com/javase/8/javafx/api/> (дата обращения: 26.12.2020)